Laboratorio di Web Scraping
MAGICIAN: Mining and gAmblinG servIces sCrapIng and ANalysis
Progetto di Fine Corso A.A. 2024/25

Lo scopo del progetto è lo studio del comportamento dei seguenti due servizi attivi nel primo periodo di attività di **Bitcoin** (anni 2009-2012)

- la mining pool, **Deepbit.net** (https://en.bitcoin.it/wiki/DeepBit9). Una **mining pool** è un gruppo di miners (cioè nodi che cercano di risolvere la PoW) che uniscono le proprie risorse di calcolo per ridurre la varianza nella risoluzione della PoW in modo da inserire il blocco successivo nella blockchain ed ottenere la conseguente ricompensa più frequentemente;
- un servizio di scommesse (gambling), **DiceOnCrack**.
  Un **servizio di gambling** di **Bitcoin** è un **servizio di gioco d'azzardo online** (tipo casinò, lotteria, gioco di dadi, ecc.) che utilizza **Bitcoin o altre criptovalute** come metodo di pagamento o scommessa.

A questo scopo, è richiesto di lavorare su un DataSet, fornito con il progetto, contenente il sottoinsieme delle transazioni di **Bitcoin** effettuate dal momento del suo lancio, avvenuto nel gennaio 2009, fino al 31-12-2012. Vengono richieste un insieme di analisi generali sulle transazioni di questi due servizi contenute nel DataSet, e un insieme di analisi che sfruttano le informazioni ottenute mediante scraping dal sito **WalletExplorer** per la deanonimizzazione di alcuni indirizzi.

1. Descrizione del DataSet

Viene fornito un DataSet di **Bitcoin** che contiene le transazioni incluse nei blocchi compresi tra il blocco genesi, minato da Satoshi Nakamoto in data **03-01-2009, 17:15:05** e il blocco di altezza **214562**, minato in data **31-12-2012, 23:52:37.** 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.

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 all'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: encoding dell'hash corrispondente all'indirizzo. E' l'encoding dell'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.

