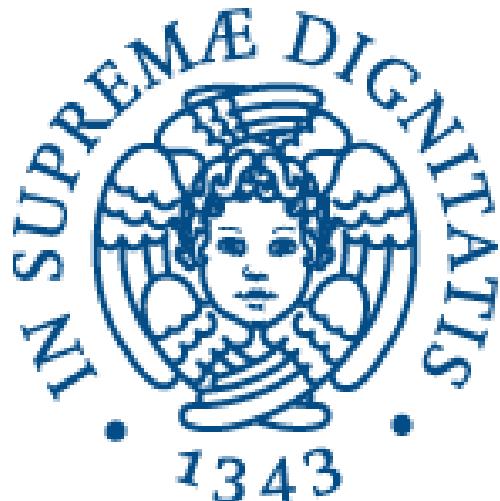


UNIVERSITÀ DI PISA



Dipartimento di Informatica

Corso di Laurea Triennale in Informatica

Analisi del social media token YUP

Relatori:

Prof.ssa Barbara GUIDI

Dott. Andrea MICHENZI

Presentata da:

Biancamaria BOMBINO

Anno Accademico 2020/2021

Indice

1	Introduzione	4
2	Stato dell'arte	10
2.1	Blockchain	10
2.1.1	Decentramento	13
2.1.2	Struttura del blocco	13
2.1.3	Validazione dei blocchi	14
2.2	Ethereum	16
2.2.1	Account in Ethereum	17
2.2.2	Transazioni	18
2.2.3	Ether	21
2.2.4	Smart contract	22
2.2.5	Token ERC20	23
2.2.6	MEV: Miner Extractable Value	24
2.3	Formato delle transazioni	27
2.4	DeFi ed Exchange Decentralizzati (DEX)	32
2.4.1	Uniswap	34
2.4.2	Token Uniswap (UNI)	36
2.4.3	DEX in Ethereum	37
3	Analisi del social token YUP	38
3.1	Piattaforma Yup	38
3.2	Servizi di Yup	40

3.2.1	Registrazione	40
3.2.2	Attività di voto	41
3.2.3	Creazione di Collezioni	42
3.2.4	Valore Sociale	43
3.2.5	Integrazioni	44
3.3	YUP Protocol	44
3.3.1	Influenza	45
3.3.2	Meccanismo delle ricompense	47
3.4	YUP Token	49
3.5	Yup in Ethereum	50
3.5.1	Ricompense Liquidity Provider	51
3.6	Scopo della tesi	51
3.6.1	Fase di download	52
3.6.2	Fase di analisi	52
4	Implementazione	54
4.1	Strumenti utilizzati	54
4.1.1	Etherscan	54
4.1.2	Selenium in Java	55
4.1.3	Libreria Guava in Java	56
4.1.4	Matplotlib in Python	56
4.2	Implementazione Web Scraper	57
5	Risultati	68
5.1	Dataset	69
5.1.1	Dataset D1	69
5.1.2	Dataset D2	70
5.2	Utenti	71
5.2.1	Analisi Mittenti	76
5.2.2	Analisi dei Destinatari	76
5.2.3	Sender giornalieri e settimanali	79

5.2.4	Receiver giornalieri e settimanali	80
5.3	Transazioni	83
5.3.1	Numero transazioni giornaliere e settimanali	83
5.4	Function transazioni	85
5.5	Ammontare Tokens YUP	88
5.6	Token Transferred	90
5.6.1	Campo From	91
5.6.2	Campo To	94
5.6.3	Campo For	96
5.7	Function delle transazioni	100
5.8	Address FROM e TO delle transazioni	104
5.9	Tipi delle transazioni	107
6	Conclusioni e Lavoro Futuri	109
6.1	Lavori futuri	111

Capitolo 1

Introduzione

Negli ultimi anni i Social Media hanno rivoluzionato il modo di comunicare delle persone divenendo una delle piattaforme più utilizzate per questo scopo. La rivoluzione non riguarda solo il modo di comunicare, ma anche il modo di fare politica, informazione e business. Negli ultimi anni, gli utenti delle piattaforme sociali sono cresciuti in maniera evidente. In particolare nel 2020 gli utenti sono incrementati del 13%, con circa mezzo miliardo di nuovi iscritti. L'aumento dell'utilizzo di queste piattaforme, specialmente negli ultimi due anni, è in parte da attribuire alla pandemia da COVID-19. Infatti dall'inizio del lockdown è aumentato il tempo speso dagli utenti online con una crescita sensibile dell'uso dei Social Media, e in particolare delle videochiamate. Al giorno d'oggi circa 3,8 miliardi di persone utilizza i Social Media e il tempo medio trascorso online è di circa 6 ore e 40 minuti al giorno¹. Inoltre, in un contesto storico caratterizzato da processi di digitalizzazione e trasformazione tecnologica, i Social Media sono diventati fondamentali anche per specifiche attività: si pensi all'*advertising*, al *crowdfunding* e al *marketing*. Con l'espressione **Social Media** si identificano tutti i mezzi di comunicazione che rendono possibile creare, condividere e scambiare contenuti generati dagli utenti attraverso l'uso di piattaforme *web-based*. Per *web-based* si intende un programma accessibile da un qualsiasi browser e che quindi non necessita di alcuna installazione sui computer degli utilizzatori.

La rapida espansione dei Social Media ha portato varie problematiche legate al loro utilizzo. Tra le più importanti, possiamo citare i problemi relativi alla privacy dei dati degli utenti, i

¹<https://www.rebelstudio.it/2020/10/16/luniverso-dei-social-media/>

quali spesso sono venduti a terze parti e/o vengono messi a rischio in seguito ad attacchi alle piattaforme. L'esempio più eclatante e più rappresentativo è Facebook, ed uno dei casi che ha generato più scalpore è stato quello di *Cambridge Analytica*², una società di consulenza britannica. Quest'ultima ha raccolto i dati personali di 87 milioni di account facebook e li ha usati per scopi di propaganda politica. *Cambridge Analytica* è infatti nota per scandali connessi alla gestione dei dati per influenzare le campagne elettorali. Il metodo utilizzato combinava il data mining, l'intermediazione dei dati e l'analisi dei dati con la comunicazione strategica per la campagna elettorale. Tramite la combinazione di queste discipline con gli studi della psicometria, era in grado di sfruttare il profilo psicologico degli utenti per individuarne una precisa personalità e andare a colpire le paure e le debolezze di questi individui attraverso una campagna elettorale costruita tramite messaggi estremamente specifici.

Un'ulteriore problema presente consiste nella presenza di informazioni false (fake news), ovvero notizie prive di fondamento. Anche la censura rappresenta un problema estremamente attuale. Ogni piattaforma ha delle regole su quanto è lecito pubblicare e queste politiche sui contenuti possono variare anche in base alle linee guida dettate dai governi nazionali. Questo è pericoloso poiché possono essere censurati contenuti con la sola colpa di essere in contrasto con il pensiero politico nazionale, e potrebbe causare una limitazione della libertà di espressione.

Le più note piattaforme Social Media, come Youtube e Facebook, sono basate su architetture centralizzate. La centralizzazione dei dati è considerata un punto debole dei Social Media, che amplifica il problema della privacy, in quanto i dati degli utenti che ne usufruiscono vengono raccolti in server di proprietà della società che fornisce il servizio sociale. Inoltre, comporta problematiche relative alla scalabilità conseguenze legate alla scalabilità. Per questo motivo, ricercatori e non solo hanno proposto soluzioni alternative basate sulla decentralizzazione dei servizi sociali. Nascono così i Decentralized Online Social Networks (DOSNs). Queste piattaforme si basano principalmente su reti Peer-to-Peer (P2P) e rappresentano un sistema distribuito senza o con minime dipendenze limitate da qualsiasi infrastruttura centrale [7]. Successivamente, grazie all'introduzione della tecnologia Blockchain, le DOSNs sono evolute in quelle che definiamo oggi come Blockchain Online Social Media (BOSMs). Con questo

²<https://www.theguardian.com/news/series/cambridge-analytica-files>

termine si fa riferimento alle attuali Social Media che utilizzano la tecnologia blockchain per fornire servizi sociali e per permettere agli utenti di condividere contenuti di valore. Il vantaggio significativo di tali piattaforme è che offrono crittografia end-to-end per ogni interazione consentendo alle persone di avere, potenzialmente, maggiore privacy e controllo sulle proprie informazioni.

Un’innovazione delle BOSMs è quella che riguarda la premiazione dei propri utenti per le loro azioni sociali all’interno della piattaforma. Nelle BOSMs gli utenti possono pubblicare i loro contenuti ai quali, a seguito di un comprovato impatto sociale, potrebbe essere associata una ricompensa economica. Questi sistemi di ricompensa spesso misurano l’impatto sociale di un contenuto tramite il numero di feedback positivi (“mi piace”/upvote) che il contenuto riceve. L’autore del contenuto premiato viene ricompensato dalla piattaforma tramite la criptovaluta generalmente definita apposta. Per incentivare una valutazione ragionata dei contenuti, anche i cosiddetti curatori (ovvero gli utenti che per primi scoprono un contenuto che diventa virale) vengono ricompensati, seppur in maniera più contenuta. In questo sistema, al contrario delle piattaforme centralizzate, sono gli utenti che decidono quali sono i contenuti di qualità o meno. La particolarità di queste piattaforme è quindi l’introduzione di un aspetto economico che affianca l’attività sociale, e che viene rappresentato attraverso l’introduzione di token sociali, ovvero delle criptovalute ottenute all’interno di piattaforme sociali.

Oltre ai citati sistemi di ricompensa, le BOSMs introducono altri vantaggi, come l’eliminazione della censura e la possibilità di poter contrastare notizie false. Inoltre, l’utilizzo della blockchain come registro in cui salvare le informazioni sull’attività degli utenti all’interno delle BOSMs rende difficile modificare, hackerare, o manomettere irrimediabilmente il sistema. Ma ovviamente esistono anche degli svantaggi, ovvero la scalabilità a causa della struttura decentralizzata, il problema della moderazione del contenuto, e la comparsa di problemi dovuti alla difficoltà nel verificare l’identità degli utenti. Negli ultimi 5 anni, il numero di BOSMs è incrementato notevolmente, così come le blockchain utilizzate. La BOSM ad oggi più nota è Steemit³ [11, 10], con circa 1.5 milioni di utenti iscritti. Steemit si basa sulla blockchain Steem, progettata specificatamente per lo sviluppo e la gestione di un social media. Tra le altre BOSMs, possiamo menzionare Minds, BitClout, Hive Blog e PeakD. Purtroppo, il loro successo

³<https://steemit.com/>

risulta ancora limitato se paragonato alle classiche OSNs. Questo potrebbe essere imputabile alla difficoltà di modificare le abitudini delle persone. Una novità in questa direzione è rappresentata dalla piattaforma Yup, la quale si pone come obiettivo quello di fornire una piattaforma di valutazione dei contenuti che appaiono sulle comuni OSNs, come ad esempio Twitter o YouTube, premiando economicamente i creatori dei contenuti a più alto impatto sociale. L'idea alla base di Yup è innovativa nello scenario delle BOSMs. Infatti a differenza della maggior parte delle precedenti BOSMs, non si propone come un'alternativa alle principali piattaforme sociali già esistenti, bensì come un'integrazione con queste ultime. L'utente può infatti utilizzare Yup semplicemente scaricando una estensione sul proprio browser che gli permette di iniziare a curare (valutare) contenuti e ricevere in cambio dei token come ricompensa, continuando ad utilizzare gli stessi social media che utilizzava precedentemente. E' fondamentale come, da un punto di vista anche solo psicologico, l'individuo non sia "costretto" ad abbandonare le sue piattaforme preferite. Attualmente quest'ultima risulta essere una delle dApp sociali più utilizzate, con più di 6,000 utenti attivi giornalmente⁴.

Tuttavia, l'introduzione dell'aspetto economico sulle BOSMs ha un forte impatto sull'attività sociale, troppo spesso guidata dal semplice guadagno [11, 10]. Questo problema è molto più evidente ed importante in una piattaforma come Yup, dove l'attività sociale delle classiche OSNs potrebbe essere condizionata da una possibile ricompensa economica. Attualmente, Yup fornisce la ricompensa sotto forma della sua criptovaluta, denominata YUP. Questo token, implementato sulla blockchain di EOS, può essere utilizzato come meglio si crede, ad esempio dando delle mancine a dei creatori di contenuti che sono ritenuti particolarmente validi. Un altro aspetto estremamente innovativo di Yup è l'adozione di un approccio multichain. Infatti Yup, oltre ad implementare la sua logica, inclusa la registrazione dei voti e la distribuzione delle ricompense, su EOS, offre anche la possibilità di fare bridging dei propri token sulla blockchain di Ethereum. In questo modo gli utenti possono "trasferire" i propri token tra le due blockchain. Avere accesso alla blockchain di Ethereum permette di allargare immensamente i propri orizzonti, specialmente sotto il punto di vista economico. Tramite l'ecosistema formato intorno ad Ethereum è infatti possibile accedere ad un vastissimo mercato di scambio token ed investimenti. Questo significa che un individuo potrebbe intraprendere attività fi-

⁴<https://www.dapp.com/app/yup>

nanziarie con il token YUP, tramite altre blockchain, al di fuori dell’ambito sociale per cui Yup si propone. Inoltre, l’attività sociale degli utenti, inclusa la creazione e la curazione di contenuti, potrebbe essere svolta con il solo obiettivo di massimizzare la ricompensa, e quindi il guadagno economico, portando ad una situazione di socialità artificiale.

Lo scopo di questa tesi consiste nell’analizzare l’utilizzo del token sociale YUP, criptovaluta della piattaforma Yup, al fine di comprendere quanto l’economia digitale stia prevalendo sulle attività sociali che dovrebbero rappresentare il fulcro di queste piattaforme. Questo tipo di analisi, ci ha permesso di valutare quanto le persone che stanno attualmente utilizzando Yup sono coinvolte nel mondo cripto. In particolare, ci siamo concentrati maggiormente sui token circolanti sulla blockchain di Ethereum perché sono quelli che più prontamente possono essere utilizzati per effettuare azioni di investimento, ed entrare in un mercato economico estremamente variegato.

Per conseguire l’obiettivo finale, abbiamo organizzato il lavoro in due fasi distinte: una fase di download dei dati ed una fase di analisi di questi ultimi. Durante la fase di download, illustrata nel Capitolo 4, abbiamo collezionato le informazioni relative alle transazioni YUP presenti su Ethereum. Successivamente abbiamo recuperato ulteriori informazioni relative ai trasferimenti di token ERC20 presenti sulla stessa blockchain, in modo da comprendere ulteriormente le meccaniche nascoste dietro alle operazioni svolte dagli utenti di Yup su Ethereum. Nella fase di analisi, descritta nel Capitolo 5, ci siamo concentrati, come detto precedentemente, sull’analisi delle transazioni svolte dagli utenti di Yup e presenti su Ethereum per via dello YUP Bridge, delineato nel Capitolo 2; cercando di comprendere anche la frequenza giornaliera e settimanale con cui vengono svolte determinate operazioni.

Questa tesi è organizzata come segue. Nel Capitolo 2 viene presentato lo stato dell’arte inerente al campo delle Blockchain Online Social Networks ed una panoramica della tecnologia Blockchain, in particolare viene mostrata la blockchain Ethereum elencando le sue caratteristiche, funzionalità e problematiche. Infine vengono descritti i sistemi decentralizzati, e in particolare Uniswap e gli exchanger particolarmente ricorrenti su Ethereum. Nel Capitolo 3 illustreremo la piattaforma Yup e lo scopo di questa tesi, andando a delineare le due fasi di cui si compone. Nel Capitolo 4 vengono introdotti gli strumenti utilizzati per la fase di download dei dati fondamentali per le nostre analisi. Nel Capitolo 5 viene presentata l’analisi condot-

ta sulle transazioni YUP presenti nella blockchain di Ethereum. Concludendo, il Capitolo 6 è dedicato a un compendio generale sul lavoro svolto e sui possibili sviluppi utili per il suo proseguimento.

Capitolo 2

Stato dell'arte

In questo Capitolo intordurremo le nozioni fondamentali relative alla tecnologia blockchain e alla piattaforma Yup, oggetto del nostro lavoro. Inizieremo descrivendo cos'è una Blockchain e le differenze tra quest'ultima e i Distributed Ledger Technologies (DLT) [15]. Proseguiremo descrivendo la struttura delle blockchain e dei suoi blocchi, e il meccanismo di validazione di questi ultimi. Successivamente parleremo di Ethereum, delineando le sue caratteristiche e il suo funzionamento, proseguendo poi con la descrizione del formato delle transazioni. Concluderemo il Capitolo introducendo alcuni concetti riguardo la Finanza Decentralizzata (DeFi), gli Exchange Decentralizzati (DEX), come Uniswap, e la problematica Miner Extractable Value (MEV) ad essi collegati.

2.1 Blockchain

La **Blockchain** [9] è una struttura dati condivisa e immutabile. E' definita come un registro digitale le cui voci sono raggruppate in blocchi. Questi ultimi sono concatenati in ordine cronologico e la loro integrità è garantita dall'uso della crittografia. In particolare, la struttura a catena è garantita dal fatto che ogni blocco deve contenere l'hash del blocco precedente. I blocchi di una blockchain contengono insiemi di transazioni confermate ed ogni blocco è collegato al precedente tramite il suo hash, formando così la catena. Tra i blocchi si viene quindi a creare un ordine, anche chiamato *altezza*, dove il primo blocco ha altezza 0, il successivo ha altezza 1 e così via. Sebbene la dimensione di una blockchain sia destinata a crescere nel

tempo, essa è immutabile poiché il suo contenuto, una volta scritto, non è più né modificabile, né eliminabile, a meno di non invalidare l'intera struttura.

La creazione dei blocchi che compongono la blockchain è delegata a dei nodi particolari, chiamati miner, che generalmente si offrono di partecipare al processo. La partecipazione a questo processo prevede degli obblighi, tra cui controllare che le transazioni da inserire nei blocchi siano ben formate, e dei benefici, come ottenere delle ricompense economiche derivate dalla creazione del blocco. Dato che il processo di creazione di blocchi è distribuito tra l'insieme di miner, talvolta risulta possibile che alcuni nodi della rete producano simultaneamente più blocchi concorrentemente, ossia non collegati in sequenza tra di loro, ma tutti collegati ad uno stesso blocco già esistente. Ciò dà origine ad una cosiddetta biforcazione (*fork*) nella catena. Il selezionamento del blocco da accettare tra quelli concorrenti, e quindi quale delle possibili catene che si formano è da seguire, avviene tramite un protocollo che generalmente favorisce il primo blocco che viene ricevuto dal nodo. Nel caso in cui uno dei rami differenti cresce più velocemente, i nodi invece considereranno come principale quel ramo e cercheranno di estenderlo. In Figura 2.1 è mostrata una rappresentazione grafica della blockchain, con il blocco di genesi (blocco verde), ovvero il primo blocco della catena, l'unico a non avere un blocco precedente, gli altri blocchi della catena principale (i blocchi neri). Nel caso di una fork, alcuni blocchi vengono creati "a vuoto", anche chiamati orfani (blocchi viola in figura), che non entreranno a far parte della blockchain e le transazioni contenute non sono da considerare valide.

La blockchain è solo una delle possibili implementazioni di DLT, ossia sistemi che si basano su un registro contabile distribuito che può essere letto e modificato da più nodi di una rete. Le caratteristiche che accomunano i sistemi sviluppati con le Blockchain e con Distributed Ledger sono i seguenti [8]:

1. Digitalizzazione dei dati
2. Decentralizzazione
3. Disintermediazione
4. Tracciabilità e programmabilità dei trasferimenti

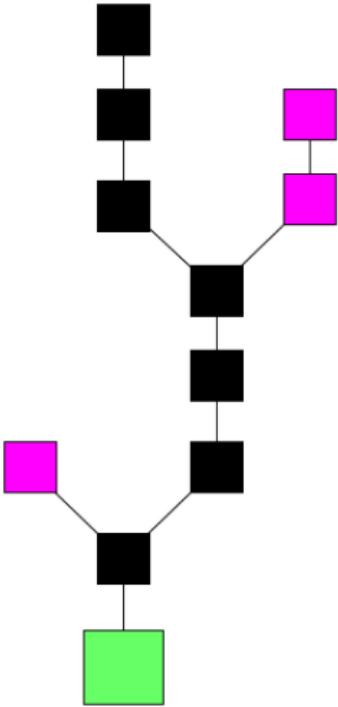


Figura 2.1: Rappresentazione della blockchain

5. Trasparenza e verificabilità

6. Immutabilità del registro

Spesso si fa confusione tra i concetti di blockchain e DLT. Sia in un DLT che in una blockchain tutti i nodi della rete mantengono una copia del registro e possono accedervi per operazioni di lettura o scrittura. Nel caso delle blockchain le modifiche al registro sono regolate tramite algoritmi di consenso [21] che permettono la regolarizzazione tra le varie versioni del registro presenti sulla rete. L'utilizzo di algoritmi di consenso uniti ad un ampio uso della crittografia, hanno lo scopo di garantire la sicurezza e l'immutabilità del registro. Per questi motivi, la blockchain è considerata una valida alternativa in termini di sicurezza, affidabilità e trasparenza ai registri gestiti in maniera centralizzata. La tabella in Figura 2.2 riporta le principali differenze tra la blockchain e il DLT.

CARATTERISTICA	BLOCKCHAIN	DLT
STRUTTURA BLOCCO	Dati rappresentati come catena di blocchi	Dati rappresentati con qualsiasi tipo di struttura
SEQUENZA	Tutti i blocchi seguono una specifica sequenza	Diversi tipi di DLT hanno diverse sequenze
CONSUMO ENERGETICO	Alto consumo dovuto agli algoritmi di consenso	Non è necessario alcun tipo di consenso perciò il consumo è basso
TOKENS	Differenti tipi di tokens e valute presenti in differenti tipi di blockchain	Non c'è necessità di alcun token o valuta

Figura 2.2: Tabella Differenze DLT e Blockchain

2.1.1 Decentramento

La blockchain sfrutta una rete distribuita per fare in modo di memorizzare i dati su i suoi nodi, ed evitare di avere una centralizzazione che potrebbe essere sfruttata per abbattere l'intero sistema. Tra i metodi di sicurezza, la blockchain sfrutta molto la crittografia a *chiave pubblica*. Una chiave pubblica rappresenta un indirizzo che corrisponde ad un nodo sulla blockchain, e i token inviati nella rete vengono registrati come appartenenti ad una chiave pubblica. La *chiave privata*, controparte della pubblica, fa come da password e permette al suo proprietario di accedere alle sue risorse digitali e di interagire con le funzionalità della blockchain. Ogni nodo partecipante mantiene una copia della blockchain permettendo un grado di ridondanza dei dati massivo, il che rende estremamente difficile causare la perdita di tutte le copie della blockchain. I nodi miner validano le nuove transazioni e le aggiungono al blocco che stanno costruendo dopo aver verificato l'intera blockchain. Una volta completato il blocco, lo trasmettono agli altri nodi della rete.

2.1.2 Struttura del blocco

Dato che la validazione di una singola transazione potrebbe essere molto costosa computazionalmente, le transazioni sono generalmente raggruppate nei blocchi della blockchain. Il numero di transazioni all'interno di ognuno di questi blocchi varia in base alla dimensione della transazione stessa poiché in genere i blocchi hanno una dimensione massima. Invece

la dimensione della transazione varia in base al numero di input e di output della stessa. Un blocco è composto da due parti principali: l'**header** e il **body**. Le transazioni sono racchiuse nel body del blocco, mentre nell'header sono presenti sette campi di gestione del blocco stesso, descritti di seguito:

- **Versione** dipende dalla versione del software utilizzato;
- **Hash del blocco precedente (PrevHash)** è un hash che serve per fare riferimento al precedente blocco della catena;
- **Merkle root** è l'hash di tutti gli hash di tutte le transazioni nel blocco;
- **Timestamp** è il time stamp dell'ultima transazione ;
- **Nonce**, presente nelle blockchain basate su Proof of Work, è un valore che viene aggiunto al blocco in modo che l'output della funzione di hash vari facendo in modo che risulti inferiore al valore target;
- **Numero di transazioni** presenti nel blocco.

2.1.3 Validazione dei blocchi

I miner, per poter svolgere il loro lavoro, devono inizialmente ottenere una copia della blockchain da altri nodi e verificarne l'integrità. Per essere certi di aver ottenuto il ramo più lungo attualmente attivo, è raccomandato chiedere l'intera struttura a più nodi. Una volta raggiunto l'ultimo blocco creato, i miner iniziano a collezionare le nuove transazioni generate dai nodi della rete, ancora non validate, e, secondo un protocollo di consenso stabilito, suggeriscono alla rete quale dovrebbe essere il nuovo blocco. I protocolli di consenso più utilizzati possono variare, ma in genere si basano sulla possibilità di dimostrare di avere una certa risorsa.