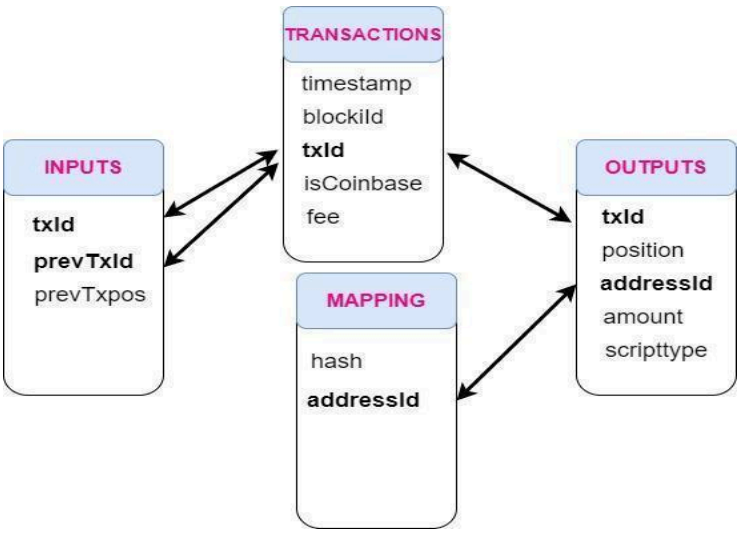

Figura 1: Struttura del DataSet

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. Acquisizione dei dati

I dati per le analisi sono sia quelli contenuti nel DataSet che quelli reperiti mediante **WalletExplorer** (https://www.walletexplorer.com/), la cui pagina principale è mostrata in Fig. 2. Il servizio effettua clustering degli indirizzi **Bitcoin**, applicando un'euristica per individuare tutti gli indirizzi probabilmente appartenenti alla stessa entità, inoltre tenta di associare un'identità reale ad ogni cluster/wallet di indirizzi, quando possibile. WalletExplorer utilizza il seguente algoritmo di clustering. Gli indirizzi vengono uniti transitivamente tra loro in un unico cluster se sono stati **spesi insieme in una stessa transazione**. Quindi, se gli indirizzi A e B sono stati spesi insieme nella transazione T1, e gli indirizzi B e C sono stati spesi insieme nella transazione T2, allora tutti gli indirizzi A, B e C saranno considerati come parte di uno stesso wallet. **WalletExplorer** associa quindi un nome e un colore al wallet così individuato. Mediante scraping di siti web e forum che si occupano di **Bitcoin**, il servizio cerca anche di deanonimizzare il wallet ovvero di associare una identità/servizio agli indirizzi di un cluster/wallet, se ci riesce associa il nome reale del servizio a quel wallet, altrimenti il nome associato è generato da WalletExplorer e non corrisponde ad un servizio reale.

WalletExplorer consente :

- ricerca di un wallet mediante il suo nome: restituisce le transazioni di quel wallet e gli indirizzi associati a quel wallet (al cluster corrispondente a quel wallet);
- ricerca di un indirizzo mediante il suo hash: restituisce le informazioni del wallet a cui appartiene quell'indirizzo;
- ricerca di transazioni mediante il loro hash: restituisce le informazioni sulla transazione.

Il passo iniziale del progetto è l'acquisizione di tutti gli indirizzi **`Deepbit_addrs`** associati a **`Deepbit.net`** e di tutti gli indirizzi **`DiceOnCrack_addrs`** associati a **`DiceOnCrack.`** A questo scopo, effettuare una query, mediante scraping su Wallet Explorer, utilizzando il nome del servizio. I dati **devono essere acquisiti mediante scraping, non dai file .csv disponibili sul sito.**

Per approfondire l'euristica ed i servizi maggiormente presenti su **`Bitcoin`** nel primo periodo di attività si può fare riferimento a [1].