Ad esempio, nel caso del protocollo PoW, i miner devono essere in grado di dimostrare di poter risolvere un puzzle matematicamente difficile, e quindi di dedicare una grande potenza computazionale alla blockchain. Molte blockchain PoW come puzzle adottano l'inversione di una funzione hash crittografica e hanno come obiettivo quello di trovare un valore, chiamato nonce, tale per cui l'applicazione della funzione hash ritorni un risultato che inizia con un

dato numero di bit pari a zero. Il numero di bit richiesto per poter ritenere un blocco valido viene anche chiamato "difficoltà" del puzzle: dato che il miglior processo risolutivo per questo tipo di puzzle crittografici è quello di andare a tentativi, sarà solo un caso riuscire a risolverlo e l'unico modo per avere più chance di risolverlo è semplicemente avere a disposizione più potenza computazionale per poter provare più nonce. Il primo miner che risolve il puzzle trasmette il blocco, contenente la soluzione, nella rete dove verrà controllato dagli altri nodi, miner inclusi, e se il blocco viene ritenuto ben formato, verrà accettato come blocco successivo nella catena. Questo crea una competizione tra i miner che porta ad un grande spreco di risorse computazionali dato che per la creazione del blocco successivo i miner devono includere l'hash dell'ultimo blocco minato, generando quindi un puzzle nuovo. Esempi di blockchain che utilizzano PoW sono Bitcoin (funzione hash di riferimento: SHA-256) ed Ethereum (funzione hash di riferimento: SHA-3).

Un altro protocollo di creazione di blocchi e di consenso molto utilizzato è il **Proof-of-Stake** (PoS). Il protocollo PoS sfrutta il concetto di validatori, che impegnano (mediante staking) la propria criptovaluta, per prendere parte al processo di creazione di blocchi. Un validatore è scelto a caso per creare nuovi blocchi in base alla quantità di criptovaluta impegnata. Questo è un sistema che sposta la competizione dalla potenza computazionale alla quantità di valuta posseduta, portando ad un processo dove viene sprecata pochissima energia. Questo protocollo è stato creato per risolvere alcuni problemi con del PoW come:

- **Mancanza di scalabilità e velocità:** Il processo di mining aggiunge un alto livello di latenza per poter approvare le transazioni e produrre nuovi blocchi. Con PoS questa situazione viene evitata. Nelle blockchain che utilizzano il protocollo PoS, le verifiche vengono eseguite da nodi con un'elevata holding di monete. In questo modo le verifiche avvengono velocemente, impattando positivamente sulla scalabilità e velocità della rete.
- **L'elevato consumo energetico del processo di mining:** Il processo di mining in PoW richiede un'elevata potenza di calcolo, e di conseguenza un elevato consumo d'elettricità. Ma PoS cambia questa visione, trasformando il processo di mining in un processo partecipativo. Una partecipazione riflessa nella detenzione di monete o di tempo all'interno della rete. Infatti questo sistema cerca d'incoraggiare i partecipanti ad avere sempre una

certa quantità di monete. Il possesso di monete gli consente d'essere scelti dal processo di selezione casuale che viene effettuato per assegnare i compiti. Secondo questo schema, chi ha più riserve, ha maggior peso nella rete e maggiori possibilità di essere eletto. Una volta scelti, possono convalidare le transazioni e creare nuovi blocchi, in modo da ricevere guadagni e incentivi per il lavoro svolto.

- **Il decentramento della rete:** Questo è un problema che colpisce fortemente le reti PoW e centralizza la rete nelle mani di pochi. PoS cerca di risolvere questo problema, diversificando e democratizzando l'accesso dei partecipanti ai diversi compiti della rete.
- **Previene gli attacchi:** Gli attacchi sono una delle paure nelle reti PoW. È sufficiente che un mining pool abbia il 51% della potenza di calcolo per dargli la potenza necessaria per manipolare la blockchain. Ma in un sistema PoS, questo è possibile solo se l'attaccante possiede il 51% di tutte le monete. Se l'attaccante effettua un tale attacco, il valore della moneta diminuirebbe repentinamente, e ciò porterebbe ad ingenti perdite economiche per l'attaccante. Questa situazione è utilizzata per prevenire questi attacchi e mantenere la sicurezza della rete.

Ethereum ha annunciato da tempo di voler abbandonare il proprio schema basato su PoW per addentrarsi nella PoS, mentre altre blockchain come EOS [22] lo hanno già adottato. Esistono poi blockchain come quella di Steem [11] che ne hanno adottato una variante che si ispira alla democrazia rappresentativa, dove i possessori di criptomoneta possono eleggere i loro rappresentanti.

2.2 Ethereum

Ethereum [6] è una blockchain che supporta il Web 3.0 grazie alla sua decentralizzazione e, in particolar modo, agli smart contracts [6, 2]. Gli smart contract di Ethereum sono pezzi di codice la cui esecuzione viene garantita dalla blockchain stessa e sono scritti in un linguaggio di programmazione Turing-completo. Per poter girare sulla rete peer-to-peer, i contratti di Ethereum "pagano" l'utilizzo della sua potenza computazionale tramite una unità di conto, detta **Ether**, che funge quindi sia da criptovaluta sia da "carburante". Di conseguenza Ethe-

reum non è solo un network per lo scambio di valore monetario, ma una rete per l'esecuzione di contratti basati su Ethereum. L'esecuzione degli smart contract avviene nella cosiddetta Ethereum Virtual Machine (**EVM**) [12] ovvero una macchina virtuale sul cui stato è condiviso dai nodi della rete Ethereum. Chiunque partecipi alla rete Ethereum ha una copia dello stato della EVM. Ogni nodo può trasmettere una richiesta per eseguire calcoli sulla EVM sotto forma di transazione Ethereum, la quale corrisponde all'esecuzione di una funzione di uno smart contract. Ogni volta che una richiesta viene trasmessa, i miner di Ethereum verificano, convallidano ed eseguono il calcolo inserendo la transazione in un blocco. Questo provoca un cambio di stato dell'EVM, che viene salvato e propagato a tutta la rete. Una volta che le transazioni sono state verificate come valide e aggiunte alla blockchain, non possono essere modificate o invertite in futuro, grazie ai meccanismi crittografici introdotti in Sezione 2.1.

2.2.1 Account in Ethereum

Un account Ethereum è un'entità identificata dalla sua chiave pubblica che possiede saldo in ether (ETH) che può inviare transazioni su Ethereum. Ethereum ha due tipi di account:

- **Personale:** controllato da chiunque possiede la chiave privata. La creazione di un account non costa nulla e ogni account può avviare transazioni.
- **Contratto:** ovvero uno smart contract distribuito sulla rete Ethereum. La creazione di un account ha un costo poiché viene utilizzato spazio di archiviazione della rete (il codice deve essere salvato sulla blockchain per garantire la trasparenza della sua esecuzione). Questi account possono inviare transazioni solo in risposta alla ricezione di una transazione. Le transazioni da un account esterno a un account contratto possono attivare codice, che a sua volta può eseguire azioni come il trasferimento di token o anche la creazione di un nuovo contratto.

Entrambi questi account hanno la possibilità di ricevere, conservare e inviare ETH e token e possono anche interagire con gli smart contract distribuiti.

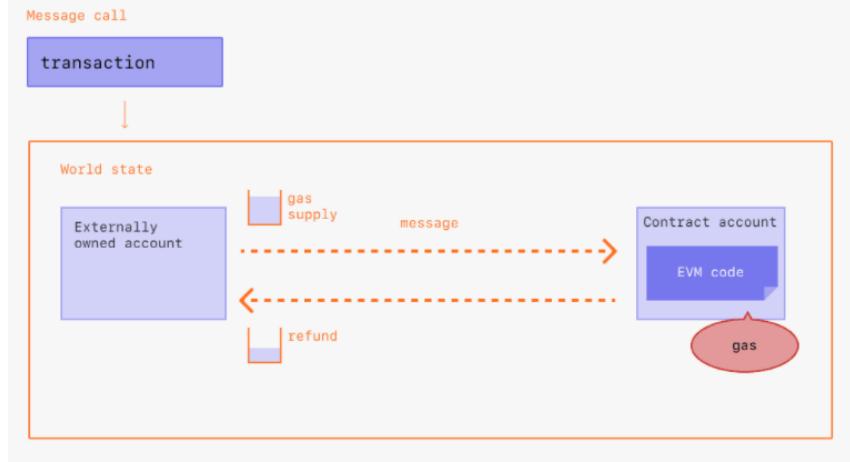


Figura 2.3: Esecuzione di una transazione

2.2.2 Transazioni

Con transazione Ethereum ci si riferisce ad un’azione iniziata da un account personale, ovvero gestito dall’uomo e non da un contratto. Le transazioni, che cambiano lo stato dell’EVM, devono essere trasmesse all’intera rete. Ogni nodo può trasmettere una richiesta di esecuzione di una transazione sull’EVM. In seguito, un miner eseguirà la transazione e propagherà il cambiamento di stato risultante al resto della rete. Dato che il linguaggio per la scrittura degli smart contract è Turing-completo, e anche per incentivare un utilizzo ragionato della potenza di calcolo, le transazioni richiedono una commissione che in gergo viene chiamata *gas*. In particolare per ogni transazione viene specificato il *gas limit* e il *gas price*, i quali determinano la commissione massima che si è disposti a pagare per la transazione. La commissione in gas è proporzionale alla quantità di calcolo che il miner deve fare per eseguire la funzione dello smart contract invocata tramite la transazione, e il gas non utilizzato viene rimborsato sull’account dell’utente. In Figura 2.3 mostriamo uno schema di una transazione tra un account esterno e un account contratto e lo scambio di gas effettuato. Una volta inviata una transazione, viene generato un hash crittografico, racchiuso nel campo *transaction_hash* descritto nella Sezione 2.3 che servirà ad identificarla univocamente. La transazione viene poi inviata alla rete e inclusa in un gruppo di altre transazioni da confermare, chiamato *mempool*. I miner si occupano di controllare che le transazioni siano ben formate e, successivamente, sceglierle, eseguirle ed includerle in un blocco. Solo quando una transazione appare in un blocco può

essere considerata *riuscita*. Per ricompensarli del loro lavoro, le commissioni delle transazioni inserite in un blocco vengono date al miner che ha creato quel blocco. Per questo motivo, i miner danno priorità alle transazioni con un *gas price* più elevato poiché otterranno una commissione di conseguenza. La transazione avrà anche un numero di conferme, ovvero il numero di blocchi creati successivi al blocco che include la transazione. Più è alto il numero, maggiore è la probabilità che la transazione possa essere considerata definitiva.

Tipologie transazioni in Ethereum

In Ethereum al momento sono presenti due tipologie di transazioni:

- **0 (Legacy)**: rappresenta le transazioni canoniche di tipo 0, che sarebbe la vecchia tipologia standard delle transazioni
- **2 (EIP-1559)**: rappresenta la nuova tipologia introdotta di recente.

Con la *Ethereum Improvement Proposal (EIP)* numero 1559, è stato introdotto il tipo di transazione **2 (EIP - 1559)**. Con la **EIP - 1559** viene modificato il meccanismo di determinazione delle commissioni che vengono pagate per ogni transazione eseguita su Etherum.

Le differenze fondamentali tra i due tipi di transazione vengono rappresentanti nella figura 2.4. Con il tipo 0, il metodo utilizzato per determinare le tasse di transazione è il *first-price auction*, ovvero gli utenti fanno un'offerta per ottenere dello spazio e includere la propria transazione nel blocco successivo. Questo avviene definendo il massimo gas price che gli utenti sono disposti a pagare per vedere la propria transazione validata dai miner. Questo gas è rappresentato nella figura 2.4 dalle **FEES**. Come abbiamo visto nella Sezione 2.2.2, le transazioni, per essere convalidate, devono essere "minate" dai miner, i quali ottengono in cambio le fees offerte per l'esecuzione della transazione. Con questa tipologia standard, essendo la dimensione di un blocco limitata, si genera una competizione per lo spazio al suo interno, e gli utenti che fanno le offerte più alte, vedono le proprie transazioni scelte per prime. Nei momenti di congestione della rete, questo meccanismo tradizionale può risultare poco efficiente per gli utenti, e accade spesso di dover pagare tasse molto elevate anche per transazioni semplici.

La **EIP-1559** propone di migliorare questo aspetto della tipologia standard, e si pone alcuni obiettivi fondamentali:

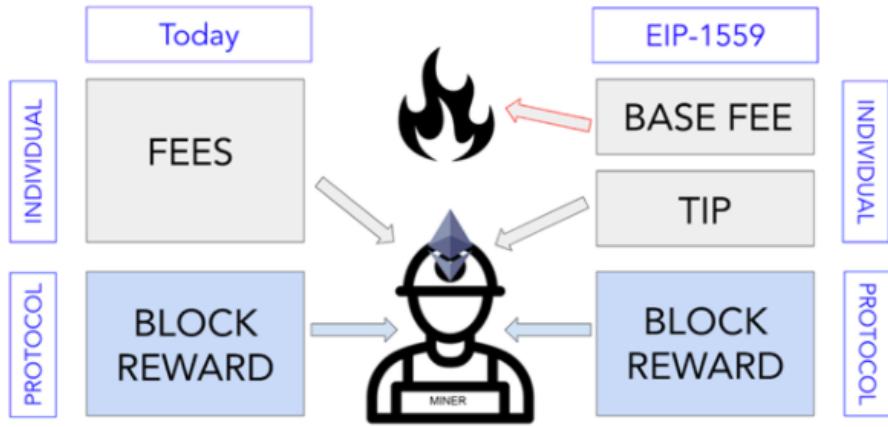


Figura 2.4: 0 (Legacy) vs 2 EIP-1559

1. Rendere le commissioni più facilmente prevedibili
2. Ridurre il tempo di conferma delle transazioni
3. Automatizzare il sistema di calcolo delle tasse

Il cambiamento principale avviene con il metodo di funzionamento del gas, raffigurato sempre attraverso la figura 2.4. L’utente non deve pagare tutto il gas ai minatori, infatti una parte del gas viene bruciata. Nella figura 2.4, possiamo notare che il gas è diviso in **BASE FEE** e **TIP**. **BASE FEE** rappresenta la parte di gas che viene bruciata, mentre **TIP** rappresenta la ”mancia” che viene guadagnata dai miner per il loro lavoro. Il prezzo del gas viene quindi sostituito con l’introduzione di due nuovi campi, ovvero *”tariffa max prioritaria per gas”* e *”tariffa max per gas”* descritti nella Sezione 2.3. Viene introdotto un nuovo concetto, cioè il costo base del gas, ovvero un valore fisso che cambia gradualmente a seconda del carico di rete (se il numero delle transazioni previste è basso, il costo base diminuisce gradualmente; appena diventano di più, aumenta gradualmente). Ogni volta che viene inviato un nuovo tipo di transazione, per il calcolo del gas viene sempre usato il costo base del gas moltiplicato per la quantità di gas consumato. Questa innovazione aiuta ad eliminare in parte la competizione tra gli utenti che desiderano effettuare transazioni e, di conseguenza, anche il problema legato al pagamento elevato di gas per transazioni. Infatti, mentre prima le commissioni che si era disposti a pagare dipendevano interamente dall’utente che doveva conoscere il costo tipico in gas di una

transazione, con la nuova tipologia il costo è più facilmente prevedibile. I cambiamenti principali per gli utenti sono la possibilità di modificare la tassa del gas per aumentare o diminuire l'eventuale cifra da pagare ai miner. Inoltre durante i periodi di alta congestione della rete, gli utenti avranno consapevolezza dello stato del network in base a quanto è alta la tassa proposta. Se la rete è troppo congestionata, l'utente deciderà di aspettare un momento meno critico. Una conseguenza significativa per i miner invece è la diminuzione dei loro guadagni derivati dalle commissioni pagate dagli utenti. Infatti, se con la tipologia standard i minatori ottengono gli ETH emessi ad ogni blocco, definiti nella figura 2.4 come **BLOCK REWARD**, e l'intero ammontare della tassa delle transazioni contenute nel blocco validato, con EIP-1559 le tasse **BASE FEE** vengono bruciate e i miner ottengono solo il **BLOCK REWARD** e le **TIP** degli utenti. Infine, con l'introduzione di EIP-1559 e il conseguente meccanismo per cui le tasse vengono bruciate, si ha una progressiva riduzione del numero di Ether in circolazione. Infatti, la EIP-1559 è progettata per stabilire un circolo virtuoso tra l'attività on-chain e il prezzo dell'asset ETH, poiché più transazioni vengono convalidate e maggiore è la quantità di Ether bruciati, diminuendo così di conseguenza la *circulating supply*¹ della moneta e aumentandone potenzialmente il valore.

2.2.3 Ether

Lo scopo di **Ether** è consentire l'esistenza di un mercato per il calcolo. Questo mercato costituisce un incentivo economico per i partecipanti per verificare ed eseguire le richieste di transazione e per fornire risorse di calcolo alla rete. Ogni partecipante che trasmette una transazione deve anche offrire un certo quantitativo di ether alla rete, come ricompensa da assegnare a chi poi si occuperà di verificare la transazione, eseguirla, salvarla nella blockchain e trasmetterla alla rete. La quantità di ether pagati varia in funzione alla lunghezza del calcolo. Questo impedisce che partecipanti malevoli intasino intenzionalmente la rete inviando richieste di cicli infiniti o script con alta richiesta di risorse, perché verrebbero addebitati in continuazione. Di seguito procediamo con la descrizione delle tre azioni che possono essere effettuate sul ether.

¹numero di monete di una criptovaluta che sono pubblicamente disponibili e circolano nel mercato

Minting Ether

Il **Minting** è il processo in cui viene creato un nuovo ether sul Ethereum ledger. Il protocollo Ethereum sottostante crea il nuovo ether e non è possibile per un utente crearlo. L'ether viene coniato quando un miner crea un blocco sulla blockchain di Ethereum. Come incentivo per i minatori, il protocollo garantisce una ricompensa in ogni blocco, incrementando il saldo di un indirizzo impostato dal minatore del blocco.

Burning Ether

Oltre a creare ether attraverso ricompense di blocco, l'ether può essere distrutto da un processo chiamato **"burning"**. Quando l'ether viene bruciato, viene rimosso dalla circolazione in modo permanente. Questo processo di eliminazione dell'ether si verifica in ogni transazione su Ethereum. Infatti, quando gli utenti pagano per le loro transazioni, la loro tariffa base per il gas viene distrutta dal protocollo. A seconda della domanda di rete, alcuni blocchi bruciano più ether di quanti ne creano.

Trasferring Ether

Ogni transazione su Ethereum contiene un campo valore, che specifica la quantità di ether da trasferire dall'indirizzo del mittente all'indirizzo del destinatario. Quando l'indirizzo del destinatario è uno smart contract, questo ether trasferito può essere utilizzato per pagare il gas quando lo smart contract esegue il suo codice.

2.2.4 Smart contract

Uno **Smart Contract** è un programma eseguito sulla blockchain di Ethereum, ovvero è una raccolta di codice (le funzioni) e dati (lo stato) che risiede a un indirizzo specifico sulla blockchain. Gli Smart Contract sono un tipo di account Ethereum che hanno un saldo e possono inviare transazioni in rete. Inoltre non sono controllati da un utente, ma distribuiti in rete ed eseguiti come programmato. Gli account degli utenti possono quindi interagire con uno Smart Contract inviando transazioni che eseguono una funzione definita sul contratto. Chiunque può scrivere Smart Contract e distribuirli in rete. È sufficiente sapere programmare in un linguag-

gio per Smart Contract e avere abbastanza ETH per distribuire un contratto. Distribuire uno Smart Contract è una transazione, quindi bisogna pagare gas come avviene per un semplice trasferimento di ETH. I costi del gas per la distribuzione di un contratto sono però molto più elevati. Questi contratti sono pubblici su Ethereum ed è possibile che distribuiscano altri contratti. Essi possono essere utilizzati in maniera sicura per eseguire un vasto numero di operazioni: sistemi elettorali, registrazione di nomi di dominio, mercati finanziari, piattaforme di crowdfunding, proprietà intellettuale, e così via.

2.2.5 Token ERC20

I **token ERC20** [19, 20] sono smart contracts che vengono eseguiti nella blockchain di Ethereum. L'utilità principale di questi token è standardizzare l'interfaccia per la creazione e l'emissione di nuovi token sulla rete. Ciò si ottiene applicando determinate regole e parametri per la sua accettazione. Ad esempio, per modificare o spostare un token ERC20, un utente deve avere Ether. L'obiettivo dei token ERC20 è progettare uno standard per creare interoperabilità e compatibilità tra i differenti token e promuovere miglioramenti nell'ecosistema Ethereum.

Caratteristiche dei token ERC20

Le principali caratteristiche che rendono questi token uno standard in Ethereum sono le seguenti:

1. Hanno un nome o un identificatore e un simbolo associato. Attraverso questi due valori è possibile identificare e differenziare i token l'uno dall'altro all'interno della blockchain di Ethereum.
2. È in grado di gestire gli aspetti economici di base della sua emissione. Dati come il sistema di precisione decimale e l'emissione totale sono una parte fondamentale del token nella sua struttura dati.
3. Gestisce un'interfaccia per controllare e rivedere i bilanci degli indirizzi dei suoi proprietari. In questo modo il token è in grado di riportare il saldo totale dei fondi contenuti in uno specifico indirizzo.

4. Può gestire il sistema di trasferimento in modo originario. Questo grazie al fatto che il token possiede funzioni per gestire i trasferimenti di fondi.
5. Il token è in grado di gestire autonomamente prelievi parziali di fondi da un indirizzo.

2.2.6 MEV: Miner Extractable Value

Il Miner Extractable Value (**MEV**) [5, 14] è rappresentato come il problema più serio presente su Ethereum e sulle blockchain. MEV significa *Valore estraibile dal minatore*, e si riferisce ai profitti che possono essere realizzati estraendo valore dagli utenti di Ethereum, riordinando, inserendo o censurando le transazioni all'interno dei blocchi. I miner o i produttori di blocchi sono responsabili della sequenza delle transazioni e della decisione di quali operazioni includere nei blocchi e in quale ordine. I minatori possono trarre profitto dal gioco MEV in due modi:

1. Vendendo lo scarso spazio di blocco agli estrattori MEV non minatori attraverso le *Aste prioritarie di gas* (PGA), in cambio di commissioni di transazione esorbitanti e catturando MEV direttamente attraverso il riordino.
2. Censurando le transazioni per trarre profitto dalla liquidazione a catena o dalle opportunità di arbitraggio per se stessi.

MEV coinvolge gli utenti finali che commerciano su scambi decentralizzati. Gli utenti sono la parte più sfruttata in questo gioco in quanto emettono una certa quantità di valore che può essere catturato dai minatori e dagli estrattori MEV non minatori. In teoria il MEV va interamente ai miner, perché i minatori sono l'unica parte in grado di garantire l'esecuzione di un'opportunità MEV redditizia. Ma in pratica, gran parte del MEV viene estratta da partecipanti alla rete indipendenti, quindi non minatori, denominati **"ricercatori"**. I ricercatori eseguono algoritmi complessi sui dati blockchain per rilevare opportunità MEV redditizie e dispongono di bot per inviare automaticamente tali transazioni redditizie alla rete. I due modi principali in cui i ricercatori partecipano al gioco MEV sono:

1. Offrendo prezzi esorbitanti del gas per posizionare strategicamente le loro transazioni in posizioni specifiche all'interno di blocchi dai minatori

2. Esprimendo le preferenze di ordinamento delle transazioni ai minatori utilizzando una nuova estrazione MEV e strumenti come Flashbot.

I minatori ottengono comunque una parte dell'intero importo del MEV perché i ricercatori sono disposti a pagare commissioni elevate sul gas in cambio di una maggiore probabilità di inclusione delle loro transazioni in un blocco. La tariffa del gas che un ricercatore è disposto a pagare sarà un importo fino al 100% del MEV del ricercatore. Infatti per alcune opportunità MEV altamente competitive, i ricercatori potrebbero dover pagare il 90% o anche più delle loro entrate MEV totali in commissioni sul gas al minatore. Questo perché l'unico modo per garantire che la loro transazione di arbitraggio venga eseguita è inviare la transazione con il prezzo del gas più alto. Quando individuano un'opportunità di estrazione MEV, i ricercatori analizzano la logica dietro lo scambio e creano un pacchetto, ovvero una o più transazioni raggruppate ed eseguite nell'ordine in cui vengono fornite. Questi pacchetti di transazioni dei ricercatori possono fare riferimento alle transazioni in sospeso di altri utenti nel mempool, e indirizzare blocchi specifici per l'inclusione. Una volta creato il pacchetto, un ricercatore lo invia a un miner utilizzando reti off-chain come MEV-Geth di *Flashbots*. Ciò consente loro di esprimere le proprie preferenze di ordinazione delle transazioni in modo rapido e senza rischi, direttamente ai minatori. Quando un miner include un pacchetto o una transazione nel suo blocco, il processo di estrazione MEV è completo e la transazione del ricercatore viene confermata sulla catena.

Di seguito descriveremo i tre attacchi più comuni, inerenti a questa problematica:

Front-running

Il front-running implica l'ottenimento di una transazione in prima linea nella coda di esecuzione, prima di una transazione in sospeso nota. In Ethereum, i ricercatori eseguono bot specializzati front-running che scansionano la rete alla ricerca di ordini di grandi dimensioni su scambi decentralizzati e inviano transazioni concorrenti con commissioni di gas più elevate per farli minare prima della transazione della vittima.

Attacchi Sandwich

Un attacco sandwich è un attacco in cui un trader predatore effettua due transazioni, una prima e l'altra subito dopo una transazione vittima in sospeso. I ricercatori utilizzano in genere attacchi sandwich per estrarre MEV da ignari trader su scambi decentralizzati manipolando il prezzo di un bene. Ad esempio, un trader può identificare un token che una vittima sta per acquistare e fare uno scambio per aumentare il prezzo, quindi vendere il token subito dopo che l'ordine di acquisto della vittima ha aumentato ulteriormente il prezzo.

Liquidazioni

Le liquidazioni del protocollo di prestito presentano un'altra nota opportunità MEV. I protocolli di prestito come Maker e Aave funzionano richiedendo agli utenti di depositare una sorta di garanzia (tipo gli ETH). Gli utenti possono quindi prendere in prestito diversi asset e token da altri utenti a seconda di ciò di cui hanno bisogno, fino a un certo importo della loro garanzia depositata, ad esempio il 30% (l'esatta percentuale del potere di prestito è determinata dal protocollo). Gli utenti da cui prendono in prestito gli altri token fungono da prestatori. Se, a causa delle fluttuazioni del mercato, il valore delle attività prese in prestito supera, ad esempio, il 30% del valore della loro garanzia (la percentuale esatta è determinata dal protocollo), il protocollo consente a chiunque di liquidare la garanzia, pagando istantaneamente il finanziatore. In caso di liquidazione, il mutuatario deve pagare una forte commissione di liquidazione, parte della quale va al liquidatore, ed è qui che entra in gioco l'opportunità MEV. I ricercatori competono per analizzare i dati blockchain il più velocemente possibile per determinare quali mutuatari possono essere liquidati in modo da essere i primi a presentare una transazione di liquidazione e riscuotere la commissione di liquidazione per se stessi.

Conseguenze e risoluzione

MEV porta alla congestione della rete ed esercita una pressione sul rialzo dei prezzi del gas. Destabilizza Ethereum a livello di protocollo, perché mette in discussione la finalità e l'immutabilità delle transazioni. Inoltre nessuno dei due aggiornamenti di Ethereum, compreso l'ultimo lanciato di recente *EIP-1559*, riescono a risolvere questa problematica. Sebbene EIP-1559

sia principalmente progettato per migliorare la prevedibilità delle commissioni di transazione, l'aggiornamento presenta anche una funzione di riduzione delle commissioni che influisce negativamente sulla redditività dei minatori, che a sua volta può portare i minatori ad aumentare l'estrazione di MEV per compensare la riduzione della ricompensa. Invece Ethermine, un pool minerario che rappresenta circa il 20% della potenza hash di Ethereum, ha già introdotto un programma di estrazione MEV per ridistribuire i profitti estratti tramite MEV tra tutti i minatori del pool.

2.3 Formato delle transazioni

In questa Sezione descriveremo i vari campi che sono presenti nelle transazioni Ethereum che variano in base al tipo di transazione. Su Ethereum infatti sono presenti due tipologie di transazioni, le 0 (legacy) e le 2 (EIP-1559). Il loro formato è composto dai seguenti campi:

- **Status** per raffigurare lo stato di una transazione, in modo da capire se si è conclusa con successo. Per le transazioni scaricate lo stato è sempre "Success".
- **Transaction hash** è un identificatore unico, di 66 caratteri, che viene generato quando una transazione viene eseguita e serve per identificare una specifica transazione.
- **Block** indica il numero del blocco in cui la transazione viene registrata.
- **Timestamp** indica la data e l'ora dell'esecuzione della transazione.
- **Transaction Action** evidenzia solo gli eventi chiave che sono rilevanti per l'utente finale. In questo modo gli utenti sono in grado di dire rapidamente cosa sta effettivamente accadendo all'interno di una transazione.
- **From** rappresenta l'indirizzo del mittente e quindi colui che esegue ed invia la transazione.
- **To** rappresenta l'indirizzo destinatario, ovvero la parte ricevente della transazione.
- **Tokens Transferred** indica la lista di token trasferiti all'interno della transazione, specificando il token destinatario e mittente.

- **Value** indica l'importo di ETH da trasferire al destinatario.
- **Transaction Fee** rappresenta l'ammontare della tassa pagata al minatore per processare la transazione.
- **Input Data** contiene dati aggiuntivi inclusi per la transazione considerata. Comunemente viene utilizzato come parte dell'interazione contrattuale o come messaggio inviato al destinatario. Contiene la funzione utilizzata all'interno della transazione e il Method ID associato.
- **Ether Price** rappresenta il prezzo di chiusura dell'unità di misura Ether, utilizzata per misurare il lavoro svolto da Ethereum per effettuare transazioni in rete, nella data in cui è stata eseguita la transazione considerata.
- **Txt Type** che specifica il tipo di transazione, che può essere 0 (Legacy) oppure 2 (EIP-1559). A volte questo campo è assente.

Fondamentale nell'esecuzione di un'attività è il **Gas**. Ricordiamo che il gas rappresenta la ricompensa data al minatore, ovvero colui che crea il blocco nella blockchain, per la sua attività. La quantità di gas usato dipende dalla quantità di risorse computazionali usate per l'esecuzione della transazione, ma ogni utente può impostare la propria proposta di gas. In queste transazioni sono presenti due campi per il gas:

- **Gas Price** che indica la commissione addebitata durante l'elaborazione di una transazione, o l'esecuzione di un contratto, sulla piattaforma Ethereum. Vengono utilizzate unità decimali di Ether, chiamate Gwei.
- **Gas Limit** specifica l'ammontare massimo che il mandante può pagare. Se il gas usato in una transazione eccede questo limite, la computazione viene bloccata e la transazione fallisce.

Per evitare attacchi di tipo "replay", ovvero casi in cui una transazione creata da un utente viene ripetuta da un attore malintenzionato, alle transazioni viene aggiunto un campo chiamato "Nonce". Il Nonce è un valore scalare pari al numero di transazioni inviate da un determinato indirizzo. Questo campo rende ogni richiesta univoca in modo che un utente malintenzionato

Figura 2.5: Formato transazioni "NO Type"

non possa riprodurla in un contesto diverso. Inoltre attraverso questo valore è possibile visualizzare l'ordine cronologico di esecuzione delle transazioni. I miner estraggono ed includono in un blocco valido la transazione con il Nonce inferiore. Attraverso le Figure 2.6 e 2.5 è possibile visualizzare il formato delle transazioni 0 (Legacy) e "NO Type" precedentemente descritte. Queste due hanno tutti i campi identici, ma nelle seconde non c'è il campo **Txt Type**, mentre nelle transazioni 0 (Legacy) è presente. Proseguendo nel delineamento dei formati dei nostri dati, le transazioni 2 (EIP-1559) contengono tutti i campi illustrati per le 0 (Legacy) e le "NO Type", ma si distinguono da queste ultime per i seguenti campi aggiuntivi:

- **Base Fee Per Gas** che indica la tariffa base per gas del blocco.
 - **Max Fee Per Gas** ovvero l'importo massimo che un utente è disposto a pagare per la propria transazione.

Figura 2.6: Formato transazioni 0 (Legacy)

Figura 2.7: Formato transazioni 2 (EIP-1559)

- **Max Priority Fee Per Gas** è un valore impostato dall'utente e rappresenta la parte della commissione di transazione che va al miner.
 - **Burnt Fees** rappresentano le commissioni bruciate in questa transazione.
 - **Txn Saving** indica la commissione di transazione risparmiata e si calcola usando la formula: $(\text{commissione max per gas}) - (\text{commissione base per gas} + \text{commissione prioritaria per gas}) * (\text{gas usato})$.

Nella Figura 2.7 è visibile una transazione appartenente alle 2 (EIP-1559), in cui è possibile visualizzare il formato di questa tipologia di transazioni e le principali differenze evidenziate.

Txn Hash	Method ⓘ	Age	From	To	Quantity
0x799fb7a470320923a4...	Swap	8 days 18 hrs ago	0x74de5d4fcf63e00296...	0xc780c4bdee7a0186...	2,168.134738802588936936
0x799fb7a470320923a4...	Swap	8 days 18 hrs ago	Uniswap V2: YUP	0x74de5d4fcf63e00296...	2,168.134738802588936936
0x8e9a75111da463049...	Swap Exact Token...	9 days 14 hrs ago	0xb84969ea02fd374ad1...	Uniswap V2: YUP	293.830068266581774178
0x28b04ff11da80ac6f964...	Transfer	9 days 17 hrs ago	0x972eebcb30252640c6...	0xcbd6ed4fd56d83bac9...	165,311.392996047365934841
0x412e973d64841a8cee...	Remove Liquidity...	9 days 18 hrs ago	Uniswap V2: Router 2	0x972eebcb30252640c6...	165,311.392996047365934841
0x412e973d64841a8cee...	Remove Liquidity...	9 days 18 hrs ago	Uniswap V2: YUP	Uniswap V2: Router 2	165,311.392996047365934841
0xa3b93b7010da49877e...	Add Liquidity ET...	9 days 18 hrs ago	0x972eebcb30252640c6...	Uniswap V2: YUP	165,311.392996047365936497
0xc137e87757f804b03d...	Exec Transaction	9 days 19 hrs ago	0xcbd6ed4fd56d83bac9...	0x972eebcb30252640c6...	165,311.392996047365936497
0x50b959460976a3bae7...	Swap Exact Token...	9 days 20 hrs ago	0x83de69633b281f2a49...	Uniswap V2: YUP	1,871.665948117229013791
0xfd83ba300fd7dccac6c...	Swap Exact Token...	10 days 6 hrs ago	0xf8e8f48d0e72e41a21a...	Uniswap V2: YUP	82.130275598129758504
0xb9def4b71fbc7b1f9e8...	Swap Exact ETH F...	10 days 7 hrs ago	Uniswap V2: YUP	0xf8e8f48d0e72e41a21a...	82.130275598129758504
0xa33ba066e96c669c30...	Swap	10 days 12 hrs ago	0x74de5d4fcf63e00296...	Uniswap V2: YUP	253.9118
0xa33ba066e96c669c30...	Swap	10 days 12 hrs ago	0xcc993aba2ffd89db99c...	0x74de5d4fcf63e00296...	253.9118
0xae8c1265f06f014e29...	Transfer	10 days 15 hrs ago	0xbcb5dc467d09d518a0...	0x39ecbd3ddcf4800f33f...	2,141.799917510521106536
0xac7c535c8bd36ddc30...	Transfer	10 days 22 hrs ago	0xf8e8f48d0e72e41a21a...	0xcc993aba2ffd89db99c...	96

Figura 2.8: Formato trasferimenti token ERC20

Concludendo, attraverso la Figura 2.8 è possibile visualizzare i campi relativi ai trasferimenti di token:

- **Method** rappresentate la funzione eseguita in base ai dati di input decodificati.
- **Age** simile al campo **Timestamp**, indica il numero di giorni e di ore passati dall'esecuzione della transazione.
- **Quantity** indica l'ammontare di token trasferiti all'interno di una transazione.

I campi di trasferimento di token ERC20 non variano in base alla tipologia di transazione, perché sono legati allo standard ERC20 e quindi gestiti da una tipologia specifica di smart contract, in maniera indipendente dal formato delle transazioni che servono per attivarli.

2.4 DeFi ed Exchange Decentralizzati (DEX)

La finanza decentralizzata **DeFi** è una forma sperimentale di sistema finanziario che non si basa su intermediari finanziari centrali come broker, exchange o banche e utilizza invece smart contract sulla blockchain. Il motore di questa tipologia di finanza è rappresentando dai **DEX**.

VANTAGGI

Nessuna custodia dei fondi
da parte dell'exchange

Nessuna KYC:
non c'è bisogno di verificare
l'identità del cliente

Possibilità di scambiare tutti i token
compatibili con la blockchain:
scambiare anche criptovalute
non presenti nei classici exchange,
l'importante è che siano compatibili
con la blockchain utilizzata

Tabella 2.1: Vantaggi DEX

Un **exchange decentralizzato (DEX)** è un luogo dove è possibile effettuare uno scambio peer to peer con un altro utente senza l'intermediazione di nessuno. Non c'è un ente centrale (come una piattaforma centralizzata) che prende in custodia i fondi degli utenti, ma questi ultimi hanno il loro controllo totale. Gli ordini sugli exchange decentralizzati avvengono tramite smart contract e sono tutti *on-chain*, ovvero si effettuano direttamente sulla blockchain senza l'intermediazione di nessuna società. Esistono DEX che offrono ordini *off-chain*, ma hanno un grado superiore di centralizzazione. Infatti in questo caso gli ordini non vengono registrati direttamente sulla blockchain, ma vengono messi da parte in un altro luogo. Infine ci sono i *Market Maker Automatizzati (AMM)*, come per esempio **Uniswap**. Essi mettono in comunicazione più smart contract e propongono incentivi agli utenti per farli partecipare e di conseguenza portare avanti tutte le operazioni. I DEX hanno sia vantaggi sia svantaggi. Procediamo elencando, di seguito, i principali vantaggi nella tabella 2.1: In prosieguo, attraverso la tabella 2.2 descriviamo alcuni svantaggi dei DEX.

SVANTAGGI

Poco user-friendly:

meno facile da utilizzare

Liquidità:

il volume di trading non è garantito,

puoi scambiare crypto solo se esiste la controparte

e non essendoci un ente centrale

ciò non può essere garantito.

Commissioni:

in alcuni casi le commissioni sui DEX

sono più elevate.

Tabella 2.2: Svantaggi DEX

2.4.1 Uniswap

Quando parliamo di **Uniswap** [1] ci riferiamo a due diverse realtà tecnologiche. Da un lato il protocollo su rete Ethereum che gestisce un exchange automatizzato e decentralizzato, dall'altro invece il token di questo protocollo, che sarebbe più proprio chiamare **UNI**. **Uniswap** è principalmente uno scambio decentralizzato (DEX) che consente a chiunque di scambiare token appartenenti allo standard ERC20, che è quello proprio di Ethereum, per realizzare token basati sulla sua blockchain. Uniswap è progettato per funzionare come un bene pubblico, per scambiare token senza commissioni di piattaforma o intermediari. Inoltre, a differenza della maggior parte degli scambi che abbinano acquirenti e venditori per determinare i prezzi ed eseguire operazioni, Uniswap utilizza una semplice equazione matematica e pool di token ed ETH per svolgere lo stesso lavoro. Questa equazione è rappresentata con il meccanismo di determinazione dei prezzi chiamato "*Modello di Market Maker di prodotti costanti*". Attraverso questo ultimo, qualsiasi token può essere aggiunto a Uniswap finanziandolo con un valore equivalente di ETH e il token ERC20 scambiato. Infatti Uniswap invece di collegare acquirenti

e venditori per determinare il prezzo di un token, utilizza questa equazione costante 2.1.

$$x * y = k \quad (2.1)$$

In cui, x e y rappresentano la quantità di token ETH ed ERC20 disponibili in un pool di liquidità e k è un valore costante. Questa equazione utilizza l'equilibrio tra i token ETH ed ERC20 e l'offerta e la domanda per determinare il prezzo di un particolare token. Grazie a questo meccanismo di determinazione del prezzo dei token, Uniswap agisce come un *Automated Market Maker*, ovvero come un protocollo automatico. Gli AMM sono smart contract che contengono riserve di liquidità contro cui è possibile fare trading. Queste riserve sono finanziate da fornitori di liquidità. Chiunque può essere un fornitore di liquidità depositando un valore equivalente di due token nella pool. In cambio, i trader pagano una commissione alla pool che viene in seguito distribuita ai fornitori di liquidità in base alla loro quota della pool. Essendo Uniswap un protocollo che si appoggia alla blockchain di Ethereum ed essendo il suo token di governance un token ERC 20, all'interno del DEX possono essere inseriti tutti i token che utilizzano lo stesso protocollo. Questo lascia a tutti una certa libertà, per quanto riguarda i token da aggiungere. E' possibile quindi creare un proprio token ERC20 e inserirlo all'interno del progetto senza dover ottenere autorizzazioni. Tutti gli scambi e le quotazioni di nuovi token avvengono tramite smart contract che sono supportati all'interno della blockchain di Ethereum. Al giorno d'oggi esistono tre versioni di Uniswap:

- **Uniswap v1** rappresenta un sistema on-chain di smart contract su Ethereum, che implementa un protocollo di liquidità automatizzato, basato sulla formula di prodotto costante precedentemente descritta. Ogni coppia di Uniswap v1 immagazzina le riserve riunite di due asset e fornisce liquidità per quei due beni, mantenendo l'invariante che il prodotto delle riserve non può diminuire.
- **Uniswap v2** è la nuova implementazione basata sulla stessa formula, che consente la creazione di coppie ERC20/ERC20, invece di supportare solo coppie ERC20/ETH. Questo risulta essere vantaggioso per gli scambi poiché permette di scambiare tra di loro token ERC20, invece di dover necessariamente prima trasformare i token in possesso in ETH e poi successivamente trasformare l'ETH ottenuto in token desiderati, andando così a ridurre le tasse da pagare per i trasferimenti. Abilita i "flash swap" in cui gli utenti possono

ricevere beni liberamente e utilizzarli altrove sulla catena, solo pagando (o restituendo) tali beni al termine della transazione. Uniswap v2 risolve anche alcuni problemi minori con Uniswap v1, ovvero riduce la superficie di attacco di Uniswap e rende il sistema più facilmente aggiornabile minimizzando la logica nel contratto “core” che detiene i fondi dei fornitori di liquidità.

- **Uniswap v3** [16] è differente dalle versioni precedenti per gli yield farmers, ovvero chi mette a disposizione liquidità per i Liquidity Providers (LP). Le due differenze principali sono la liquidità concentrata, cioè Uniswap v3 offre la possibilità ai LP di decidere un range nel quale offrire liquidità, e l’efficienza del capitale che consiste che i liquidity provider guadagnano una percentuale uguale finché il pool si muove nel margine/range da loro indicato. Più stretto è il range, maggiore è l’interesse.

2.4.2 Token Uniswap (UNI)

UNI è il token nativo del protocollo Uniswap e conferisce diritti di governance a chi lo possiede. Questo significa che i possessori di UNI possono votare le modifiche al protocollo. In fase di rilascio sono stati coniati 1 miliardo di token UNI. Il 60% è stato distribuito ai membri esistenti della community di Uniswap. Parte della distribuzione alla comunità è avvenuta attraverso il *liquidity mining*, ovvero il token UNI è stato distribuito a coloro che forniscono liquidità nelle seguenti pool di Uniswap:

- ETH/USDT
- ETH/USDC
- ETH/DAI
- ETH/WBTC

Chiunque usi Uniswap, può richiedere 400 token UNI per ogni indirizzo con cui hai usato Uniswap. Infine detenendo il token UNI è possibile votare riguardo una decisione, ed è possibile anche delegare il proprio voto ad una terza parte.

2.4.3 DEX in Ethereum

Uniswap è tutt'oggi l'exchanger decentralizzato maggiormente utilizzato dagli utenti, per via delle sue caratteristiche come l'accesso al protocollo, che può essere effettuato utilizzando qualsiasi portafoglio web3 e la possibilità di creare applicazioni personalizzate su di essi. E inoltre permette di creare uno scambio per qualsiasi token ERC20. Ma oltre questo DEX, su Etheruem sono disponibili ulteriori exchanger, elencati di seguito:

- **Sushiswap**: cerca di incentivare gli utenti a gestire una piattaforma in cui possono acquistare e vendere risorse crittografiche in modo semplice.
- **1inch V3**: aggregatore DEX che seleziona i prezzi di criptovaluta più economici su tutti gli scambi decentralizzati.
- **1 inch.exchange v2**: introduce le API Pathfinder che aiutano gli utenti a trovare i migliori percorsi per uno scambio di token in breve tempo.
- **Gitcoin Grants**: Questa piattaforma è stata creata per finanziare e coordinare lo sviluppo di prodotti open source utilizzando metodi innovativi, quali il finanziamento quadratico.
- **Mirror Protocol**: protocollo basato sulla creazione di asset sintetici che permettono di esporsi al prezzo del corrispettivo asset reale attraverso un sistema peer-to-peer.
- **Balancer**: si concentra sulla creazione automatica di mercati, utilizzando pool di liquidità e offrendo capacità di cambio valuta decentralizzata.
- **Aave**: protocollo open source per guadagnare interessi su depositi e attività di prestito.
- **DODOEX**: protocollo che fornisce liquidità on-chain.
- **Alchemist**: piattaforma che offre l'accesso a più piattaforme per guadagnare premi e partecipare ad aste di lancio equo di token.

Capitolo 3

Analisi del social token YUP

In questo Capitolo illustriamo le caratteristiche principali e le funzionalità della piattaforma Yup. Dopo una sezione introduttiva, nella quale presentiamo le principali caratteristiche, descriveremo le sue funzionalità, concentrando in particolar modo sul sistema di voto che rappresenta la meccanica intorno alla quale si basa la dApp. Questo ci porterà a parlare di molti concetti, come l’Influenza e tutti i parametri necessari per calcolarla. Data la loro centralità per questa tesi, discuteremo anche alcune funzionalità più legate all’aspetto economico della piattaforma, tra cui il funzionamento del meccanismo delle ricompense, del token YUP, dell’EOS-ETH Bridge, ovvero di come è possibile trasferire il token YUP tra le due blockchain. Infine presenteremo lo scopo di lavoro di questa tesi, introducendo brevemente le varie fasi del lavoro.

3.1 Piattaforma Yup

Yup è un’applicazione decentralizzata (dApp) per la valutazione dei contenuti disponibili su Internet. Iniziamo la descrizione dei concetti della dApp rilevanti a questa tesi, introducendo i principali aspetti innovativi introdotti da questa piattaforma. L’applicazione è implementata tramite una collezione di smart contract che sono dislocati su due blockchain. In particolare, Yup sfrutta EOS per la sua scalabilità e la possibilità di poter creare numerose transazioni solo tramite un investimento iniziale limitato. La blockchain di Ethereum è invece scelta per la sua

liquidità e per il vastissimo mercato di altri token che mette a disposizione dei suoi utilizzatori, fornendo accesso ad un mercato dalle infinite possibilità.

Gli utenti di questa piattaforma sono nominati "Yupsters" e all'interno della community possono valutare i contenuti sulle varie piattaforme sociali, esprimendo opinioni su differenti tipi di contenuto pubblicato. Esempi di contenuti che sono più comunemente votati sono: post, video, immagini o tweet. Tramite la piattaforma è anche possibile votare un qualsiasi contenuto, non necessariamente social, purché abbia un link univoco. Un'ulteriore funzionalità che Yup offre è creare un archivio in cui inserire i contenuti preferiti dell'utente, e questi contenuti sono soggetti a votazioni da parte di altri utenti. Le votazioni effettuate da un utente permettono a quest'ultimo di ricevere una ricompensa. In Yup infatti esiste un sistema di ricompense, in cui queste ultime sono assegnate ad una persona basandosi su opinioni largamente condivise tra gli utenti. Inoltre, la ricompensa assegnata ad ogni utente per un suo voto viene calcolata in base ai voti effettuati dopo il voto dell'utente. Questo meccanismo è utile per evitare che una persona dia un giudizio ad un contenuto solo perché influenzato dal voto di altre persone. Inoltre viene promosso l'essere i primi a votare i contenuti pubblicati. Di conseguenza ogni contenuto e ogni individuo è classificato in base al proprio valore.

La principale caratteristica di Yup è che non nasce come un'alternativa alle piattaforme di Social Media già esistenti, ma è progettata per integrarsi con queste ultime. Un utente per potervi accedere deve solamente scaricare l'estensione sul proprio browser, non abbandonando le piattaforme già utilizzate. Questo è reso possibile da due componenti fondamentali:

- **Yup protocol** è la componente software decentralizzata della dApp che opera sulla blockchain EOS.
- **Yup token** è un asset crittografico fungibile utilizzato per incentivare il protocollo Yup e in particolare per le ricompense.

Yup gestisce i servizi offerti tramite gli smart contract. In particolare vengono utilizzati i seguenti:

- **yupyupyup** è il protocollo principale
- **token.yup** è il protocollo relativo al token YUP

- **bridge.yup** è relativo al bridge EOS-ETH
- **lptoken.yup** è relativo al token per i Liquidity Provider
- **lpbridge.yup** è il protocollo relativo al bridge di token YUP-ETH LP

3.2 Servizi di Yup

Prima di poter fare una presentazione tecnica del funzionamento della dApp, riteniamo necessario descrivere il suo funzionamento, ad alto livello, da un punto di vista di un utente e concentrando in particolar modo sull'attività di voto.

3.2.1 Registrazione

Inizialmente, per poter usufruire di questa piattaforma, bisogna registrarsi ad essa in modo da diventare un utente di Yup. E' possibile effettuare questa operazione in tre modalità: **Twitter Account, Email, Wallet Connect**. Attraverso l'ultima modalità l'iscrizione è immediata, mentre per iscriversi tramite l'account Twitter bisogna che l'account esista da un certo periodo di tempo e che abbia minimo 50 followers. Se invece la registrazione avviene tramite un indirizzo email, l'iscrizione sarà completata dopo la revisione e l'autorizzazione. Queste condizioni imposte dagli ultimi due metodi, servono per evitare l'iscrizione di bot sulla piattaforma. Gli utenti iscritti tramite Twitter Account o Wallet Connect, partono con un valore di influenza, un valore cruciale usato principalmente nel calcolo delle ricompense (vedi Sezione 3.3.1) pari ad 80, mentre chi si registra tramite email parte con solo 20 di influenza. Di conseguenza si ha la possibilità di trarre profitti maggiori e si ha anche una capacità di voto giornaliera differente tra i vari utenti. Su Yup esistono due tipi di account: **non-Mirror** e **Mirror**. La prima tipologia corrisponde agli account di persone reali. Il secondo tipo rappresenta account creati e gestiti dalla piattaforma Yup e corrispondono ad influencers di Twitter. Questi replicano le attività eseguite dall'account di Twitter e il loro obiettivo è rappresentare i token guadagnati dai vari influencers creando contenuti. Questi account sono stati creati perlopiù per fare pubblicità alla piattaforma, mostrando le potenziali ricompense di personalità social famose.