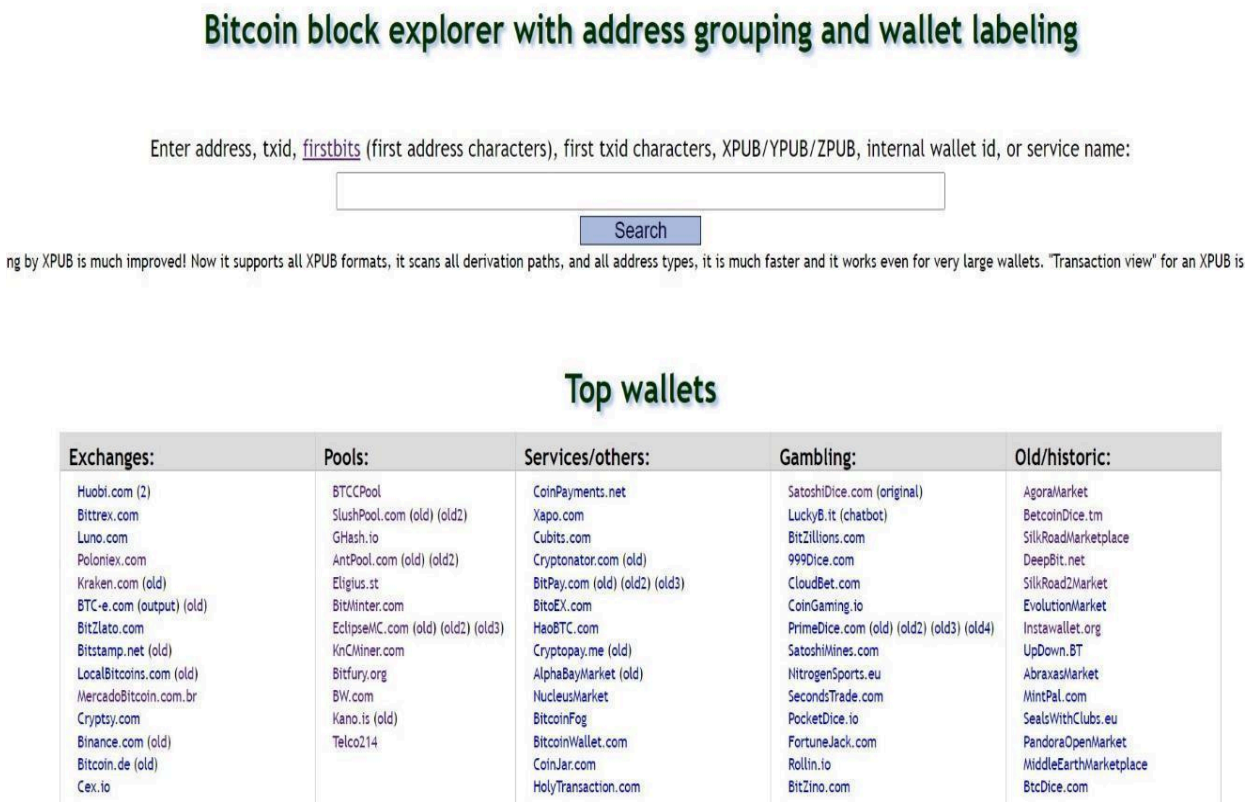

Fig 2. Main page di WalletExplorer

3. Analisi della Mining Pool

Individuare nel DataSet tutte le transazioni relative a **`Deepbit.net`**, ovvero le transazioni che presentano almeno un indirizzo di input o di output appartenente a **`Deepbit_addrs.`**

Si effettuino quindi le seguenti analisi :

- distribuzione dei blocchi minati da **`Deepbit.net`** (~~corrispondenti alle transazioni COINBASE con output verso un indirizzo appartenente a Deepbit_addrs~~) nel periodo compreso nel Dataset, considerando diversi campionamenti temporali nel periodo considerato, e utilizzando i metodi offerti da PANDAS per la gestione di serie temporali.
  Si tenga presente che **`Deepbit.net`** utilizza lo schema presentato in Fig. 3 per ricevere ricompense e fee per un blocco minato.