3.2.2 Attività di voto

Una volta terminata la fase di registrazione, l’utente può accedere alla piattaforma e usufruire dei servizi da essa offerti. Tra questi servizi rientra la possibilità di votare dei contenuti pubblici. Gli utenti su Yup hanno una capacità di voto giornaliera limitata, la **Voting Power**. Questo parametro viene resettato ogni 24 ore, e il conteggio di queste ore inizia dal momento in cui l’individuo effettua la sua prima votazione.

Ad ogni contenuto è possibile assegnare un valore da 1 a 5, dove le votazioni da 1 a 2 sono negative, mentre da 3 a 5 sono intese come positive. Ogni votazione richiede una diversa porzione della propria Voting Power e ne consegue anche una diversa assegnazione delle ricompense. Per questo motivo ogni utente tende ad esprimere votazioni totalmente negative, o totalmente positive solo per quei contenuti che ritiene effettivamente validi. Ogni persona può esprimere un numero di voti diverso, infatti il numero di voti che ogni utente ha a disposizione in un giorno dipende dalla sua influenza. I nuovi utenti possono effettuare 20 voti con valori pari a 2/5 o 3/5, 5 voti con un valore pari a 4/5 o 1/5 e circa 3 voti con un valore pari a 5/5. Gli utenti con influenza pari o maggiore a 90, possono effettuare il doppio dei voti rispetto ai nuovi individui. La piattaforma, inoltre, mette a disposizione dell’utente la possibilità di cambiare le opinioni espresse nel tempo. Il voto espresso può quindi essere modificato o perfino eliminato. In base all’azione effettuata, viene modificata la **Voting Power**. Quest’ultima viene ridotta se la nuova valutazione è più estrema rispetto alla precedente (esempio: cambiando un voto da 2 a 1), aumentata se è meno estrema (esempio: un voto cambiato da 5 a 3). I voti possono essere anche cancellati, ed in questo caso la voting power viene completamente rimborsata. Questo servizio torna utile agli utenti, in caso di errata valutazione o di ripensamento, ma potrebbe favorire un abuso di questa funzionalità eseguendo la cancellazione di un voto per cui è già stata ricevuta la ricompensa o per contenuti vecchi che difficilmente verranno votati nuovamente, e ottenendo un rimborso della Voting Power, in modo da usufruire di maggiori voti durante una giornata.

Il sistema di votazione di Yup consente alle persone di esprimere la propria opinione in modo più accurato e guadagnare influenza in specifici campi di competenza, tramite la presenza delle categorie. L’utente può scegliere la categoria in cui esprimere la sua valutazione.

Attraverso le varie categorie la piattaforma prevede di migliorare la condivisione delle opinioni. Al momento sono presenti 11 categorie, rappresentate nella tabella 3.1. Infine è possibile

CATEGORIE
like: una misura della popolarità (per il contenuto)
smart: una misura della popolarità (per il contenuto)
funny: una misura di ilarità (per il contenuto)
chill: una misura di disinvolta (per le persone)
useful: una misura di competenza (per le persone)
knowledgeable: una misura di utilità (per i corsi)
engaging: una misura di interesse (per i corsi)
easy: una misura di sforzo (per i corsi)
interesting: una misura di intrigo (per i corsi)
affordable: una misura del costo (per i luoghi)
beautiful: una misura di bellezza (per i luoghi)

Tabella 3.1: Tabella Categorie

esprimere un giudizio su qualsiasi cosa possieda un URL. Per questo motivo Yup è definito come un sistema di valutazione di Internet, anche se la sua maggiore espressione riguarda il lato sociale.

3.2.3 Creazione di Collezioni

Un ulteriore servizio messo a disposizione degli utenti è la creazione di collezioni. Ogni individuo può creare un numero illimitato di collezioni e aggiungere ad esse i suoi contenuti preferiti. Questa operazione può essere effettuata sempre, poiché non dipende dalla Voting Power. Anche le collezioni possono essere votate tramite Yup, e i voti hanno diritto a ricevere le relative ricompense.



Figura 3.1: Colori valore sociale

3.2.4 Valore Sociale

L’attività di voto degli utenti ha come effetto quello di determinare il cosiddetto *Valore Sociale* dei contenuti, ovvero una sorta di valutazione della veridicità e dell’impatto sociale di ogni contenuto. Yup permette di visualizzare il valore sociale attraverso i colori. I contenuti e gli utenti vengono classificati e sottolineati con il colore della loro classificazione nell’estensione e nell’applicazione web. Ogni colore rappresenta un valore percentile in relazione a tutto il resto nella sua categoria. Il sistema di classificazione utilizza cinque colori:

- **Turchese:** top 20%
- **Verde:** 20% - 40%
- **Giallo:** 40% - 60%
- **Arancione:** 60% - 80%
- **Rosso:** 80% - 100%

Il contenuto diventa più verde quando è valutato positivamente e più rosso quando è valutato negativamente, riflettendo un aumento o una diminuzione del valore sociale del contenuto. Inoltre quest’ultimo ha una classificazione del colore diversa per ciascuna categoria di classificazione. Per determinare questi percentili, il contenuto viene valutato in base alla classificazione di altri contenuti in quella categoria. La classificazione del colore viene raffigurata di seguito, nella Figura 3.1.

3.2.5 Integrazioni

Come anticipato, uno dei punti di forza di Yup è la sua possibilità di integrarsi con altre piattaforme. Infatti gli utenti hanno a disposizione alcune modalità per collegare i propri account social o utilizzare l'estensione Yup per monitorare automaticamente i likes e i ratings. Il grado di integrazione differisce a seconda delle piattaforme. Di seguito sono elencate alcune piattaforme con i loro rispettivi gradi di integrazione:

- **Twitter**: OAuth
- **Youtube**: Full Overlay
- **Reddit**: Full Overlay
- **Google e Google Maps**: Full Overlay
- **SuperRare**: Action Tracking
- **Audius**: Action Tracking
- **Rarible**: Action Tracking

Twitter è attualmente l'unica piattaforma ad essere integrata tramite **OAuth**, e quindi è possibile registrarsi su Yup e verificare il proprio account tramite quello di Twitter. La possibilità di esprimere un voto tramite l'**Overlay** piuttosto che l'estensione è disponibile solo su alcune piattaforme, come Youtube, Reddit, Google e Twitter. Invece SuperRare, Audius e Rarible supportano per ora l'**Action Tracking**, ovvero l'estensione registra i like espressi su queste piattaforme e genera un rating corrispondente con un valore di 3/5, nella categoria "Like".

Avendo concluso una discussione ad alto livello riguardo le funzionalità accessibili agli utenti, passiamo ora a discutere gli aspetti tecnici riguardo il meccanismo delle ricompense

3.3 YUP Protocol

Il protocollo di Yup è un protocollo di consenso sociale per Internet. Rappresenta un software autonomo che stabilisce l'infrastruttura per la rete di Yup, determinando l'asset allocation, semplificando la misura, lo scambio e l'acquisizione del capitale sociale, e inoltre garantisce:

1. Trasparenza degli account e strumenti di filtraggio in base alla loro influenza
2. Monetizzazione equa e diretta di opinioni, influenze e contenuti
3. Identità digitali con in gioco il capitale sociale
4. Codici di condotta guidati dalla comunità
5. Governance della rete determinata dall'influenza diretta
6. Equa distribuzione delle entrate
7. Proprietà senza fiducia dell'impronta di rete

Il protocollo sociale di Yup è implementato tramite gli smart contract di Yup attualmente disponibili sulla blockchain EOS. Di seguito ci focalizzeremo sull’Influenza di un utente e i parametri che definiscono tale valore. Infine descriveremo il meccanismo delle ricompense e parleremo del EOS-ETH Bridge. Quest’ultimo permette a Yup di essere una piattaforma cross-chain e cross-platform.

3.3.1 Influenza

L’Influenza è la metrica utilizzata per valutare la distribuzione della ricompensa dei token, la governance della rete, la rappresentazione trasparente del valore sociale e l’impegno della rete. L’influenza minima è 0, ma non ha nessun limite superiore. Per ogni individuo, più alto è il valore della sua influenza e più peso avranno le sue opinioni. Il suo valore viene calcolato dal protocollo, per ogni persona, attraverso l’equazione 3.1:

$$I = \beta_1 \sqrt{A} + \beta_2 \sqrt{a} + \beta_3 \sqrt{s} + b \quad (3.1)$$

I 4 parametri che governano l’Influenza sono:

- **Coin Age (A)**
- **Activity (a)**
- **Social Level Consensus (s)**

- **Boost (b)**

Partiamo illustrando la prima componente, ovvero la **Coin Age**. Tramite questo parametro, i token detenuti da un account corrispondono ad una **Age**. In questo modo, oltre a considerare la quantità di token, viene tenuto in considerazione anche da quanto tempo l'utente ne è in possesso. Da questa componente derivano dei vantaggi:

1. Impedisce a nuovi utenti di incrementare in maniera significativa la propria influenza acquistando grandi quantità di token in un breve periodo.
2. Permette al protocollo di sostituire il processo di staking-unstaking, senza rinunciare a sicurezza e stabilità.
3. Rende il tempo una vera e propria risorsa per gli utenti.

Questa componente è espressa attraverso l'equazione 3.2:

$$A = \sum_{i=1}^n Y_i t_i \quad (3.2)$$

Dove il parametro Y rappresenta il token value di una transazione che assegna dei token ad un utente, e il parametro t è il tempo trascorso da quando sono stati ottenuti i token. Oltre all'Age, l'**Activity** è una componente importante per la valutazione del capitale sociale, ma è difficile da misurare correttamente. Essa rappresenta il contributo di un utente all'interno della rete. Per determinarla si usufruisce dei token YUP ottenuti dall'utente come ricompensa per la creazione e la cura di contenuti pubblici. Non si considerano invece i token YUP semplicemente acquistati. L'Activity viene calcolata definendo il parametro R_u come i premi ricevuti dai token YUP appena coniati per ogni azione eseguita da un account specifico u . Il parametro a invece rappresenta l'attività e la rappresentazione del valore di rete dei contributi di un account misurando i premi che ha ricevuto dall'impegno precedente i . Di seguito riportiamo la formula per iscritto attraverso l'equazione 3.3:

$$a = \sum_{i=1}^n R_u, i \quad (3.3)$$

Per evitare che gli utenti effettuino un'eccessiva attività, priva di qualità, viene imposto un valore limite di 10 azioni giornaliere. Le azioni effettuate successivamente a queste 10, hanno un impatto nullo.

Un'altra importante componente che determina l’Influenza, è il **Boost (Burning Mechanism)**. Questo meccanismo è fornito dalla piattaforma con lo scopo di mantenere la fornitura di YUP tokens in distribuzione e aumentarne la richiesta. Questo permetta ad un utente di bruciare in modo permanente i token posseduti, in modo da ottenere un incremento temporaneo della sua Influenza su un’azione da lui eseguita. Il numero di token che un account può bruciare è limitato. Il Boost è matematicamente definito attraverso l’equazione 3.4:

$$b = \varsigma Y_{burn}, u \quad (3.4)$$

L’ultima componente che determina l’Influenza è la **Social Level Consensus**. Il Social Level è un rank numerico corrispondente ad ogni indirizzo (utente), e viene determinato in base a tutti gli altri utenti presenti in rete. Ogni account detiene una lista ordinata come preferisce e modificabile di tutti gli altri utenti. Matematicamente questo parametro si calcola attraverso l’equazione 3.5:

$$\bar{s} = \sum_{i=1} \frac{s_i * (\log(w_i) + 1)}{\sqrt[5]{e_p}} \$\$e = \sum_{i=1}^n (ri - \mu_i)^2 \quad (3.5)$$

dove w (*weight*) è un peso relativo ad un **ordine utente (UO)** e si calcola in base al livello dell’utente nell’**ordine aggregato (AO)** del blocco precedente. e (**error**) è invece una misura che dipende dalla relazione tra la lista mantenuta dall’utente e quella aggregata. Questo parametro incentiva l’utente ad effettuare un ordinamento corretto. In caso quest’ultimo sia errato, l’utente subisce una riduzione del suo livello sociale.

3.3.2 Meccanismo delle ricompense

Il meccanismo delle ricompense è una parte fondamentale del protocollo e si occupa della generazione di nuovi token e della loro distribuzione. Esistono tre tipi di ricompense: **Ricompense creatore**, **Ricompense curatore** e **Ricompense LQ** (destinate ai liquidity provider). Le ricompense LQ verranno discusse in Sezione 3.5.1, mentre le ricompense per creatori e curatori

verranno discusse nel resto della Sezione. La maggior parte dei token generati sono distribuiti agli autori di contenuti in maniera proporzionale alle valutazioni che essi ricevono dalla community. Questa porzione di token prende il nome di **creation allocation**. Per rappresentare matematicamente quest'ultima, bisogna definire altri due valori. Per primo indichiamo con V_h il valore di una determinata azione che viene eseguita nella rete tramite l'interazione con il protocollo. Questa è espressa, attraverso l'equazione 3.6, come il rapporto tra l'Influenza dell'azione e quella totale delle azioni eseguite in rete in un determinato arco di tempo.

$$V_h = \frac{I_i}{I_{\bar{i},t}} Y_{c,t} \quad (3.6)$$

Il secondo valore da introdurre viene indicato come R_i e prende il nome di **creaction reward**. Questo è il valore in token di un contenuto, espresso attraverso l'equazione 3.7 come la sommatoria dei valori delle azioni V_j eseguite per la creazione di quest'ultimo.

$$R_i = \sum_{j=1}^m V_j \quad (3.7)$$

Infine possiamo ora definire matematicamente la **creaction allocation**. Quest'ultima, raffigurata con l'equazione 3.8, è definita in relazione ad un determinato periodo di tempo t come la sommatoria di tutte le corrispondenti **creation reward**.

$$Y_{c,t} = \sum_{i=1}^n R_i \quad (3.8)$$

Ricompense Creatori e Curatori

La ricompensa di un **Creatore** viene definita come la porzione della ricompensa di un contenuto assegnata al creatore di esso. Il creatore riceve almeno la metà della ricompensa, mentre la parte rimanente viene divisa tra il creatore e tutti i curatori che hanno contribuito al contenuto.

Matematicamente questa ricompensa viene rappresentata attraverso l'equazione 3.9.

$$R_c = R_i \left(\frac{1 + \frac{I_c}{I_{pool,t}}}{2} \right) \quad (3.9)$$

dove, I_c è l'influenza del curatore durante il periodo che intercorre dal momento della creazione, e $I_{pool,t}$ è la somma totale dell'influenza di tutti i curatori che hanno votato il contenuto. Quindi la porzione della seconda metà assegnata al creatore, dipende dal rapporto tra la sua influenza e quella della pool.

Ogni **Curatore**, invece, riceve delle ricompense tenendo conto dei voti a loro successivi. In questo modo sono ricompensati maggiormente i curatori che hanno espresso una valutazione corretta sin da subito, rispetto a coloro che assegnano un voto in ritardo e in certi casi potrebbero anche esprimere opinioni condizionate dai giudizi già effettuati. La ricompensa del Curatore viene rappresentata dall'equazione 3.10, in cui V_q è il valore assegnato all'azione del curatore, V_j è un'azione j a lui successiva, mentre $V_{h,j}$ rappresenta la somma dei valori delle azioni precedenti a V_j .

$$R_q = \left(\frac{1 - \frac{I_c}{I_{pool,t}}}{2} \right) \sum_{j=1}^n \left(\frac{V_q}{V_{h,j}} \right) V_j \quad (3.10)$$

3.4 YUP Token

Il token YUP è un **crypto-asset** fungibile della blockchain di EOS utilizzato per incentivare il protocollo Yup e creare un'economia sociale fondata sull'attività della community. Questo token è stato introdotto nel protocollo nel 2019, al termine della fase di Beta Testing. Precedentemente veniva utilizzato un altro token **YUPX**, privo di valore e con il solo scopo di eseguire test di controllo del corretto funzionamento del protocollo. Al termine della fase di Beta Testing, i token YUPX sono stati sostituiti da token YUP al tasso 1:1 tramite air drop. Esiste una versione del token YUP sia su Ethereum sia su EOS e il valore varia in base alla liquidità presente sulle blockchain. Il token permette di raggiungere i seguenti obiettivi:

- **Incentivazione alla partecipazione:** in quanto vengono utilizzati come ricompense

- **Governance:** poiché è usato per governare il protocollo
- **Incentivazione alla liquidità:** in quanto i fornitori di liquidità ottengono YUP tokens come ricompensa

I nuovi token vengono coniati secondo un programma predeterminato e sono distribuiti attraverso il meccanismo delle ricompense.

3.5 Yup in Ethereum

Come anticipato, la piattaforma di Yup, sfrutta la blockchain di Ethereum per la liquidità dei propri token e per i gateway per moneta legale richiesti dal protocollo Yup. Yup si integra con Ethereum attraverso l'utilizzo di uno smart contract¹, il quale implementa un token che segue lo standard ERC20, tipico di Ethereum. Il meccanismo per poter far interagire le due blockchain si chiama Bridge, il quale consente, tramite operazioni atomiche, di trasferire dei token YUP dalla blockchain di EOS a quella di Ethereum e viceversa. Il Bridge funziona tramite la modalità nota col nome di **mint - burn**. Con questo metodo i token non vengono effettivamente spostati, ma eliminati su una blockchain (burn) e successivamente viene creato lo stesso quantitativo sull'altra (mint). Dato che il servizio del Bridge richiede l'esecuzione di una transazione Ethereum, richiede il pagamento del gas per la sua esecuzione il quale in genere viene addebitato all'utente che richiede di utilizzare questo servizio.

I token YUP possono essere utilizzati su Ethereum per differenti operazioni:

1. Per la liquidity pool: un gruppo di token YUP vengono depositati in un pool e bloccati in uno smart contract. I fornitori di liquidità, per incentiviarla, coltivano il rendimento Yup con i loro token YUP-ETH LP, puntandoli e collegandoli allo smart contract EOS. Questi token permettono il prestito decentralizzato (lending), il commercio (trading) e altre funzioni fornendo liquidità.
2. Per eseguire scambi su DEX: gli utenti possono effettuare scambi di un certo numero di token YUP per un altro insieme di token oppure per una somma di ETH. Questi scambi sono possibili perché il token YUP segue lo standard dei token ERC20.

¹Smart Contract: 0x69bbc3f8787d573f1bbdd0a5f40c7ba0aee9bcc9

3. Per effettuare staking dei token ETH-YUP: ovvero acquistare e conservare monete digitali per trarre guadagni. Inoltre attraverso lo staking l'utente ottiene diritti di voto sulle decisioni riguardo gli aggiornamenti e sul futuro della blockchain.
4. Per acquistare beni o servizi.
5. Per fornire alle persone accesso a un prodotto o servizio.
6. Per finanziare progetti basati su blockchain.

3.5.1 Ricompense Liquidity Provider

Completiamo il quadro delle ricompense descrivendo ora quelle riservate agli utenti che forniscono liquidità nella pool YUP-ETH di Uniswap. Infatti, senza una componente in grado di fornire liquidità nel pool di liquidità YUP-ETH di Uniswap, il token sarebbe poco appetibile per gli utilizzatori di Yup perché difficilmente utilizzabile al di fuori della piattaforma. Gli account che desiderano fornire liquidità per il protocollo, devono aggiungere alla liquidity pool di Uniswap una quantità di token ETH equiparabile alla quantità dei token YUP. Per ottenere i token YUP su Ethereum, bisogna acquistarli tramite DEX della blockchain, oppure trasferire da EOS, tramite il Bridge, quelli guadagnati come ricompensa. Una volta aggiunta la liquidità, gli account vengono ricompensati con dei token YUP-ETH Uniswap LP che certificano la magnitudine del loro contributo nel liquidity pool. Tramite un ulteriore Bridge è possibile trasferire questi token su EOS e se vengono messi in stake su Yup Racing, è possibile ricevere dei token YUP come ricompensa per la liquidità fornita.

3.6 Scopo della tesi

Le piattaforme di Social Media basate sulle blockchain sono considerate una valida alternativa alle soluzioni centralizzate, e il meccanismo di ricompense è un ottimo metodo per incentivare la partecipazione degli utenti. Lo scopo di questa tesi consiste nel far luce riguardo il funzionamento e gli utilizzi del token YUP della dApp di Yup, che è oggigiorno una delle applicazioni decentralizzate maggiormente utilizzate. Principalmente ci siamo concentrati sull'analisi del

token YUP presente sulla blockchain di Ethereum, e in particolare abbiamo analizzato la tipologia di utenti che si interfaccia con questa dApp e le transazioni effettuate su Ethereum nel suo corso di vita. Tramite queste ultime abbiamo cercato di comprendere le azioni svolte, i DEX maggiormente utilizzati e i token YUP scambiati tra i vari utenti.

Il lavoro è organizzato in 2 fasi:

1. **Download:** in questa fase iniziale abbiamo vagliato le varie opzioni per scaricare le informazioni riguardo le transazioni Ethereum. A seguito delle nostre considerazioni, abbiamo infine implementato uno strumento per il download delle informazioni desiderate.
2. **Analisi:** in questa fase abbiamo processato i dati ed estratto le informazioni che ci hanno permesso di effettuare le nostre analisi.

Nelle prossime Sezioni descriveremo più in dettaglio le varie fasi.

3.6.1 Fase di download

Per scaricare i dati dalla blockchain di Ethereum abbiamo effettuato **web scraping** sul blockchain explorer **Etherscan**. Etherscan consente di visualizzare dettagliatamente a facilmente i dati pubblici riguardo transazioni, smart contract, indirizzi e altro ancora sulla blockchain di Ethereum. Abbiamo collezionato tutte le informazioni relative alle transazioni effettuate nell'ultimo anno, in file in formato *.json*. I dati relativi a queste transazioni comprendono tutti i campi descritti nella Sezione 2.3. Inizialmente sono stati scaricati solo i campi utili per individuare la quantità di token YUP scambiati, gli utenti che interagiscono maggiormente effettuando transazioni e le azioni eseguite più frequentemente. Successivamente, sempre tramite web scraping del sito Etherscan, abbiamo ricavato i restanti campi che ci hanno permesso di analizzare i token trasferiti all'interno delle varie transazioni.

3.6.2 Fase di analisi

Durante la fase di analisi ci siamo focalizzati sullo studio delle transazioni degli utenti, ricavate nella fase precedente. Tramite questo studio abbiamo cercato di comprendere le transazioni

svolte su Ethereum. In dettaglio, abbiamo osservato i motivi per cui un utente effettua una determinata transazione, ovvero per aggiungere o rimuovere liquidità, per effettuare scambi o trasferimenti e così via. Di conseguenza abbiamo effettuato analisi riguardanti gli utenti, sia mittenti sia destinatari, e la quantità di token YUP da essi scambiati. Inoltre abbiamo individuato quali sono gli exchanger decentralizzati più popolari all'interno dei nostri dati, e infine abbiamo riscontrato alcune problematiche, ovvero i Miner Extractable Value, che colpiscono gli utenti Ethereum. Nel Capitolo 5 sono riportate tutte le analisi eseguite e le considerazioni sui risultati ottenuti.

Capitolo 4

Implementazione

In questo Capitolo, andremo a presentare la fase di implementazione e il codice prodotto per ottenere i dati. Inizieremo presentando gli strumenti utilizzati, le librerie e i linguaggi di programmazione impiegati. Successivamente illustreremo il formato dei dati contenuti nel dataset utilizzato, descrivendone i vari campi. Mostreremo attraverso esempi la struttura di queste transazioni, sottolineando alcune diversità fra esse. Infine, affinché questa fase sia riproducibile in futuro, riporteremo il codice dei programmi utilizzati per scaricare le informazioni da noi ottenute.

4.1 Strumenti utilizzati

In questa Sezione presentiamo, in ordine, gli strumenti utilizzati per scaricare i dati ed effettuare le analisi condotte su di essi, con l'obiettivo di rendere più comprensibile il lavoro alla base della nostra tesi e i procedimenti seguiti durante la sua realizzazione.

4.1.1 Etherscan

Etherscan è un block explorer, il più popolare per Ethereum. Esso ci consente di visualizzare tutti i dati pubblici riguardo transazioni, smart contract e indirizzi sulla blockchain di Ethereum. Tutte le interazioni su Ethereum sono pubbliche, ed Etherscan agisce come un motore di ricerca che ci consente di visualizzarle. Questo block explorer non richiedere una

registrazione, a meno che non si voglia usufruire di funzionalità aggiuntive, come il servizio di notifica, l'accesso agli strumenti per sviluppatori e la creazione dei data feed. Etherscan non fornisce un wallet di Ethereum da utilizzare e non è possibile fare trading. Il suo unico scopo è agire come fonte di informazioni on-chain sulla blockchain e come database per gli smart contract. Attraverso Etherscan è possibile comprendere più nel dettaglio le transazioni. Su questo motore di ricerca è possibile cercare tutti i dettagli di una transazione attraverso il proprio codice unico identificativo, e capire se quest'ultima è andata a buon fine. Inoltre è possibile effettuare ricerche di indirizzi che possono rappresentare sia contratti, sia utenti. Dopo aver effettuato la ricerca di un indirizzo, è visualizzabile l'elenco di tutte le transazioni ad esso associate. Queste ultime possono essere contrassegnate come **In** o come **Out**. Nel primo caso il simbolo *In* indica che l'indirizzo è sul lato ricevente, viceversa per il secondo caso. Tramite Etherscan possiamo anche visualizzare una panoramica riguardo il saldo del wallet associato ad un determinato indirizzo e ricercare gli smart contract. Tramite l'individuazione di questi ultimi è possibile verificare che il contratto sia quello corretto in caso di invio di fondi, e si possono aggiungere nuovi token ad un determinato wallet. L'indirizzo degli smart contract, inoltre, contiene la logica del token, quindi come vengono effettuati i trasferimenti del token e altri comportamenti. Ricercando questi contratti è possibile vedere anche i dettagli del loro saldo e altre informazioni generali come l'indirizzo creatore di un determinato contratto e la transazione in cui è avvenuta questa creazione. Tramite l'utilizzo del sito **Etherscan**¹, è stato possibile scaricare le transazioni analizzate e condurre ricerche per individuare informazioni più dettagliate, come le transazioni e il periodo in cui sono stati creati determinati contratti e il loro creatore.

4.1.2 Selenium in Java

Il linguaggio di programmazione utilizzato per eseguire lo scraping del sito Etherscan, è **Java**. Con questo linguaggio di programmazione abbiamo usufruito di **Selenium** [4]. Quest'ultimo è un tool open source per la gestione automatizzata dei browser, utilizzato come framework di testing. Selenium è in realtà composto da diversi strumenti: Selenium IDE, Selenium Builder,

¹URL: <https://etherscan.io/token/0x69bbc3f8787d573f1bbdd0a5f40c7ba0aee9bcc9>

Selenium Grid, Selenium WebDriver. Selenium IDE è un ambiente di sviluppo integrato per i test, che consente di creare, registrare e fare il debug dei test. Selenium Builder è un'estensione che traduce il comportamento utente in comandi per strutturare i test Selenium. Selenium WebDriver è lo strumento che simula il comportamento di un utente reale all'interno di un browser e viene utilizzato per eseguire localmente o in remoto i test all'interno dei browser supportati. Nel nostro caso il browser utilizzato è Chrome, ma non avendo la necessità di un server per eseguire i comandi, il WebDriver può interagire con i browser più diffusi. Selenium Grid potenzia WebDriver, consentendo di eseguire contemporaneamente più test su più browser e sistemi operativi. I test sono registrati in Selenese, uno speciale linguaggio di scripting ideato appositamente per Selenium. Infine Selenium è uno strumento multipiattaforma, che può essere eseguito su Windows, come nel nostro caso, o anche su Linux e Mac. In particolare, nel nostro lavoro di tesi, è stato utilizzato selenium-server-standalone-3.141.59.

4.1.3 Libreria Guava in Java

Oltre a Selenium, abbiamo utilizzato la libreria Guava [3]. Quest'ultima copre molti aspetti: le collection, la cache, la gestione della concorrenza, il processazione delle stringhe, l'I/O e così via. Questa libreria permette di creare collection in modo rapido, introduce nuovi metodi come `HashCode`, `toString` e `compareTo`, le MultiMaps per memorizzare valori multipli in corrispondenza di una stessa chiave, e le HashSet thread-safe usando la classe Sets.

4.1.4 Matplotlib in Python

Dopo aver concluso la fase di scraping, abbiamo proseguito effettuando le analisi sui dati ottenuti. I risultati di queste indagini sono stati riportati graficamente attraverso codici scritti nel linguaggio di programmazione **Python 3.0**. La libreria che abbiamo utilizzato è la **Matplotlib** [13]. Quest'ultima è una libreria completa per la creazione di visualizzazioni statiche, animate e interattive per Python e per la libreria matematica NumPy. Matplotlib ha le seguenti funzioni:

- 1. Crea grafici di qualità di pubblicazione.
- 2. Crea figure interattive in grado di eseguire lo zoom, la panoramica e l'aggiornamento

- 3. Personalizza lo stile visivo e il layout.
- 4. Esporta in molti formati di file.
- 5. Incorpora in JupyterLab e nelle interfacce utente grafiche.
- 6. Usa una ricca gamma di pacchetti di terze parti basati su Matplotlib.

La maggior parte delle utilità Matplotlib si trova nel sottomodulo **pyplot** e di solito vengono importate con l'alias *plt*. **Matplotlib.pyplot** è una raccolta di funzioni che fanno funzionare Matplotlib come MATLAB. Ciascuna funzione pyplot apporta alcune modifiche a una figura, come la creazione della figura o di un'area di stampa in una figura, traccia alcune linee in un'area di stampa, decora la stampa con *label*, *ticks*, *titolo*, modificando i colori o la posizione delle etichetta e così via.

4.2 Implementazione Web Scraper

Discuteremo in questa Sezione l'implementazione del nostro Web Scraper, che ci ha permesso di reperire le informazioni necessarie per lo studio proposto in Sezione 5. Illustreremo le parti fondamentali dei nostri codici, con lo scopo di fornire una maggiore comprensione delle attività da noi svolte. I programmi utilizzati per effettuare il download dei dati, sono stati realizzati in linguaggio Java, attraverso l'utilizzo della libreria *Selenium* e *Guava*, e usufruendo dell'ambiente di sviluppo *IntelliJ IDEA*. Inizialmente abbiamo prodotto un codice per scaricare i differenti campi delle transazioni principali, ovvero quelle che contengono almeno un trasferimento del token YUP, attraverso il sito di Etherscan². Nella Figura 4.1 viene illustrato un esempio visivo di una pagina del block explorer utilizzato, in cui sono visibili alcune transazioni con i loro rispettivi campi che costituiscono il formato di queste prime transazioni ottenute. I campi riportati in figura verranno descritti nella Sezione 2.3. A seguire illustriamo il codice Java utilizzato per ottenere questi primi dati. Questo programma contiene tre funzioni: il *main* visualizzabile in figura 4.2, il metodo creato per ricavare ed eventualmente salvare su file le transazioni ed è presente nelle Figure 4.3, 4.4 e 4.5. E infine il metodo *wait*,

²URL: <https://etherscan.io/token/0x69bbc3f8787d573f1bbdd0a5f40c7ba0aee9bcc9>

Txn Hash	Method ⓘ	Age	From	To	Quantity
0x799fb7a470320923a...	Swap	8 days 18 hrs ago	0x74de5d4fcf63e00296...	0xc780c4bdee7a0186...	2,168.134738802588936936
0x799fb7a470320923a...	Swap	8 days 18 hrs ago	Uniswap V2: YUP	0x74de5d4fcf63e00296...	2,168.134738802588936936
0x8e9a751111dab463049...	Swap Exact Token...	9 days 14 hrs ago	0xb84969ea02fd374ad1...	Uniswap V2: YUP	293.830068266581774178
0x28b04f11da80ac6f964...	Transfer	9 days 17 hrs ago	0x972eebcb30252640c6...	0xcbd6ed4fd56d83bac9...	165,311.392996047365934841
0x412e973d64841a8cee...	Remove Liquidity...	9 days 18 hrs ago	Uniswap V2: Router 2	0x972eebcb30252640c6...	165,311.392996047365934841
0x412e973d64841a8cee...	Remove Liquidity...	9 days 18 hrs ago	Uniswap V2: YUP	Uniswap V2: Router 2	165,311.392996047365934841
0xa3b93b7010da49877e...	Add Liquidity ET...	9 days 18 hrs ago	0x972eebcb30252640c6...	Uniswap V2: YUP	165,311.392996047365936497
0xc137e87757f804b03d...	Exec Transaction	9 days 19 hrs ago	0xcbd6ed4fd56d83bac9...	0x972eebcb30252640c6...	165,311.392996047365936497
0x50b959460976a3bae7...	Swap Exact Token...	9 days 20 hrs ago	0x83de69633b281f2a49...	Uniswap V2: YUP	1,871.665948117229013791
0xfd83ba300fd7dccac6c...	Swap Exact Token...	10 days 6 hrs ago	0xf8e8f48d0e72e41a21a...	Uniswap V2: YUP	82.130275598129758504
0xb9def4b71fb7b1f9e8...	Swap Exact ETH F...	10 days 7 hrs ago	Uniswap V2: YUP	0xf8e8f48d0e72e41a21a...	82.130275598129758504
0xa33ba066e96c669c30...	Swap	10 days 12 hrs ago	0x74de5d4fcf63e00296...	Uniswap V2: YUP	253.9118
0xa33ba066e96c669c30...	Swap	10 days 12 hrs ago	0xcc993aba2ffd89db99c...	0x74de5d4fcf63e00296...	253.9118
0xae8c1265f06f014e29...	Transfer	10 days 15 hrs ago	0xbcb5dc467d09d518a0...	0x9ecbd3ddcf4800f33f...	2,141.799917510521106536
0xac7c535c8bd36ddc30...	Transfer	10 days 22 hrs ago	0xf8e8f48d0e72e41a21a...	0xcc993aba2ffd89db99c...	96

Figura 4.1: Formato transazioni

```

17 public class crawler_principali {
18     public static void main(String[] args) throws IOException {
19         //find chrome driver
20         System.setProperty("webdriver.chrome.driver", "/Users/bianc/Desktop/Tesi YUP/chromedriver.exe");
21         save_trans();
22     }

```

Figura 4.2: Metodo main

riportato nell'immagine 4.6. Nel *main* della Figura 4.2 viene semplicemente gestita l'inizializzazione del driver Chrome nel primo passaggio, tramite l'utilizzo di *System.setProperty*. E successivamente viene chiamata la funzione *save_trans()*, di tipo *void*, per effettuare lo scraping e il salvataggio dei dati ricavati. I passaggi fondamentali del programma sono rappresentati nel metodo *save_trans()*. In questo metodo, raffigurato in Figura 4.3, inizialmente viene creata l'istanza di Chrome e successivamente viene eseguita l'inizializzazione di questo browser. Di seguito attraverso *driver.get()*, a cui passiamo l'URL del nostro sito, possiamo iniziare a navigare nella pagina richiesta. WebDriver attenderà fino al completo caricamento della pagina prima di restituire il controllo allo script. Di conseguenza vengono inizializzate variabili e strutture utili al funzionamento del nostro programma, come *JSONArray array* che sarà fondamentale per il salvataggio delle informazioni scaricate in file JSON [18]. Infine attraverso

```

24     private static void save_trans() throws IOException {
25         //creo istanza chrome
26         ChromeOptions options = new ChromeOptions();
27         //inizializzazione browser
28         WebDriver driver = new ChromeDriver(options);
29         //link della pagina
30         driver.get("https://etherscan.io/token/0x69bbc3f8787d573f1bbdd0a5f40c7ba0aee9bcc9");
31         driver.manage().window().maximize();
32
33         int i = 1;
34         int j = 1;
35         String path;
36         boolean exit = false;
37
38         JSONArray array = new JSONArray();
39
40         driver.switchTo().frame("tokentxnsiframe");
41
42         //numero di pagine da scorrere
43         path = "//*[@id='maindiv']/div[1]/nav/ul/li[3]/span/strong[2]";
44
45         WebElement pages = driver.findElement(By.xpath(path));
46         String numero_pagine = pages.getText();
47         int n = Integer.valueOf(numero_pagine);
48
49         WebElement element_trans;
50         WebElement element_function;
51         WebElement element_age;
52         WebElement element_from;
53         WebElement element_to;
54         WebElement element_quantity;

```

Figura 4.3: Prima parte del metodo save trans

l’elemento di Selenium, ovvero `driver.switchTo().frame`, possiamo passare facilmente al frame che ci interessa. Il codice procede salvando in una variabile denominata *path*, di tipo *String*, l’XPath corrispondente al numero di pagine da scorrere per prelevare tutte le transazioni eseguite su Ethereum. Successivamente questo numero viene salvato in una variabile di tipo *int*. Per salvare gli elementi da scaricare ai fini delle nostre analisi, sono state dichiarate sei variabili *WebElement*. Queste ultime servono per identificare e lavorare con gli element objects nel DOM (Document Object Model). In Figura 4.4, è visibile la parte principale del metodo *save_trans*. Quest’ultima consiste in un ciclo while effettuato sul numero di pagine da scorrere e in un secondo ciclo while interno eseguito sul numero delle transazioni presenti in una determinata pagina (corrispondente a circa 25). In questo secondo while viene inizializzata una

```

56     JavascriptExecutor js = (JavascriptExecutor) driver;
57     js.executeScript("window.scrollBy(0,350)", "");
58
59     while(j <= n){
60         FileWriter writer = new FileWriter("/Users/bianc/IdeaProjects/CrawlerE/src/trans_principali_nuove.json");
61         //scorro le trasazioni all'interno di una pagina
62         // ci sono 25 transazioni per pagina circa
63         while(i < 26 && !exit){
64             JSONObject json = new JSONObject();
65             path =("//*[@id='maindiv']/div[2]/table/tbody/tr["+i+"]/td[2]/span";
66             element_trans = driver.findElement(By.xpath(path));
67             json.put("Transaction_Hash", element_trans.getText());
68             path =("//*[@id='maindiv']/div[2]/table/tbody/tr["+i+"]/td[3]/span";
69             element_function = driver.findElement(By.xpath(path));
70             json.put("Function",element_function.getText());
71             path =("//*[@id='maindiv']/div[2]/table/tbody/tr["+i+"]/td[5]/span";
72             element_age = driver.findElement(By.xpath(path));
73             json.put("Age",element_age.getText());
74             path =("//*[@id='maindiv']/div[2]/table/tbody/tr["+i+"]/td[6]/a";
75             element_from = driver.findElement(By.xpath(path));
76             json.put("From",element_from.getText());
77             path =("//*[@id='maindiv']/div[2]/table/tbody/tr["+i+"]/td[8]/span";
78             boolean ispresent = driver.findElements(By.xpath(path)).size() > 0;
79             if(ispresent){
80                 element_to = driver.findElement(By.xpath(path));
81                 json.put("To", element_to.getText());
82             }
83             path =("//*[@id='maindiv']/div[2]/table/tbody/tr["+i+"]/td[9]";
84             element_quantity = driver.findElement(By.xpath(path));
85             json.put("Quantity", element_quantity.getText());

```

Figura 4.4: Seconda parte del metodo save trans

variabile `JSONObject json`, che corrisponde ad una raccolta non ordinata di coppie chiave e valore. Le chiavi sono stringhe univoche che non possono essere nulle, e i valori possono essere qualsiasi cosa. Il `JSONObject` viene rappresentato da una stringa racchiusa tra parentesi graffe con chiavi e valori separati da due punti e coppie separate da una virgola. In questa raccolta inseriamo, tramite il metodo `put()`, le coppie key-value rappresentanti le informazioni delle transazioni. Il parametro `value` viene recuperato attraverso il metodo `getText()` applicato agli `element objects`. Questi ultimi vengono individuati in base ai valori di localizzazione forniti e all'XPath, attraverso la `driver.findElement(By.xpath())`. Questo procedimento appena descritto viene replicato per ogni transazione presente all'interno di ogni pagina. Nell'ultima parte della funzione `save_trans`, vi è principalmente l'aggiunta dell'oggetto `json` al `JSONArray array` e quest'ultimo successivamente verrà riportato nel file JSON, tramite la `writer.append()`, al termine dell'iterazione del primo ciclo while, ovvero quando viene conclusa la scansione della prima pagine di transazioni. L'apertura in scrittura del file JSON avviene, precedentemente, prima della scansione di una determinata pagina di transazioni. Infine, attraverso il comando

```

87     array.add(json);
88
89     js.executeScript("arguments[0].scrollIntoView()",element_trans);
90     i++;
91
92     //stringa ultima transazione dell'ultima pagina
93     if(element_trans.getText().equals("0x6928aa71056ed25ea48500326754e709476fc3b01362dbdac0c9340bbd3296b")){
94         exit = true;
95     }
96 }
97 js.executeScript("window.scrollBy(0,350)", "");
98 writer.append(array.toJSONString());
99 writer.flush();
100 writer.close();
101
102 WebElement element_click;
103 if(j!=n){
104     path = "//*[@id='maindiv']/div[2]/div/div/ul/li[4]/a/span[1]/i";
105     wait(driver, 10,path);
106     element_click = driver.findElement(By.xpath(path));
107     element_click.click();
108 }
109 i=1;
110 j++;
111 }
112 driver.close();
113 driver.quit();
114 }
```

Figura 4.5: Terza parte del metodo save trans

`driver.close()` chiudiamo la finestra del browser corrente con lo stato attivo e con `driver.quit()` terminiamo l'intera sessione del browser insieme a tutte le finestre, le schede e i pop-up associati del browser. L'ultima funzione presente in questo programma è la `wait`. Questa è di tipo `void`, e viene utilizzata all'interno del metodo `save_trans`, all'interno del primo ciclo while, per simulare un attesa di 10 secondi tra due iterazioni successive. Infatti quando una pagina viene caricata dal browser, gli elementi all'interno di quella pagina possono essere caricati a intervalli di tempo diversi. Ciò rende difficile l'individuazione degli elementi: se un elemento non è ancora presente nel DOM, la funzione solleverà un'eccezione. Usando le attese, preveniamo questo problema. L'attesa fornisce un po' di gioco tra le azioni eseguite, principalmente l'individuazione di un elemento o qualsiasi altra operazione con l'elemento. Tramite lo scraping, effettuato utilizzando questo codice Java, abbiamo ottenuto i campi delle transazioni utili per le nostre prime analisi. Successivamente, abbiamo eseguito nuovamente lo scraping del nostro sito di riferimento per incrementare le nostre informazioni e intraprendere nuove indagini. Il programma utilizzato per questo secondo scraping, è simile al precedente. Esso infatti comprende gli stessi metodi del codice creato per il primo scraping, e in aggiunta è presente il

```

116     public static void wait(WebDriver driver, int time, String path){
117         try {
118             WebDriverWait wait = new WebDriverWait(driver, time);
119             wait.until(ExpectedConditions.elementToBeClickable(By.xpath(path)));
120         }catch (Exception e){
121             System.err.println("Timeout");
122         }
123     }
124 }
125 }
```

Figura 4.6: Metodo wait

```

48     public static Set<String> letturafile(Set<String> transInt) throws IOException, ParseException {
49         JSONParser parser = new JSONParser();
50         JSONArray a = (JSONArray) parser.parse(new FileReader("/Users/bianc/IdeaProjects/CrawlerE/src/InterneHash.json"));
51         for(Object o : a){
52             JSONObject jsonObject = (JSONObject) o;
53             String trans = (String) jsonObject.get("Transaction_HashInterna");
54             if(!trans.equals("") & trans != null){
55                 transInt.add(trans);
56             }
57         }
58         return transInt;
59     }
```

Figura 4.7: Lettura del file

metodo per la lettura del file, visualizzabile in figura 4.7. Quest'ultimo viene utilizzato poiché, per ottenere tutti i campi associati ad una transazione, bisogna utilizzare un nuovo URL del sito Etherscan ([https://etherscan.io/tx/”+transaction_hash](https://etherscan.io/tx/), dove per transaction_hash si intende la stringa identificativa e univoca rappresentante la transazione presa in considerazione). La funzione di lettura file viene quindi utilizzata per effettuare la lettura del file JSON, contenente le informazioni delle transazioni già acquisite, e prelevare la stringa inerente al campo transaction_hash di queste ultime. Tutte queste stringhe verranno poi salvate in una struttura *HashSet*. Successivamente, scorrendo l'*HashSet* creato, per ogni transazione contenuta viene richiamata una funzione simile alla *save_trans* della Figura 4.3. In questa funzione, diversamente dalla *save_trans*, il metodo *driver.get()* viene effettuato sul URL inerente alla singola transazione³, e tramite l'utilizzo dei *WebElement* in Selenium vengono scaricate nuove informazioni che saranno salvate in file JSON. Concludendo, nella Figura 4.8, è visibile il metodo *main*, utilizzato nel codice del secondo programma di scraping, in modo da rendere visibili le differenze con il

³[https://etherscan.io/tx/”+transaction_hash](https://etherscan.io/tx/)

```

22 public class crawlerINFOInterne {
23     public static void main(String[] args) throws IOException, ParseException {
24         Set<String> transInt = new HashSet<>();
25         transInt = letturafile(transInt);
26
27         //find chrome driver
28         System.setProperty("webdriver.chrome.driver", "/Users/bianc/chromedriver_win32/chromedriver.exe");
29
30         JSONArray jsonArray = new JSONArray();
31
32         //iteratore sull'hash_set contenente tutte le transazioni ricavate dal file json
33         java.util.Iterator<String> it = transInt.iterator();
34         //transazione da passare alla funzione per ricavare le transazioni interne
35         String transazione;
36         //creo un'istanza di Chrome
37         ChromeOptions coptions = new ChromeOptions();
38         //Initialize browser
39         WebDriver driver = new ChromeDriver(coptions);
40         while(it.hasNext()) {
41             transazione = it.next();
42             datiTransInterne(transazione,driver,jsonArray);
43         }
44         driver.close();
45         driver.quit();
46     }

```

Figura 4.8: Metodo main secondo crawler

metodo *main* del primo codice di scraping. La differenza principale consiste nell'utilizzo di un iteratore sulla struttura *HashSet*, contenente i *transaction.hash* delle transazioni, e attraverso questo iteratore iteriamo su questa raccolta. Ad ogni elemento di quest'ultima, verrà poi richiamata la funzione per effettuare il download di tutti i campi presenti all'interno della pagina riferita all'elemento preso in considerazione.

Scraping transazioni secondarie

Fino ad ora in questa Sezione, ci siamo concentrati sul delineare l'implementazione dello scraping delle transazioni principali. Nel nostro lavoro di tesi, oltre alle transazioni principali, abbiamo recuperato ulteriori transazioni per proseguire le nostre analisi. A queste ultime abbiamo assegnato il nome di transazioni secondarie. Le transazioni secondarie sono quelle transazioni correlate ad un determinato indirizzo di un utente, e possono essere sia transazioni *In* sia transazioni *Out*. Le prime hanno come indirizzo mittente uno qualsiasi e come destinatario l'indirizzo dell'utente preso in considerazione, e quindi rappresentano tutte le transazioni che l'address considerato ha ricevuto. Invece le transazioni *Out*, hanno come mit-

```

47     public static Set<String> letturafile(Set<String> hash_set) throws IOException, ParseException {
48         JSONParser parser = new JSONParser();
49         JSONArray a = (JSONArray) parser.parse(new FileReader("/Users/bianc/IdeaProjects/CrawlerE/src/Transactions.json"));
50         for(Object o : a){
51             JSONObject jsonObject = (JSONObject) o;
52             String trans_from = (String) jsonObject.get("From");
53             hash_set.add(trans_from);
54         }
55         return hash_set;
56     }

```

Figura 4.9: Metodo lettura file per le transazioni secondarie

tente l'indirizzo preso in considerazione e differenti destinatari. Quindi queste ultime indicano tutte le transazioni inviate dall'utente analizzato. Le informazioni relative alle transazioni secondarie, sono state scaricate effettuando lo scraping tramite un codice Java simile ai codici ideati per le transazioni principali. Attraverso le figure 4.9, 4.10, 4.11 e 4.12 è possibile visualizzare il programma scritto per ricavare i dati inerenti a queste transazioni. Nella funzione di lettura file, in Figura 4.9, viene aperto il file JSON contenente le informazioni riguardo le transazioni principali ricavate precedentemente. Da questo file preleviamo gli indirizzi mittenti, contenuti all'interno del campo *from* del file JSON, e salviamo questi address in una struttura *HashSet*. Questa funzione di lettura file viene richiamata all'interno del *main*, raffigurato in Figura 4.10 e restituisce la struttura *hash_set*. Con quest'ultima viene utilizzato un iteratore per iterare su ogni singolo elemento della collezione, e per ogni elemento della collezione viene eseguita la funzione *saveTransInterne()*. Questa procedura, visibile nelle Figure 4.11 e 4.12, si occupa di eseguire lo scraping del sito indicato dal URL presente nel metodo *driver.get()* (["https://etherscan.io/address/"](https://etherscan.io/address/)+*from*, dove *from* è l'indirizzo presente nella struttura *hash_set* e di cui vogliamo conoscere tutte le transazioni inviate e ricevute). Successivamente vengono eseguiti gli stessi passaggi messi in atto per scaricare le informazioni relative alle transazioni principali, e in particolare in questa funzione l'unico dato scaricato corrisponde al *transaction_hash* delle transazioni secondarie. Tutte queste stringhe univoche e identificative vengono riportate su un file JSON, attraverso l'utilizzo dei *JSONObject* e dei *JSONArray*. Inoltre anche in questo programma è presente il metodo *wait*, in Figura 4.12, in modo da simulare attese di 30 secondi tra i vari downloads e prevenire eccezioni dovute allo scorretto caricamento delle pagine del browser. In conclusione, dopo aver ottenuto tutti i *transaction_hash* delle transazioni secondarie, abbiamo scaricato tutte le altre informazioni di queste ultime, andando

```

21     public static void main(String[] args) throws IOException, ParseException {
22         //set contenente gli address delle transazioni da salvare
23         Set<String> hash_set = new HashSet<>();
24         hash_set = letturafile(hash_set);
25
26         //find chrome driver
27         System.setProperty("webdriver.chrome.driver", "/Users/bianc/chromedriver_win32/chromedriver.exe");
28
29         JSONArray jsonArray = new JSONArray();
30
31         //iteratore sull'hash_set contenente tutti i from ricavati dal file json
32         java.util.Iterator<String> it = hash_set.iterator();
33         //stringa from da passare alla funzione per ricavare le transazioni secondarie di questo indirizzo
34         String from;
35         //creo un'istanza di Chrome
36         ChromeOptions options = new ChromeOptions();
37         //Inizializzo browser
38         WebDriver driver = new ChromeDriver(options);
39         while(it.hasNext()) {
40             from = it.next();
41             saveTransInterne(driver, from, jsonArray);
42         }
43         driver.close();
44         driver.quit();
45     }

```

Figura 4.10: Metodo main per le transazioni secondarie

```

58     public static void saveTransInterne(WebDriver driver, String from, JSONArray jsonArray) throws IOException{
59         driver.get("https://etherscan.io/address/"+from);
60         driver.manage().window().maximize();
61
62         //mi sposto sulla pagina completa con tutte le transazione interne da scorrere
63         String path1 = "//*[@id=\"transactions\"]/div[1]/p/a";
64         WebElement element = driver.findElement(By.xpath(path1));
65         element.click();
66         wait(driver,30, path1);
67
68         //numero di pagine da scorrere
69         String path = "//*[@id=\"ContentPlaceHolder1_topPageDiv\"]/nav/ul/li[3]/span/strong[2]";
70         boolean piupagine = driver.findElements(By.xpath(path)).size() > 0;
71         //booleano per vedere se esiste almeno una pagina
72         if(piupagine){
73             WebElement pages = driver.findElement(By.xpath(path));
74             String num_pagine = pages.getText();
75             int n = Integer.parseInt(num_pagine);
76             int i = 0;
77             int j = 0;
78             WebElement element1;
79             JavascriptExecutor js = (JavascriptExecutor) driver;
80             js.executeScript("window.scrollBy(0,350)", "");
81             while(j < n){
82                 FileWriter writer = new FileWriter("/Users/bianc/IdeaProjects/CrawlerE/src/HashTransazioniSecondarie.json");
83                 //scorro le trasazioni all'interno di una pagina
84                 // ci sono 50 transazioni per pagina circa
85                 while(i < 50){
86                     JSONObject jsonObject = new JSONObject();
87                     path = "//*[@id=\"paywall_mask\"]/table/tbody/tr["+i+"]/td[2]/span/a";
88                     element1 = driver.findElement(By.xpath(path));

```

Figura 4.11: Metodo save_trans per le transazioni secondarie

```

89         jsonObject.put("Transaction_HashInterna", element1.getText());
90         jsonArray.add(jsonObject);
91
92         js.executeScript("arguments[0].scrollIntoView()",element1);
93         i++;
94     }
95     js.executeScript("window.scrollBy(0,350)", "");
96     writer.append(jsonArray.toJSONString());
97     writer.flush();
98     writer.close();
99
100    if(j!=n-1){
101        path = "//*[@id=\"ContentPlaceHolder1_topPageDiv\"]/nav/ul/li[4]/a/span[1]/i";
102        wait(driver, 30,path);
103        element1 = driver.findElement(By.xpath(path));
104        element1.click();
105    }
106    i=1;
107    j++;
108}
109
110
111
112 public static void wait(WebDriver driver, int time, String path){
113     try {
114         WebDriverWait wait = new WebDriverWait(driver, time);
115         wait.until(ExpectedConditions.elementToBeClickable(By.xpath(path)));
116     }catch (Exception e){
117         System.err.println("Timeout");
118     }
119 }

```

Figura 4.12: Metodo save_tran e metodo wait per le transazioni secondarie

a riutilizzare gli script precedenti, in particolare le parti di codice visibili nelle figure 4.7 e 4.8.