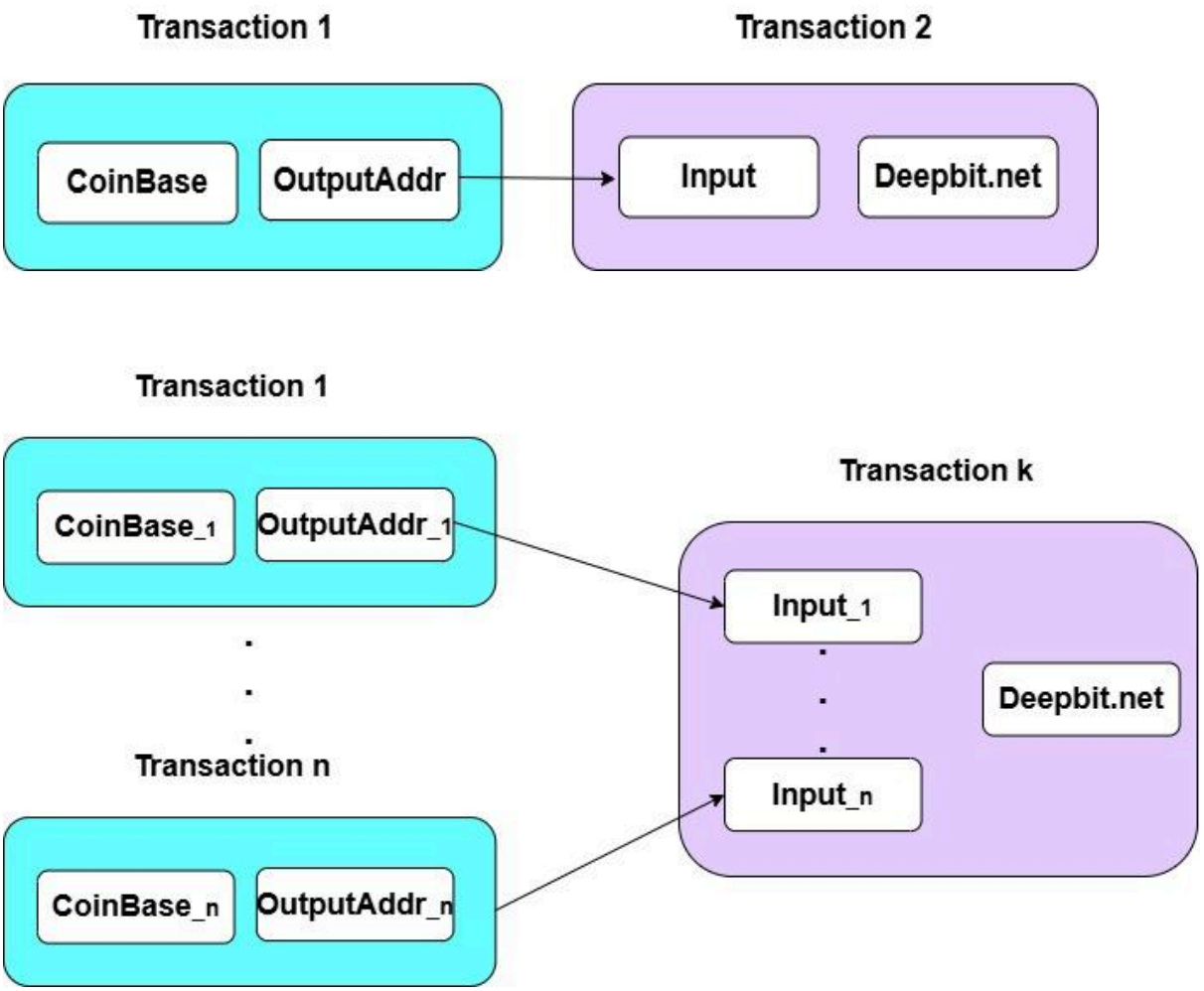

Fig 3: Pattern per la distribuzione di Rewards e Fees in Deepbit.net

Come si nota nella parte superiore di Fig 3. la Transazione 1 è una CoinBase con un valore in output pari alla somma della reward più le fee del blocco minato, in cui l'indirizzo di output (OutputAddr) non appartiene (in generale) a **`Deepbit_addrs.`** Una transazione successiva prende quindi in input il valore associato ad OutputAddr e lo trasferisce a un indirizzo in **`Deepbit_addrs.`** Come mostrato nella parte inferiore di Fig.3, lo schema precedente può comprendere più Coinbase, Transaction 1, ... Transaction n, ognuna con un diverso indirizzo di output. Tali indirizzi sono quindi riferiti in input da una stessa transazione **Transaction k** che poi li trasferisce in output a un indirizzo appartenente a **`Deepbit_addrs.`** Questi pattern venivano probabilmente utilizzati dalla mining pool per garantire un maggior livello di privacy.

Per individuare le Coinbase di **`Deepbit.net`** occorre quindi utilizzare i pattern precedenti. Altri pattern, ad esempio con distanza maggiore tra la/le CoinBase e la transazione che trasferisce il valore a un indirizzo in **`Deepbit_addrs`**, non devono essere considerati.

- distribuzione delle fee ricevute dalla mining pool **`Deepbit.net`**, per ogni COINBASE. Si ricorda che quando un miner risolve la PoW riceve, oltre alla ricompensa in **`bitcoin`** appena coniati (che può essere di 50 o 25 BTC nel DataSet considerato), la somma delle fee relative alle transazioni incluse nel blocco minato.
- calcolo degli UTXO (Unspent Transaction Output) relativi a **`Deepbit.net`**, per ogni mese compreso nel periodo considerato, ovvero calcolo della somma di tutti gli amont che

risultano non spesi alla fine di ogni mese (si noti che il valore non speso può essere stato generato anche nei mesi precedenti). Visualizzare l'andamento degli UTXO nel tempo.