Formato transazioni

In prosieguo alla descrizione della fase di implementazione, effettuata per il raccoglimento dei dati, illustriamo in questa sezione i campi delle transazioni utilizzati per eseguire le nostre analisi. I campi utilizzati per le nostre indagini sono i seguenti:

- **Transaction hash:** numero che identifica univocamente una transazione eseguita
- **To:** rappresentante il destinatario di una transazione
- **From:** indica il mittente della transazione
- **Age:** rappresenta il conto dei giorni e delle ore passati dall'esecuzione della transazione
- **Quantity:** indica l'ammontare di token scambiati e/o trasferiti in una transazione
- **Method e Input Data:** questi due campi sono simili tra di loro e sono stati entrambi utilizzati per comprendere il tipo di funzione per cui è stata eseguita una transazione
- **Token Transferred:** rappresenta la lista di token trasferiti all'interno della transazione.
- **Txt Type:** indica qual è il tipo di una transazione

Tutti questi campi, contenenti le informazioni utilizzate nei nostri studi, sono stati scaricati tramite scraping, descritto nel Capitolo 4, e salvati in file JSON.

Capitolo 5

Risultati

In questo Capitolo presentiamo le analisi effettuate sui dati collezionati attraverso lo web scraper di Etherscan, presentato in Sezione 4.1. Di seguito presenteremo le caratteristiche e il formato dei dati raccolti. Infine, mostreremo i risultati ottenuti dall’analisi dei dati. Come descritto nel Capitolo 3, Yup nasce come una piattaforma sociale, utilizzata per esprimere opinioni su vari contenuti resi pubblici. Il nostro obiettivo è quello di analizzare il comportamento degli utenti di Yup attraverso le transazioni memorizzate all’interno della blockchain di Ethereum, e vedere se le loro azioni sono svolte per fini puramente sociali, oppure se è presente un forte impatto dell’aspetto economico della piattaforma. A tale scopo vedremo quali sono le funzioni svolte dagli utenti e i contratti ad esse associati. Esamineremo quali sono i protocolli utilizzati all’interno delle transazioni, e in quali archi temporali viene registrata una maggiore attività sulla piattaforma. Analizzeremo inoltre i motivi per cui gli utenti effettuano trasferimenti e/o scambi di token YUP. Successivamente analizzeremo, tramite il campo *Token Transferred* descritto nella Sezione 2.3, quali token, oltre i token YUP e se presenti, vengono maggiormente utilizzati dagli utenti all’interno delle transazioni e quali sono i protocolli maggiormente usati per eseguire i trasferimenti di token. Concludendo, andremo ad esaminare nuovi campi delle transazioni che ci aiuteranno a comprendere le tipologie di queste ultime, descritte nel Capitolo 2, in modo da capire qual è il tipo di transazione (0 Legacy o 2 EIP-1559) più frequente.

5.1 Dataset

Per effettuare le analisi riportate in questo Capitolo 5, abbiamo usufruito di due raccolte differenti di dati, il dataset $D1$ inerente alle transazioni YUP e il dataset $D2$ inerente ai trasferimenti dei tokens ERC20.

5.1.1 Dataset D1

Il dataset D1, oggetto dei nostri primi studi, contiene un insieme di transazioni YUP presenti in Ethereum, ognuna con i relativi campi, descritti nel Capitolo 4, e rilevanti per le nostre analisi. Le transazioni ottenute riguardano un periodo che va dal 16 Ottobre 2020, giorno nel quale è stata registrata la prima transazione relativa al token YUP sulla blockchain di Ethereum, al 26 Dicembre 2021, giorno in cui è stato mandato in esecuzione il codice per eseguire il web scraping del sito Etherscan. Questo dataset copre una fascia temporale di 440 giorni, e la dimensione totale è pari a 1.115 KB, per un totale di 5,247 transazioni. Nella tabella 5.1, sono riassunte le sue caratteristiche principali.

Nel dataset D1 i dati scaricati sono composti da diversi campi, elencanti di seguito.

DATASET	
Transazioni	Dimensione
5,247	1.115 KB
Data inizio	Data Fine
16 ottobre 2020	26 dicembre 2021

Tabella 5.1: Tabella Informazioni Dataset

- **Transaction Hash:** codice unico identificativo della transazione
- **From:** mittente della transazione
- **To:** destinatario della transazione
- **Function:** funzione dello smart contract invocata all'interno della transazione

- **Age:** data dell'esecuzione della transazione
- **Quantity:** numero di tokens scambiati.

Attraverso questi campi possiamo comprendere informazioni importanti contenute all'interno delle nostre transazioni, ovvero: gli spostamenti di token, gli utenti che eseguono questi spostamenti e verso chi vengono effettuati, il numero di token YUP scambiati in una transazione, il motivo per cui viene eseguita la transazione, analizzando la sua funzione, ed infine in che arco temporale è collocata. Quest'ultimo dato ci permette di effettuare studi sui periodi di maggiore frequenza di attività degli utenti, intuendo anche quali sono i principali motivi di esecuzione delle transazioni in determinati giorni e settimane. Tutte queste analisi effettuate sui dati presenti nel dataset D1, sono visualizzabili nelle sezioni 5.2, 5.3, 5.4 e 5.5.

5.1.2 Dataset D2

Il Dataset D2 contiene ulteriori dati, scaricati successivamente per comprendere in maniera più approfondita i trasferimenti di token ERC20. Le informazioni principali sono state inserite nella Tabella 5.2.

DATASET		
Transazioni principali	Transazioni secondarie	Dimensione
2,134	57,457	85.493 KB
Data inizio	Data Fine	
11 ottobre 2020	5 agosto 2021	

Tabella 5.2: Tabella Informazioni Dataset

La dimensione totale del dataset D2 è pari a 85.493 KB. Nel dataset sono contenute circa 60,000 transazioni così ripartite: 2,134 rappresentano le transazioni principali e 57,457 sono le transazioni secondarie. Per transazioni principali intendiamo tutte le transazioni inerenti al Token YUP, con i loro relativi campi. Per transazioni secondarie invece intendiamo tutte le transazioni correlate ad un determinato indirizzo. Queste possono essere sia transazioni *In*, sia

transazioni *Out*. Questi due tipi di transazioni inerenti ad uno stesso indirizzo, vengono delineate nel Capitolo 4.

Per quanto riguarda le date corrispondenti ai dati considerati, viene specificata come data di inizio la data in cui è stata eseguita la prima transazione scaricata, ovvero 11 ottobre 2020, e come data di fine quella in cui è stata eseguita l'ultima transazione scaricata, ovvero il 5 agosto 2021. Le informazioni presenti nel dataset D2 sono composte dai seguenti campi:

- **Transaction Hash:** codice unico identificativo della transazione
- **From:** mittente della transazione
- **To:** destinatario della transazione
- **Input Data:** funzione invocata all'interno della transazione
- **Txt Type:** tipo della transazione
- **Token Transferred:** lista dei trasferimenti di token avvenuti in una determinata transazione

Attraverso questi campi riusciamo ad evidenziare qual è il tipo di transazione più frequente, e in particolare quali sono i token che vengono maggiormente scambiati e/o trasferiti nelle transazioni. Le indagini condotte su questi dati sono discusse nelle Sezioni 5.9, 5.6, 5.7 e 5.8.

5.2 Utenti

Inizialmente è stata condotta un'analisi per comprendere il numero totale di utenti unici raffiguranti come destinatari e come mittenti. Dalle nostre analisi abbiamo constatato che il numero di mittenti unici è 755 in corrispondenza ai 5,247 totali, mentre i destinatari sono 86 rispetto ai 5,247 complessivi. Successivamente, abbiamo proseguito le analisi indagando sull'identità dei sender e dei receiver. Inizialmente è stata valutata la presenza di transazioni con lo stesso indirizzo nel campo mittente e nel campo destinatario. Il numero di transazioni individuate è zero, e quindi questa particolarità non è presente negli scambi. Di conseguenza, tutte le transazioni rappresentano operazioni effettuate tra soggetti discordi. Una ricerca affine alla precedente è stata svolta per capire se ci sono transazioni che hanno come mittente un indirizzo corrispondente al destinatario di un'altra transazione differente. L'individuazio-

ne di questi tipi di transazioni possono condurci a comprendere se ci sono transazioni che avvengono frequentemente tra due utenti in particolare, e in questo caso vedere quali sono le operazioni maggiormente effettuate, il periodo in cui effettuano più operazioni e la quantità di token scambiati. Nelle nostre analisi abbiamo individuato che su 5,247 transazioni scaricate, 26 mittenti appaiono come destinatari in transazioni differenti. In seguito la lista dei 26 utenti presenti sia come mittenti sia come destinatari:

- Uniswap V2: YUP
- Uniswap V2: Router 2
- YUP : Deployer
- MEV Bot: 0x000...B40
- MEV Bot: 0x000...084
- MEV Bot: 0x000...94e
- MEV Bot: 0x000...607
- MEV Bot: 0x000...880
- ZeroEx Proxy
- Hoo.com 5
- Sablier v1.1
- Sablier v1.0: Proxy
- Sablier v1.0
- 1inch v4: Router
- 1inch.exchange v2: Router
- ParaSwap P4
- Paraswap v5: Augustus Swapper Main-net
- Gnosis Protocol: GPv2Settlement
- Zapper.Fi: Uniswap V2 Zap In 2
- Gitcoin Grants: Stackbrief
- Totle Exchange
- SushiSwap: YUP
- Disperse.app
- Mirror: Chris Martz
- Ethermine: MEV Sender
- Sybil Delegate: IndoSPNetwork

Molti di questi indirizzi sono exchanger decentralizzati (DEX), ovvero, come spiegato nel Capitolo 2, servizi online decentralizzati che permettono di effettuare uno scambio peer to peer con un altro utente, senza l'intermediazione di nessuno. I DEX principali presenti in questa

lista sono Uniswap, 1inch, SushiSwap e ZeroEx. Questi exchangers sono presenti molto spesso all'interno delle transazioni YUP del nostro dataset D1, rispetto agli altri elementi della lista. Di conseguenza abbiamo notato che gli utenti protagonisti di queste 3,025 transazioni, utilizzano questi tipi di exchangers decentralizzati. In particolare il DEX più utilizzato come mittente e destinatario è **Uniswap V2**. Precisamente 1,262 mittenti e 1,493 destinatari su 5,247 scelgono di usufruire dei servizi di questo exchanger per effettuare operazioni utilizzando i token YUP. Successivamente a Uniswap, risulta essere maggiormente scelto anche **SushiSwap**. Probabilmente molti utenti preferiscono scegliere questo exchanger poiché, rispetto a Uniswap, le pool di liquidità di Sushiswap permettono di guadagnare token in modalità passiva, mettendo in stake le proprie crypto e guadagnando interessi in percentuale.

Un'informazione molto interessante nella lista mittenti-destinatari, è la presenza dei **MEV BOT**. Il Miner Extractable Value (MEV), come delineato nel Capitolo 2, può essere definito come una tecnica utilizzata dai miner per estrarre valore addizionale dalle transazioni, oltre a quello generato tradizionalmente (ovvero la reward ottenuta per il mining di un blocco e le transaction fees). Esistono molte tecniche differenti per estrarre valore aggiuntivo ed in genere consistono nell'effettuare delle conversioni mirate di token ERC20 tramite DEX. Le transazioni MEV più comuni sono le sandwitch e le flashbot, descritte nel Capitolo 2. Le prime sono transazioni soggette ad attacchi sandwitch. In un attacco sandwich un trader predatore effettua due transazioni, una prima e l'altra subito dopo una transazione vittima in sospeso. Analizzando i dati del nostro dataset D1, su 5,247 transazioni, 74 sono soggette alla problematica MEV. In dettaglio, molte di queste transazioni sono eseguite in coppia: una prima transazione che ha come mittente MEV e come destinatario Uniswap V2, seguita dalla seconda con mittente e destinatario invertiti. Inoltre, entrambe queste transazioni vengono effettuate lo stesso giorno alla stessa ora, a distanza di pochi secondi ed all'interno di esse viene trasferita la stessa quantità di token. Infine si è notato che in mezzo a due transazioni di una stessa coppia c'è sempre un'altra transazione che viene effettuata nella quale un indirizzo Ethereum non collegato né al MEV bot, né ad Uniswap, richiede lo scambio di alcuni token. In seguito riportiamo un esempio di queste coppie.

- **Prima transazione della coppia:** "Function": "0x00000006", "TransactionHash": "0xe5cc72f856c0a081837eece84067fc96201f3b4194fd33452902d8dabcbf45e9" "Quantity":

”1,329.480405941164385713”, ”From”: ”MEV Bot: 0x000...94e”, ”To”: ”Uniswap V2: YUP”, ”Age”: ”149 days 7 hrs ago”.

- **Transazione intermedia:** ”Function”: ”SwapExactETH”, ”TransactionHash”: ”0xc332546ba5a0fc74e33dbeb96527f86c7985ed2710445366a889b9ebbb1999a8” ”Quantity”: ”5,226.338992916683036526”, ”From”: ”Uniswap V2: YUP”, ”To”: ”0x69bbc3f8787d573f1bbdd0a5f40c7ba0aee9bcc9”, ”Age”: ”149 days 7 hrs ago”.
- **Seconda transazione della coppia:** ”Function”: ”Eldddhzr”, ”TransactionHash”: ”0xc649a2155f35a09cdf47c019c7d6094f6d8d3f1521334658fa4dfa8fb83047f7” ”Quantity”: ”1,329.480405941164385713”, ”From”: ”Uniswap V2: YUP”, ”To”: ”MEV Bot: 0x000...94e”, ”Age”: ”149 days 7 hrs ago”.

Tutte le 74 transazioni soggette a MEV hanno questo formato e sono raggruppate a coppie di due. Da questa indagine possiamo dedurre che queste transazioni sono soggette ad attacchi di tipo *sandwich*: infatti sembrerebbe che la transazione al centro della coppia è una transazione vittima, e che le altre due transazioni siano effettuate una subito prima e una subito dopo dal trader predatore. Inoltre tramite questa scoperta possiamo constatare che la problematica MEV continua ad essere presente anche nell’aggiornamento *EIP-1559* della blockchain Ethereum, infatti le nostre transazioni soggette a questo problema sono sia transazioni 0 (Legacy) sia transazioni 2 (EIP-1559), descritte nel 2. Tra i protagonisti delle nostre transazioni risultano anche alcuni indirizzi rappresentati applicazioni. In particolare risalta **Disperse.app**. Quest’ultima compare come mittente e come destinatario in molte transazioni. L’operazione svolta all’interno di esse è la **Disperse Token**. Questa operazione permette ad un mittente di distribuire una certa quantità di uno stesso token a più destinatari contemporaneamente, attraverso un’unica transazione. L’utente che intende effettuare questa operazione di invio simultaneo di token, effettua una transazione avente come destinatario Disperse.app e a cui trasferisce la quantità totale di token che vuole suddividere tra i vari destinatari. Successivamente vengono eseguiti vari trasferimenti di token da parte di Disperse.app, in cui la quantità totale di token ricevuta dal mittente viene divisa e sparsa verso indirizzi destinatari differenti. Questo permette agli utenti di ridurre il numero di operazioni da effettuare e velocizzare l’invio e la ricezione dei token trasferiti. Inoltre attraverso questo metodo di Disperse.app, il

mittente che esegue la transazione paga i contributori in una unica transazione, andando così a ridurre le commissioni di gas presenti su Ethereum. Per i motivi descritti, questa applicazione potrebbe incentivare gli utenti ad effettuare più trasferimenti di token o di ETH. Proseguendo, altri mittenti-destinatari frequenti sono **Sablier** e **Gnosis Protocol**. La prima compare in 31 transazioni e consente trasferimenti di denaro tra entità. Sablier detiene il denaro trasferito/ricevuto fin quando l'utente non decide di rilasciarlo sul proprio indirizzo Ethereum utilizzando la funzione *Withdraw*. Dai nostri dati notiamo che 6 transazioni con Siblier sono effettuate per eseguire la funzione *Withdraw*. Con questo risultato notiamo che Uniswap è preferibile come protocollo dagli utenti rispetto a Siblier, poiché quest'ultimo non rilascia immediatamente il denaro, ma lo trattiene e costringe gli utenti ad effettuare un'ulteriore operazione per il prelievo dei soldi, pagando di conseguenza un'ulteriore quota di gas per eseguire la transazione. Invece con Uniswap questo non accade. **Gnosis Protocol** è una piattaforma di investimento decentralizzata ed è visibile come mittente o destinatario in 39 transazioni del nostro dataset D1. Gli investitori che usano questo protocollo sono coloro che ritengono che il mercato delle previsioni¹ diventerà uno strumento essenziale nei mercati dei capitali e nella scienza dei dati. Una feature molto importante offerta da Gnosis, è un meccanismo recentemente introdotto per combattere il fenomeno MEV discusso prima. Inoltre Gnosis consente agli utenti di scambiare i risultati di determinati eventi. Dai dati del dataset D1 risulta che sono in numero minori le transazioni in cui gli utenti possono scambiarsi risultati di previsioni di determinati eventi, e da questi primi risultati notiamo che molti utenti di Yup non utilizzano la piattaforma per le iterazioni sociali. Sono invece molto frequenti gli exchanger decentralizzati come Uniswap V2, oppure le applicazioni come Disperse.app, che permettono agli utenti di usufruire della piattaforma non solo per scopi sociali, ma anche per scopi economici.

¹Un mercato di previsione utilizza le previsioni degli utenti per aggregare informazioni su eventi futuri. Gli utenti scambiano gettoni che rappresentano il risultato di un determinato evento. Poiché è più probabile che alcuni risultati si verifichino di altri, questi token finiscono per avere valori diversi nel mercato. Una volta che l'evento si verifica, i gettoni che rappresentano il risultato finale ricevono il valore pieno mentre il resto vengono persi.

5.2.1 Analisi Mittenti

In questa Sezione descriviamo un’analisi effettuata sugli utenti, ovvero l’analisi dei mittenti delle transazioni. Questo studio è stato condotto sul nostro dataset D1 composto da 5247 transazioni, per comprendere quali sono i protagonisti che effettuano più transazioni YUP su Ethereum. I risultati hanno rivelato che il mittente più ricorrente corrisponde al protocollo **Uniswap V2: YUP**. **Uniswap V2: YUP** compare come mittente in 1262 transazioni. La maggior parte di queste sono effettuate per eseguire funzioni di Swap, ovvero di trasformazione di una data quantità di token ERC20 in un altro token ERC20. Uniswap V2 risulta essere il protocollo preferito dagli utenti per effettuare scambi. Inoltre un’altra caratteristica che, probabilmente, spinge gli utenti ad usufruire di Uniswap per gli scambi, è la creazione di liquidity pool tra coppie di token ERC20/ERC20 e non solo coppie ERC20/ETH. Questo risulta essere vantaggioso per gli scambi poiché permette di scambiare tra di loro token ERC20, invece di dover necessariamente prima trasformare i token in possesso in ETH e poi successivamente trasformare l’ETH ottenuto in token desiderati, andando così a ridurre le tasse da pagare per i trasferimenti. Un altro mittente ricorrente è il **Null Address**, ovvero 0x0 che è un indirizzo speciale che serve per la creazione o la distruzione di token ERC20. È un’abbreviazione per l’indirizzo di genesi 0x000. Tramite questo indirizzo un utente potrebbe creare un proprio token ERC20 da mettere in circolazione sulla rete di Ethereum, e successivamente potrebbe decidere di eliminarlo definitivamente dalla circolazione. Questo mittente compare in 845 transazioni. Tutte queste transazioni sono effettuate eseguendo la funzione Push Inbound Messages, che si occupa di inviare i messaggi in entrata. Un ultimo indirizzo maggiormente ricorrente come mittente è rappresentato da **Disperse.app**, descritto nella Sezione 5.2. Questo mittente compare 840 volte per effettuare trasferimenti multipli di uno stesso token.

5.2.2 Analisi dei Destinatari

Dalle ricerche condotte sui destinatari abbiamo ottenuto come risultato che i due indirizzi maggiormente ricorrenti corrispondono nuovamente a **Uniswap V2: YUP** in 1492 transazioni, e **Uniswap V2: Router 2** in 62 transazioni. Il primo, ovvero **Uniswap V2: YUP**, è presente

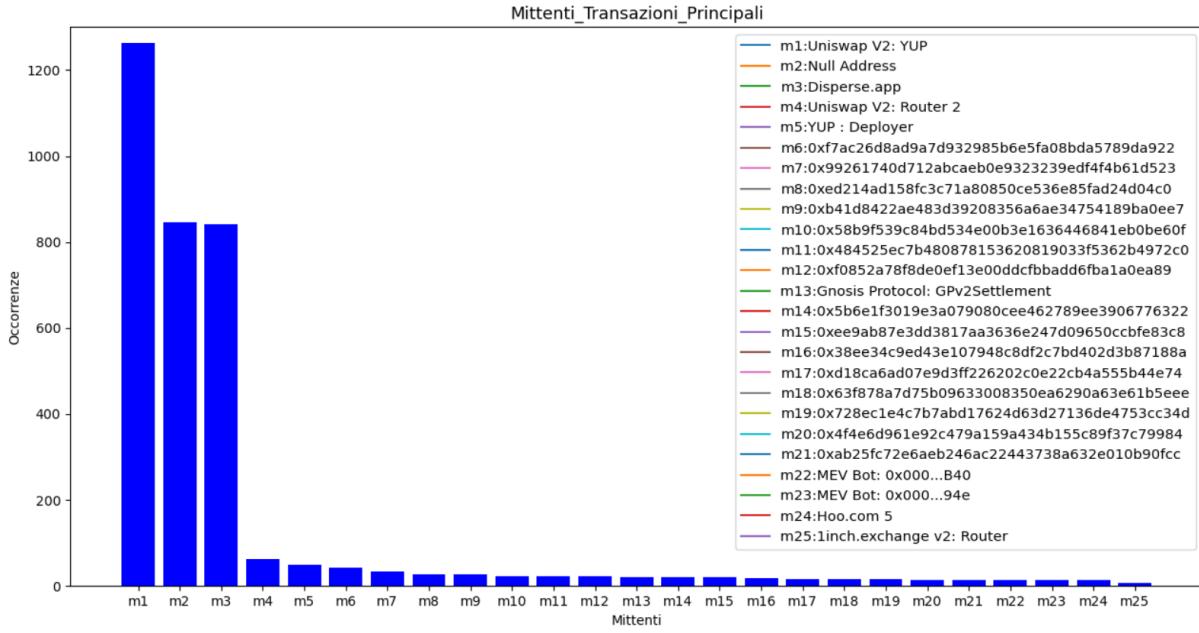


Figura 5.1: Mittenti transazioni

in transazioni che si occupano dell'aggiunta di liquidità tramite il metodo *Add Liquidity* oppure dello scambio di token o eth, tramite la funzione *Swap Exact Eth* e *Swap Exact Token*. La *Add Liquidity* in queste transazioni fornisce liquidità al pool di Uniswap V2. Con ogni transazione di fornitura di liquidità, il mittente riceve automaticamente dei token di liquidità del pool di Uniswap. Questi token tracciano il contributo dei fornitori di liquidità del pool e vengono utilizzati per distribuire le commissioni di transazione accumulate nel tempo dai fornitori di liquidità. Il secondo, ovvero **Uniswap V2: Router 2**, invece è distribuito sul contratto **0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D**. Il Router 2 compare come destinatario nelle transazioni che si occupano della rimozione della liquidità, attraverso la *Remove Liquidity*, e hanno come mittente Uniswap V2: YUP. Ciò significa che una parte degli utenti che in precedenza hanno aggiunto liquidità al pool di Uniswap V2, hanno deciso di riprendere le somme depositate più eventuali commissioni maturate, e per effettuare questa operazione devono bruciare i liquidity token. Come destinatario è presente il Contratto Router 2. Infatti quest'ultimo, attraverso la logica di routing, invia i token al contratto, e di conseguenza all'utente, corretto che ha richiesto di riavere la somma precedentemente depositata. Successivamente l'utente potrà accedere direttamente alla propria liquidità. Oltre questi due destinatari, abbiamo anche **Null Address** in 48 transazioni che eseguono la *Send Token*, attraverso cui il

destinatario riceve i token che intenzionalmente un utente vuole rimuovere dalla circolazione. Queste transazioni possono rappresentare due operazioni, una intenzionale e una non intenzionale. L'operazione *intenzionale* si ha nel caso in cui gli utenti delle nostre transazioni hanno deciso di bruciare i propri token per l'integrazione della deflazione in un sistema oppure per la stabilizzazione dei prezzi. Per azione *non intenzionale* intendiamo il caso in cui gli utenti considerati inviano per errore i token all'indirizzo di genesi, e di conseguenza non sono più in grado di recuperarli. Dai nostri dati non possiamo concludere con certezza quali sono gli utenti che hanno svolto intenzionalmente questa operazione o meno; ma controllando i mittenti abbiamo notato che ci sono 13 mittenti unici che svolgono le 48 transazioni. Tra questi 13 mittenti, 11 eseguono solo una volta l'operazione di invio dei token al null address e questo ci porta a pensare che potrebbe essere una transazione involontaria, poiché molti portafogli degli utenti hanno inizialmente impostato l'indirizzo di genesi come indirizzo predefinito. Mentre per quanto riguarda le restanti transazioni, aventi come destinatario l'indirizzo di genesi, 6 sono svolte dall'indirizzo **0x99261740d712abcaeb0e9323239edf4f4b61d523** e le altre 28 da **YUP : Deployer**. Probabilmente questi ultimi due indirizzi potrebbero aver svolto intenzionalmente l'operazione di *Send Token* verso l'indirizzo nullo per eliminare dalla circolazione alcuni token. Il penultimo destinatario ricorrente è **YUP:Deployer**, ovvero l'account che ha rilasciato il contratto del token YUP su Ethereum, presente in 52 transazioni. Queste ultime svolgono principalmente la funzione Push Inbound Messages e hanno come mittente il Null Address. Riferendoci alle 28 transazioni, citate in precedenza nella Sezione 5.2, con mittente YUP:Deployer e destinatario Null Address, notiamo che ci sono degli scambi reciproci fra questi due indirizzi. Infine abbiamo anche **Hoo.com 5**. Quest'ultimo rappresenta una piattaforma che implementa servizi economici per blockchain. Ricorre come destinatario in 43 transazioni, tutte eseguite svolgendo la funzione Transfer. Quest'ultima piattaforma ci conferma ulteriormente che i nostri utenti eseguono transazioni principalmente per scopi finanziari.

Conclusioni analisi utenti in Yup

I risultati ottenuti attraverso le indagini svolte sui mittenti e i destinatari delle transazioni YUP, rivelano nuove informazioni riguardo questa piattaforma. Inizialmente Yup era stata ideata per

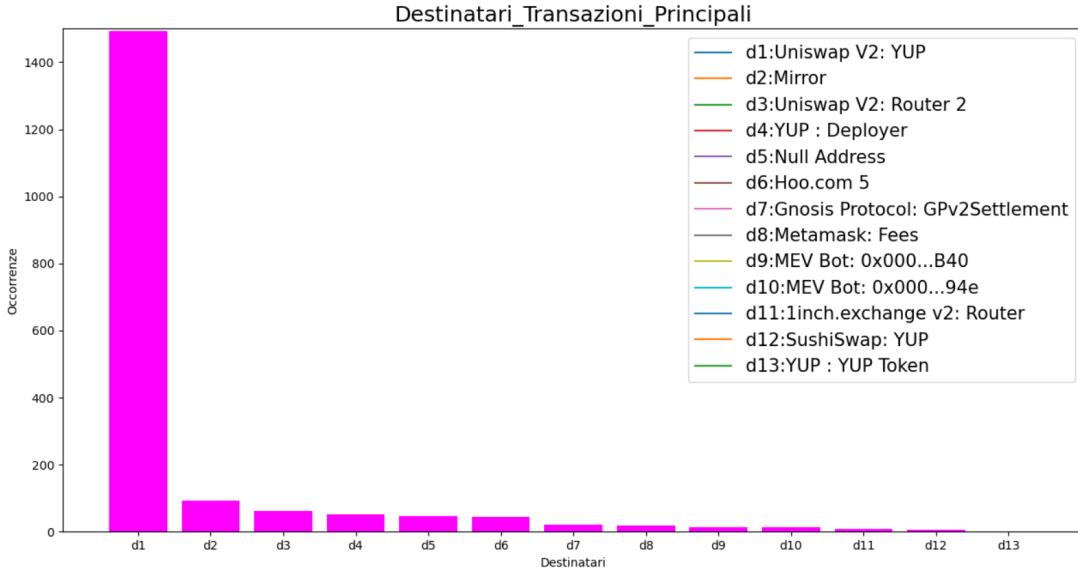


Figura 5.2: Destinatari transazioni

essere un protocollo di consenso sociale, in cui le operazioni maggiormente svolte dagli utenti consistevano nell'esprimere opinioni riguardo contenuti pubblici. Ma analizzando le transazioni YUP presenti in Ethereum abbiamo individuato la presenza di un traffico economico, in cui la quasi totalità degli utenti sembra essere interessata all'economia digitale. Infatti molti mittenti sembrano essere informati riguardo l'utilizzo e i vantaggi degli exchanger decentralizzati, scegliendoli sempre per effettuare i loro scambi in modo da avere il controllo totale sui propri fondi. Inoltre chi acquisisce Yup conosce bene anche i vari metodi messi a disposizione dai differenti sistemi decentralizzati per effettuare scambi e trasferimenti di token; infatti molti utenti effettuano scelte che li conducono a trarre maggiori profitti: come l'utilizzo di Disperse.app per risparmiare tempo e commissioni nell'invio di token, oppure l'utilizzo di Uniswap V2 per ridurre le tasse da pagare per i trasferimenti ed eventualmente la scelta di utilizzare Gnosis Protocol per cercare di prevenire attacchi MEV.

5.2.3 Sender giornalieri e settimanali

In prosieguo alla rilevazione di utenti che usufruiscono delle transazioni YUP per scopi monetari; abbiamo condotto un'analisi per individuare il numero di mittenti attivi sulla piattaforma in un determinato giorno e in una settimana, in modo da comprendere qual è il periodo in

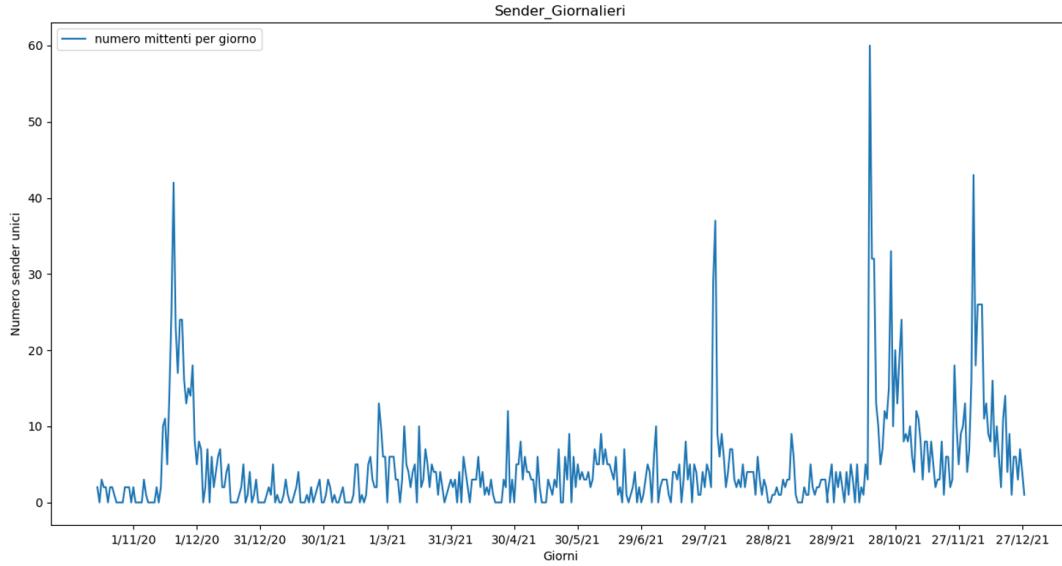


Figura 5.3: Sender unici giornalieri

cui viene registrata una maggiore attività e lo scopo di quest’ultima. I dati raccolti vengono raffigurati nelle Figure 5.3 e 5.4. I risultati dimostrano che la maggior parte delle iterazioni giornaliere da parte di mittenti differenti si è verificata il 16 ottobre 2021, in cui è stato registrato un numero di sender pari a 60. Mentre la quota maggiore di iterazioni settimanali da parte di mittenti diversi è stata registrata nella settimana compresa tra il 19 novembre 2020 e il 25 novembre 2020, con un numero di sender pari a 110. Quindi in questi periodi di tempo, si può notare che le transazioni sono in gran parte eseguite da utenti differenti e la funzione eseguita maggiormente è l’operazione di *Swap*. In altri giorni invece, per esempio il 29 ottobre 2021 e la settimana compresa tra il 4 novembre e 10 novembre 2021, essendoci pochi sender unici attivi, crediamo che le transazioni siano effettuate da uno stesso mittente o comunque da pochi utenti. Il 29 ottobre 2021 sono attivi 19 sender unici in 48 transazioni, e tra questi l’indirizzo maggiormente attivo, protagonista di 15 transazioni, è l’indirizzo nullo che effettua tutte funzioni di Push Inbound Messagges.

5.2.4 Receiver giornalieri e settimanali

Proseguendo le nostre analisi siamo passati ad analizzare i receiver, effettuando la stessa indagine condotta per i sender, in modo da comprendere il numero di receiver unici giornalieri e

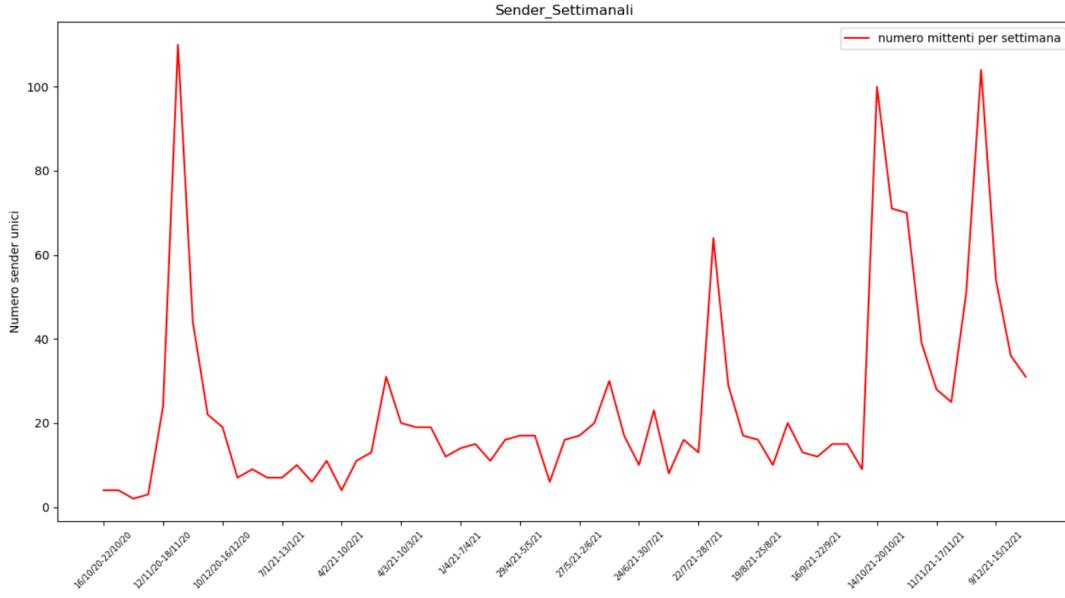


Figura 5.4: Sender unici settimanali

settimanali, i cui numeri ottenuti sono riportati nei grafici delle Figure 5.5 e 5.6. Questo studio è stato effettuato per capire se in un determinato giorno o in una determinata settimana, le transazioni sono dirette a differenti utenti o ad uno in particolare. I dati prelevati dai grafici nelle Figure 5.5 e 5.6, mostrano che le transazioni eseguite il giorno 11 giugno 2021 sono destinate per la maggior parte a destinatari differenti, in particolare ci sono 50 receiver diversi, al contrario dei giorni restanti in cui sembra che una buona parte delle transazioni effettuate siano rivolte ad uno stesso destinatario o comunque a pochi destinatari. Facendo riferimento alla statistica settimanale, le conclusioni conducono alla settimana del 10 giugno e 16 giugno 2021. Questo è il periodo temporale in cui viene registrato il valore più alto di receiver differenti, con un numero pari a 52. Quindi le transazioni effettuate in questa settimana probabilmente sono tutte indirizzate a utenti diversi, al contrario delle transazioni effettuate nei rimanenti giorni e settimane, che segnalano un numero inferiore di receiver, portando a credere che le transazioni inviate in quel periodo siano indirizzate a stessi destinatari. Il giorno con un minor numero di receiver unici, pari a 4, è il 19 ottobre 2021. In questo giorno il destinatario più ricorrente è Uniswap V2: YUP, che in 14 transazioni eseguite compare per 11 volte svolgendo la funzione di Swap Exact Token.

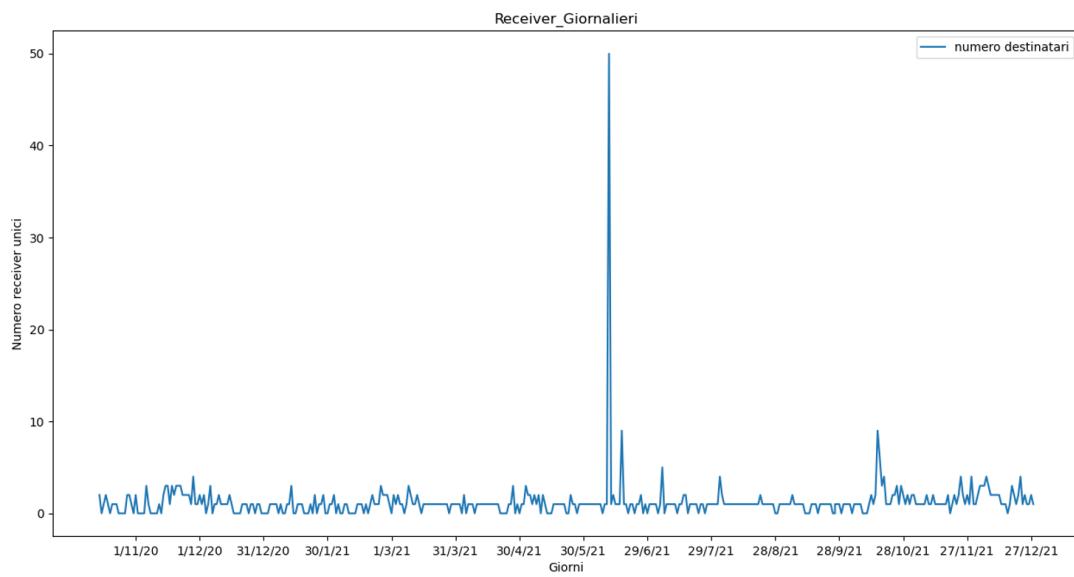


Figura 5.5: Receiver unici giornalieri

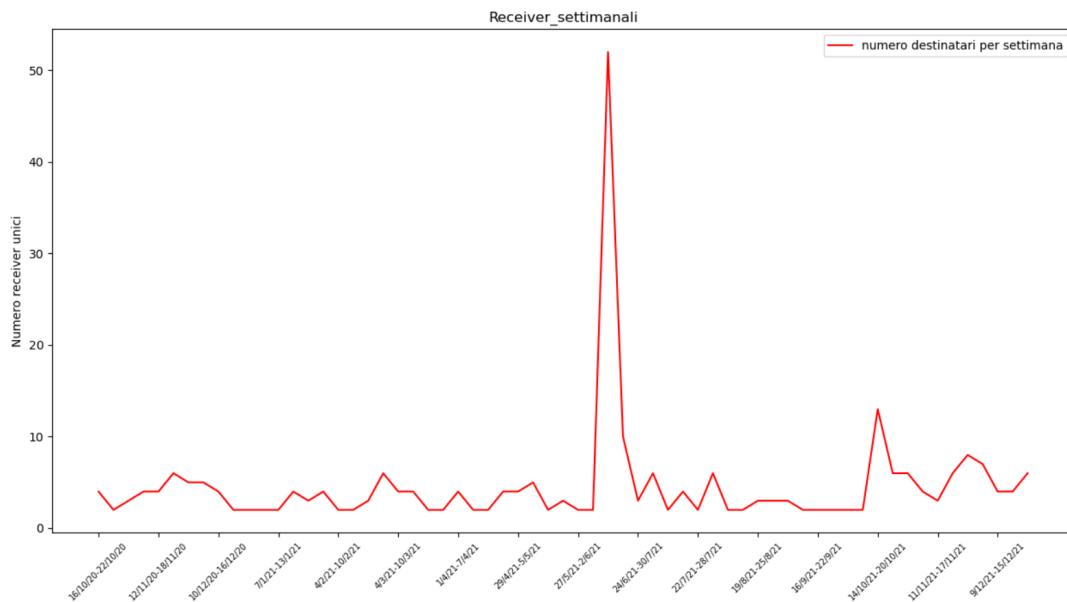


Figura 5.6: Receiver unici settimanali

Conclusioni analisi temporali

Attraverso questo studio abbiamo notato che i periodi di maggiore attività da parte degli utenti corrispondono a periodi in cui vengono effettuate operazioni di scambio, ovvero la funzione *Swap*. Quest’ultima consente di effettuare scambi di token e/o ETH. In particolare è presente la *Swap Exact Token* in cui gli utenti scelgono di stabilire la quantità esatta di token da barattare. Questo ci porta a dedurre che i periodi di maggiore attività sulla piattaforma sono dovuti ad attività finalizzate ad un ritorno economico. Inoltre anche nei giorni in cui è registrata una minore attività, quest’ultima è comunque eseguita dagli utenti del dataset D1 per condurre operazioni finanziarie. Questo ci permette di concludere che gli utenti analizzati corrispondono a persone magari esperte del settore o che comunque iniziano a addentrarsi nell’ambiente del trading online.

5.3 Transazioni

5.3.1 Numero transazioni giornaliere e settimanali

Dopo aver compreso il periodo di maggiore iterazione da parte degli utenti, e l’obiettivo prevalente delle azioni svolte da questi ultimi; ci siamo concentrati sulla distribuzione temporale del numero di transazioni uniche eseguite giornalmente e settimanalmente, con lo scopo di gettare luce sulla frequenza di scambio dei token YUP nella blockchain di Ethereum. I risultati, visualizzabili nelle Figure 5.7 e 5.8, ci permettono di stabilire quali sono i giorni e le settimane in cui la piattaforma ha registrato il maggior e il minor numero di transazione effettuate dagli utenti. Il nostro studio mostra che il massimo numero di transazioni in una giornata, pari a 848, è stato riscontrato il giorno 17 giugno 2021, mentre il minor numero, pari a 42, il giorno 6 novembre 2021. Considerando le analisi settimanali invece, il numero più elevato di transazioni uniche eseguite, pari a 395, è rilevato nella settimana compresa tra il 19 novembre e il 25 novembre 2020, mentre il minor numero pari a 86 è registrato nella settimana fra il 25 febbraio e il 3 marzo 2021. Successivamente abbiamo analizzato le transazioni eseguite il 17 giugno 2021 e abbiamo notato che, in queste 848 transazioni, vengono eseguite solo 6 funzioni. Il metodo più ricorrente corrisponde all’operazione di Disperse Token, effettuata in 842 tra-

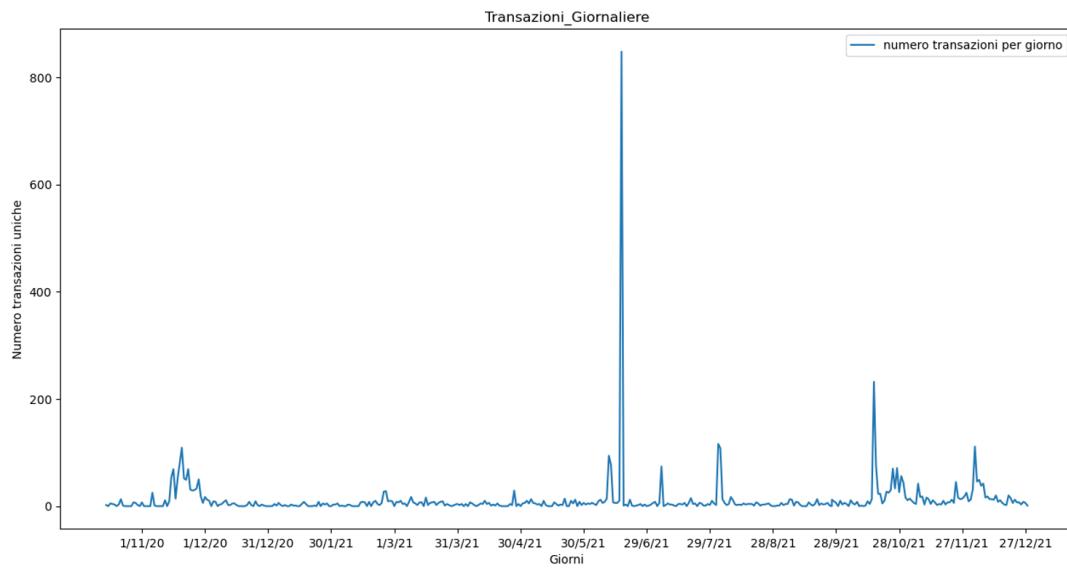


Figura 5.7: Transazioni giornaliere

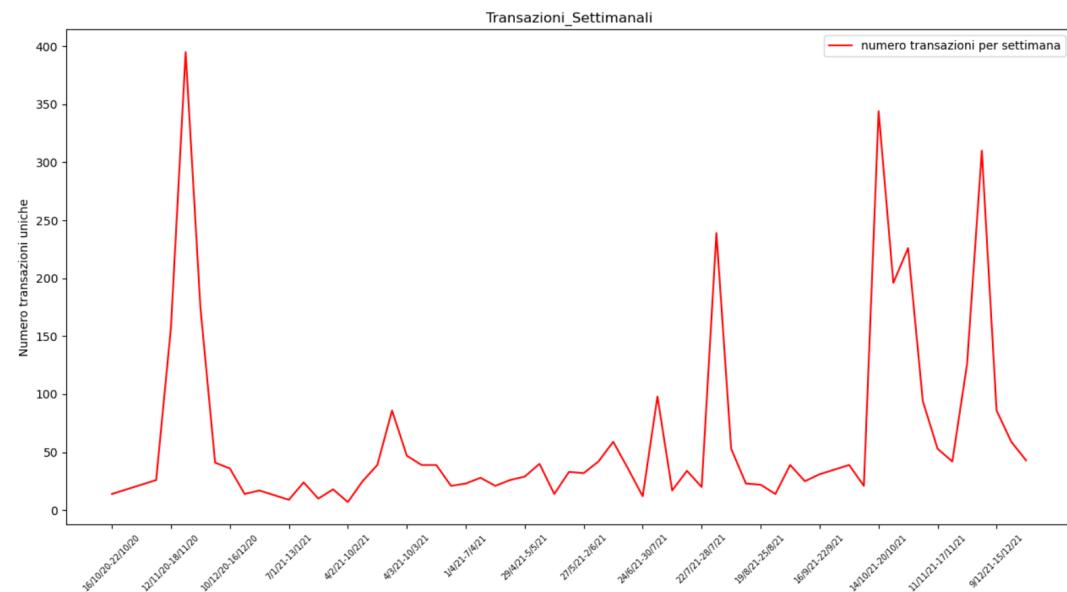


Figura 5.8: Transazioni settimanali

sperimenti di token. Le restanti funzioni invece corrispondono a differenti metodi di scambio o trasferimento di token. Questo studio ci porta a confermare che gli utenti, oggetto delle nostre ricerche, rappresentano gente esperta del settore dell'economia digitale; infatti le transazioni presenti nel nostro dataset D1 sono tutte transazioni eseguite per fare trading o staking online, e le scelte che intraprendono per effettuare queste transazioni non sembrano essere decisioni prese da persone disinformate.

5.4 Function transazioni

Successivamente alla scoperta di transazioni prevalentemente eseguite per trarre profitti; in questo Paragrafo analizziamo le funzioni con cui vengono effettuate queste transazioni in modo da ampliare la nostra visione sulle azioni maggiormente utilizzate dai nostri utenti per addentrarsi nel mondo del trading online.