Si consideri quindi, tra tutte le transazioni del blocco di altezza 153932, la transazione con timestamp 2011-11-19 07:39:30 e id nel DataSet 1883820 (corrispondente alla transazione con hash e229f0b8f826e54d173dc266b4312b4e7c355850f625c917672a2c04b3a308 ca su Wallet explorer).
La transazione individuata ha un solo indirizzo di output, con destinatario **`Deepbit.net`**. Partendo da questa transazione, costruire un grafo corrispondente ad una catena di pagamenti, come segue:

- la visita della catena di transazioni avviene seguendo sempre gli output corrispondenti ad un change address (un indirizzo sempre appartenente a **`Deepbit_addrs`**), se presente ed unico. La visita si interrompe se si incontra una transazione in cui nessun indirizzo di output è un change address oppure esiste più di un change address;
- durante la visita, inserire gli indirizzi corrispondenti agli output non appartenenti a **`Deepbit_addrs`** (non change addresses), in un insieme **`OthersDeepbit.`**

Dopo aver costruito il grafo:

- per ogni coppia di transazioni ti, ti+1 della catena calcolare rispettivamente, la differenza tra i timestamp delle due transazioni e tra i valori inviati sui change address e visualizzare le differenze ottenute su un grafico;
- considerare gli indirizzi in **`OthersDeepbit,`** e verificare, effettuando scraping su WalletExplorer, se corrispondono a entità deanonimizzate;
- rappresentare la catena di transazioni mediante un grafo costruito mediante **`NetworkX`** e rappresentato scegliendo un opportuno layout;
- il pattern di transazioni indotto dalla catena si ripete spesso nella blockchain di **`Bitcoin`**. Ipotizzare quale comportamento degli utenti/client può avere generato questo pattern.

4. Analisi del servizio di gambling

Individuare nel DataSet tutte le transazioni relative a **`DiceOnCrack,`** ovvero le transazioni che presentano almeno un indirizzo di input o di output appartenente a **`DiceOnCrack_addrs,`** quindi ordinare in senso crescente le transazioni, in base al loro **blockid**. Ricordiamo che ogni transazione riporta il **blockid** che rappresenta l'identificatore del blocco in cui quella transazione è stata inserita (l'altezza del blocco nella blockchain), quindi se le transazioni hanno lo stesso blockid, questo implica che sono state inserite in uno stesso blocco.
Considerare le transazioni di **`DiceOnCrack`** effettuate in un intervallo temporale del **`2012,`** ad esempio nel giorno **26/12/2012** (l'intervallo temporale può essere ampliato per rendere l'analisi più significativa, ma solo dopo aver verificato di non essere bloccati da **`Wallet Explorer)`**, quindi:

- individuare gli insiemi di transazioni con lo stesso block height;
- per ogni insieme di transazioni così individuato, individuare i sottoinsiemi (cluster) di transazioni caratterizzati dal fatto che, se si considera l'insieme IND di tutti gli indirizzi di input delle transazioni del cluster, tutti gli indirizzi in IND appartengono allo stesso wallet W (escludendo gli eventuali cluster associati a **`DiceOnCrack).`** Per individuare il wallet W corrispondente a un indirizzo, effettuare scraping su **`Wallet Explorer`**, accedendo mediante l'hash dell'indirizzo in input considerato;
- visualizzare, per ogni blocco, la dimensione media dei cluster, considerando solo i cluster di dimensione maggiore o uguale a 2;
- discutere quale comportamento dei gambler può avere dato origine a questo pattern di transazioni.

5. 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 relazione, integrata in un unico Notebook**.** Generare il pdf del Notebook e sottomettere tutto il materiale in formato .pdf ;
- il codice deve essere sviluppato in **Python** e per la parte di scraping si devono utilizzare le librerie **BeautifulSoup/Selenium;**
- la scelta di opportune strategie di visualizzazione delle statistiche richieste influirà sulla valutazione del progetto.

Il Notebook ed eventuali altri contenuti devono essere consegnati su Moodle in un unico archivio compresso in formato zip. La data pubblicata sul sito è la data di consegna del progetto. L’esame si svolgerà, su appuntamento, nella settimana successiva alla data di consegna.

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] Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G. M., & Savage, S. (2013, October). A fistful of bitcoins: characterizing payments among men with no names. In *Proceedings of the 2013 conference on Internet measurement conference* (pp. 127-140).