In totale, su tutte le 5,247 transazioni sono presenti 153 funzioni uniche. I metodi maggiormente ricorrenti, visualizzabili in Figura 5.9, sono i seguenti:

- **Swap Exact Token** rappresenta il metodo utilizzato per il token swap, ovvero lo scambio di token attraverso liquidity pool. Questo metodo in particolare permette di specificare la quantità esatta di uno dei due token da scambiare.
- **Swap Exact ETH** indica la funzione utilizzata per scambiare token attraverso liquidity pool, nel quale uno dei due token del liquidity pool è ETH o uno dei surrogati (esempio: il token ERC20 Wrapped Ether). Questo metodo in particolare permette di specificare una quantità esatta di ether da utilizzare nello scambio.
- **Disperse Token** è la funzione utilizzata nelle transazioni per distribuire token a più indirizzi differenti.
- **Push Inbound Messages** rappresenta il metodo utilizzato per richiedere l'invio di messaggi in entrata.
- **Transfer** è il metodo utilizzato per trasferire ether e corrisponde alla funzione maggiormente utilizzata per questa tipologia di operazioni. Inoltre si occupa di spostare i token

dalla fornitura totale a qualsiasi utente individuale che acquista durante la fase della ICO (Initial Coin Offering).

- **Swap** è la funzione di scambio istantaneo tra token di Ethereum.
- **Multisend Token** tramite cui è possibile distribuire token, Ether e ERC20 a più indirizzi di portafoglio contemporaneamente con un'unica commissione di transazione. Inoltre permette di risparmiare denaro nella distribuzione delle risorse di Ethereum.
- **Add Liquidity ETH** consiste nell'aggiungere liquidità ad un pool, e quindi richiede il deposito di un valore equivalente di token ETH ed ERC20 nel contratto di scambio associato al token ERC20.
- **Swap ETH For Exact Token** è il metodo utilizzato per lo scambio di ETH e permette di specificare la quantità esatta di token con cui barattare gli ETH.
- **Remove Liquidity** è la funzione opposta alla Add Liquidity, e viene effettuata da un utente quando decide di riprendersi le somme depositate precedentemente più eventuali commissioni maturate.
- **Sell To Uniswap** che permette agli utenti di vendere i propri token.
- **Swap Token For Exact ETH** è il metodo utilizzato per scambiare i token e permette di specificare la quantità esatta di ETH con cui si vuole effettuare lo scambio.
- **Send Token** utilizzata dai mittenti per l'invio dei token ad un determinato destinatario.

Da questo insieme di funzioni ottenute, possiamo dedurre che l'operazione maggiormente richiesta dai nostri utenti è un'operazione di *transfer* o di *swap*. In dettaglio, sono preferite le operazioni in cui vengono scambiati un numero esatto di token o di ETH. Attraverso questi metodi l'utente preferisce conoscere i dettagli del suo scambio, ovvero sapere il numero esatto di ETH che può guadagnare barattando una certa quantità dei suoi token, o viceversa nel caso in cui scambiasse ETH per ottenere una determinata quantità di token. In questo modo l'utente risulta essere guidato nell'amministrazione dei suoi fondi e nella decisione delle operazioni da svolgere. Inoltre altre operazioni effettuate dai nostri utenti, consistono nell'aggiungere e

rimuove liquidità ad un pool tramite *Add Liquidity* e *Remove Liquidity*. Molti utenti quindi interagiscono con la piattaforma solo per depositare i propri fondi in un liquidity pool per un determinato periodo di tempo e ritornare solo per riprendere i propri interessi. Di seguito sono presenti anche funzioni come la *Send Token* per inviare una quantità di token ad un determinato indirizzo, ma molti utenti preferiscono utilizzare la *Multisend token* o la *Disperse Token* per trasmettere token a più indirizzi, piuttosto che eseguire più volte la *Send Token*. Questa scelta ci fa pensare che molti utenti conoscono bene le varie operazioni disponibili e tramite questa loro conoscenza possono intraprendere la scelta giusta per ottenere un maggiore guadagno durante l'invio di token, risparmiando su eventuali commissioni ed evitando perdite di tempo.

Per comprendere maggiormente le scelte degli utenti intraprese per l'esecuzione di queste operazioni, abbiamo proseguito l'analisi cercando di comprendere quali sono i contratti che mettono a disposizione queste funzioni.

Elenchiamo di seguito i contratti ottenuti dai nostri risultati:

- **Uniswap V2: Router 2**²: mette a disposizione le funzioni: **Swap Exact ETH** in 450 transazioni, **Swap Exact Token** in 1,241 transazioni, **Swap Token For Exact ETH** in 60 transazioni, **Swap ETH For Exact Token** per 126 transazioni, **Add Liquidity** in 143 transazioni e la **Remove Liquidity** in 124 transazioni.
- **Contract 0x8ba8e74c56551c639c1f5401fda1be37f531d5bd** presente in 777 transazioni che eseguono la funzione **Push Inbound Messagges**.
- **Disperse.app**³, presente in 499 transazioni che eseguono la funzione **Disperse Token**.
- **YUP : YUP Token**⁴, associato alla funzione **Transfer** ricorrente 300 volte.
- **Metamask: Swap Router**⁵, è il contratto più ricorrente nella **Swap**.

²Contract 0x7a250d5630b4cf539739df2c5dacb4c659f2488d

³Contract 0xd152f549545093347a162dce210e7293f1452150

⁴Contract 0x69bbc3f8787d573f1bbdd0a5f40c7ba0aee9bcc9

⁵Contract 0x881d40237659c251811cec9c364ef91dc08d300c

- **CoinTool: MultiSender**⁶ presente nelle 154 transazioni aventi come metodo **Multi-send**.
- **0x: Exchange Proxy**⁷ nella funzione **Sell To Uniswap** presente in 105 transazioni.
- **Contract 0x7bb59045a304daba60b0680b478ce27ceadab7ad** per le transazioni che eseguono il metodo di invio dei token, ovvero **Send Token**, presente in 24 transazioni.

I risultati dimostrano che in ben 2,144 transazioni è abituale il contratto **Uniswap V2: Router**. Inoltre le indagini hanno manifestato che la maggior parte delle transazioni svolte dagli utenti in Ethereum, vengono svolte per eseguire uno swap di token ERC20 in 1,243 transazioni o di ETH in 452 transazioni, oppure operazioni di transfer. Queste operazioni risultano molto frequenti poiché sono metodi standard del token ERC20, descritto nel Capitolo 2, per gestire i trasferimenti di fondi. Infatti il token YUP è un token ERC20 e perciò gli utenti di Yup possono attingere facilmente a queste funzioni. Inoltre, essendo Uniswap V2 molto frequente in queste operazioni, possiamo dedurre che la maggior parte dei token vengono scambiati attraverso exchanger decentralizzati. Questi ultimi non rappresentano più un concetto di nicchia, ma risultano essere molto utilizzati dagli utenti poiché consentono alle persone di scambiare risorse senza passare attraverso gli exchange centralizzati. Questo ci porta a pensare che gli utenti preferiscono utilizzare sistemi decentralizzati, descritti nel Capitolo 2, per agire in modo anonimo e autonomo, mantenendo la custodia dei propri fondi.

5.5 Ammontare Tokens YUP

Concludiamo l'analisi dei dati del nostro Dataset D1 spostando la nostra attenzione sull'analisi del campo "Quantity" delle transazioni. Questo campo rappresenta il numero di tokens scambiati in ogni transazione. In particolare, dopo aver compreso che molti utenti di Yup utilizzando la piattaforma Ethereum principalmente per scambiare token, analizziamo l'ammontare di token YUP che vengono barattati in una transazione con lo scopo di comprendere possibili meccaniche nascoste dietro lo scambio di questi token. Dalla Figura 5.10 deduciamo

⁶Contract 0xcec8f07014d889442d7cf3b477b8f72f8179ea09

⁷Contract 0xdef1c0ded9bec7f1a1670819833240f027b25eff

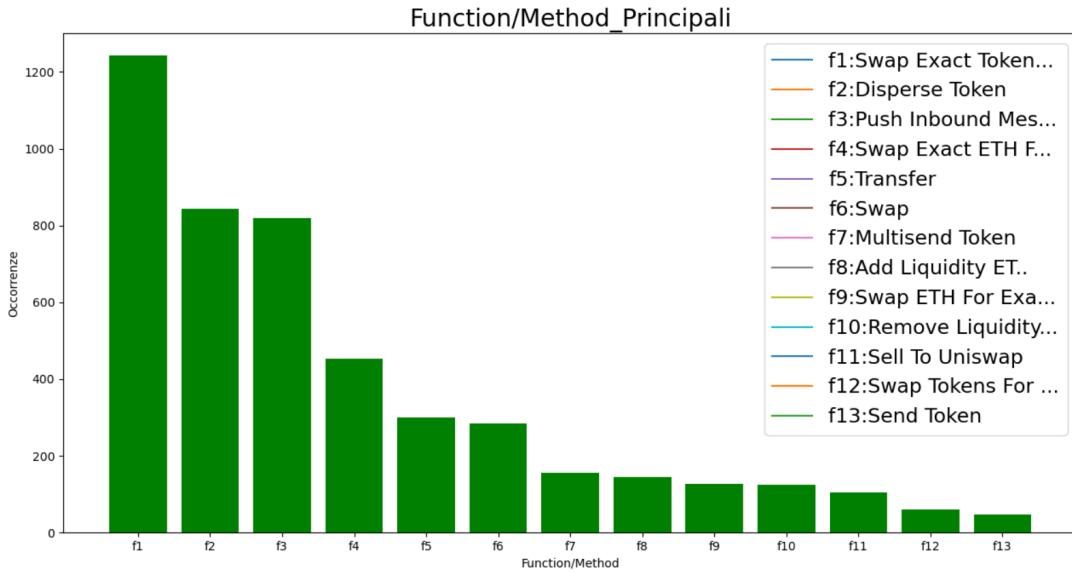


Figura 5.9: Function transazioni

che la transazione in cui avvengono più scambi di token YUP è la **0x966e1d6125d4a555615804c874b0a20874a8b38ddc84d3574a5d5444a5188f8a**, il cui numero di tokens YUP scambiati è pari a 423,586.15. Questa transazione è avvenuta il giorno 8 ottobre 2021, viene eseguita dall’utente rappresentato dall’indirizzo **0xd65c76f572088487bdac4d8fc7573f42c1886ed1** ed è destinata al contratto **Uniswap V2: YUP**. La funzione eseguita in questa transazione è la **Add Liquidity ETH**, ovvero l’aggiunta di liquidità che indica l’attitudine di un investimento a trasformarsi in denaro rapidamente e possibilmente senza perdite. L’aggiunta di liquidità richiede il deposito di un valore equivalente di token ETH ed ERC20 nel contratto di scambio associato al token ERC20, e il primo fornitore di liquidità ad aderire a un pool fissa il tasso di cambio iniziale depositando quello che ritiene essere un valore equivalente di token ETH ed ERC20. Le altre due transazioni, con un alto valore di token YUP scambiati, sono:

0x412e973d64841a8cee872a59ca57dc0218937614c09878ba1a2489a729df2484 e **0xa3b93b7010da49877e60f50ec8eb4ac647724ac8598712f8f3014873a5607a8a**. La prima è una transazione eseguita il 20 dicembre 2021 dal protocollo **Uniswap V2: YUP**, a cui è legato l’account dell’utente mittente, ed è diretta al router **Uniswap V2: Router 2**. La funzione effettuata con l’esecuzione di questa transazione è la **Remove Liquidity ETH With Permit**, e all’interno di essa vengono scambiati 165,311.392996047365934841 tokens YUP. La seconda

è eseguita sempre il 20 dicembre 2021 dall'account legato all'indirizzo **0x972eebcb30252640c60c7e4821b15118f03b267e**, ed è indirizzata al protocollo **Uniswap V2: YUP**. La mansione di questa transazione è l'aggiunta di liquidità, ovvero **Add Liquidity ETH**, come nella prima transazione descritta. All'interno di essa vengono barattati 165,311.392996047365936497 token YUP.

Da questo studio si può dedurre che lo scambio maggiore di token YUP avviene principalmente nelle transazioni che si occupano della aggiunta e della rimozione di liquidità. In particolare vengono aggiunti 588,897.542996 token YUP al liquidity pool di Uniswap V2: YUP, e 165,311.392996047365934841 token YUP corrispondono alla quantità di liquidità che viene prelevata dal pool indicato da Uniswap V2: Router 2. Questi risultati dimostrano che i token YUP vengono utilizzati soprattutto dai liquidity provider, ovvero utenti che scelgono di depositare le proprie monete per facilitare le negoziazioni da parte di altre persone, al fine di guadagnare premi sotto forma di commissioni di negoziazione. Inoltre molti utenti interagiscono con gli exchanger decentralizzati, come Uniswap, principalmente per effettuare investimenti attraverso l'utilizzo dei liquidity pool messi a disposizione da questi ultimi e riprendersi in un secondo momento le proprie somme depositate in precedenza più eventuali commissioni fruttate. Questo ampio trasferimento di token YUP eseguito per fornire liquidità nella pool YUP-ETH di Uniswap, è reso possibile grazie allo YUP bridge, descritto nel Capitolo 3, che rende possibile il trasferimento di token YUP tra le due blockchain. Infatti per ottenere token YUP su Ethereum è necessario acquistare una versione ERC20 del token da un exchange di Ethereum (Uniswap), o bridge token acquisiti su EOS utilizzando il bridge tra EOS e Ethereum.

5.6 Token Transferred

Le analisi precedenti ci hanno condotto a risultati che hanno messo in evidenza un nuovo utilizzo di YUP, rivelando il motivo per cui gli utenti effettuano scambi di token YUP, ovvero l'investimento di questi ultimi. Attraverso i dati prelevati e riposti nel dataset D2, vogliamo ora comprendere se oltre i token YUP, sono presenti ulteriori token con cui gli utenti di Yup si interfacciano attraverso Ethereum per eseguire trasferimenti. Per effettuare questa analisi ci siamo soffermati sullo studio del campo **Tokens Transferred** delle transazioni. Come

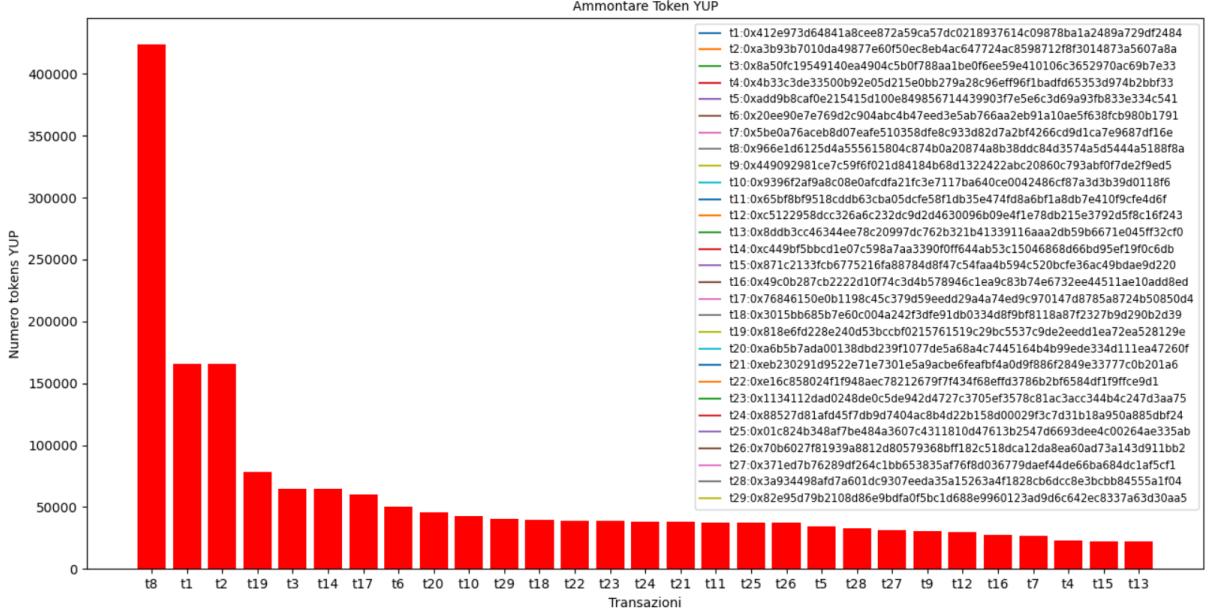


Figura 5.10: Token YUP scambiati transazioni

spiegato in Sezione 2.3 questo campo rappresenta la lista dei trasferimenti di token all'interno di ogni singola transazione. Tramite quest'ultimo, e attraverso i 3 campi contenuti al suo interno, possiamo comprendere da chi, o verso chi, vengono effettuati i trasferimenti e quali token vengono utilizzati. La ricerca viene effettuata sia per le transazioni principali, sia per le transazioni secondarie, descritte nel Capitolo 4.

5.6.1 Campo From

La nostra indagine è iniziata con l'analisi del campo From dei Tokens Transferred. Tramite questo campo abbiamo cercato di capire attraverso quali protocolli gli utenti hanno effettuato i vari trasferimenti di token. I risultati ottenuti sono riportati graficamente attraverso le Figure 5.11 e 5.12 le quali evidenziano che sia per le transazioni principali, sia per le secondarie, il protocollo maggiormente utilizzato per eseguire i trasferimenti è **Uniswap V2**. **Uniswap V2** è l'implementazione di Uniswap, spiegata nel Capitolo 2, che consente la creazione di liquidity pool per coppie di token ERC20/ERC20, invece di supportare solo coppie ERC20/ETH. Proseguendo in questo studio, abbiamo notato che alcuni utenti iniziano ad usufruire della versione V3 di Uniswap, descritta nel Capitolo 2. **Uniswap V3** è differente dalle versioni precedenti. Le

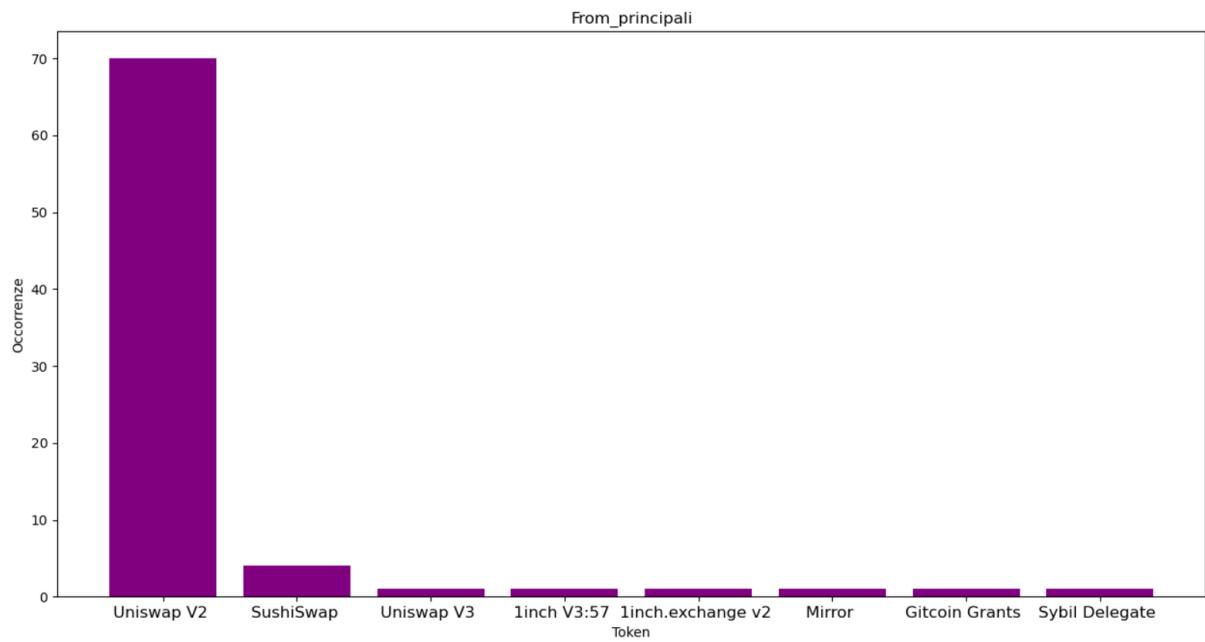


Figura 5.11: Token Transferred From principali

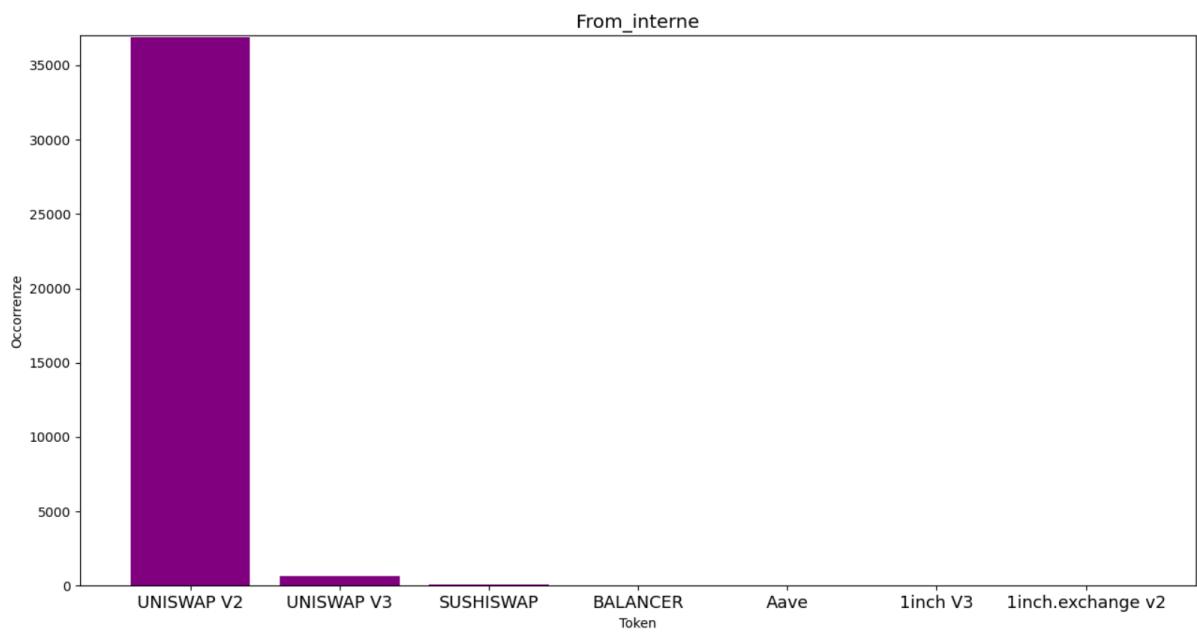


Figura 5.12: Token Transferred From interne

due differenze principali sono la liquidità concentrata, ovvero Uniswap v3 offre la possibilità ai LP di decidere un range nel quale offrire liquidità e l'efficienza del capitale dovuta ai liquidity provider che guadagnano una percentuale uguale fin quando il pool si muove nel margine da loro indicato. Un altro protocollo molto frequente, soprattutto nelle transazioni principali, è **Sushiswap**. Questo cerca di incentivare una rete di utenti a gestire una piattaforma in cui questi ultimi possono acquistare e vendere risorse crittografiche. La funzione di Sushiswap è rispecchiare uno scambio tradizionale facilitando l'acquisto e la vendita delle risorse tra utenti. Successivamente i nostri risultati mostrano che è ricorrente **1 inch v3**. Questo è un aggregatore DEX che seleziona i prezzi di criptovaluta più economici su tutti gli scambi decentralizzati, risparmiando più tempo poiché non è più necessario controllare manualmente ogni scambio per vedere i migliori tassi di cambio. Offre l'ottimizzazione dell'utilizzo del gas e commissioni di transazione inferiori. Oltre questa caratteristica la versione v3 mette a disposizione più liquidità, ovvero il mercato è più liquido e quindi si prevede che ci siano più compratori disposti ad acquistare i token di un utente al prezzo stabilito da quest'ultimo. Proseguendo si è riscontrato il protocollo **1 inch.exchange v2**. Quest'ultimo include un vantaggio per gli utenti, ovvero, l'introduzione dell'API Pathfinder. Questa nuova API contiene un nuovo algoritmo di scoperta e routing che aiuterà gli utenti a trovare i migliori percorsi possibili per uno scambio di token nel più breve tempo possibile. Estenendo il nostro studio, abbiamo osservato altri protocolli utilizzati per i trasferimenti di token. Abbiamo il **Mirror Token (MIR)** che è il token di governance del Mirror Protocol. Questo è un protocollo di finanza decentralizzata (DeFi) basato sulla creazione di asset sintetici, chiamati mAssets (mirrored assets). Questi strumenti permettono di esporsi al prezzo del corrispettivo asset reale attraverso un sistema peer-to-peer. Presente nei nostri trasferimenti è anche il **Gitcoin grants**, che ha una caratteristica principale ovvero il suo meccanismo di finanziamento democratico proprietario, chiamato finanziamento quadratico. La logica alla base del finanziamento quadratico è che un pubblico più ampio utilizza i beni pubblici, e il loro sviluppo dovrebbe quindi ricevere un sostegno più popolare invece di dipendere da poche grandi donazioni. Il protocollo **Sybil delegate**, invece, è uno strumento di governance. Lo scopo di Sybil è fornire ai titolari di token di governance un modo semplice per identificare i rappresentanti della propria comunità e aumentare il coinvolgimento nel processo di sviluppo del protocollo. Il protocollo **Balancer** è un software in

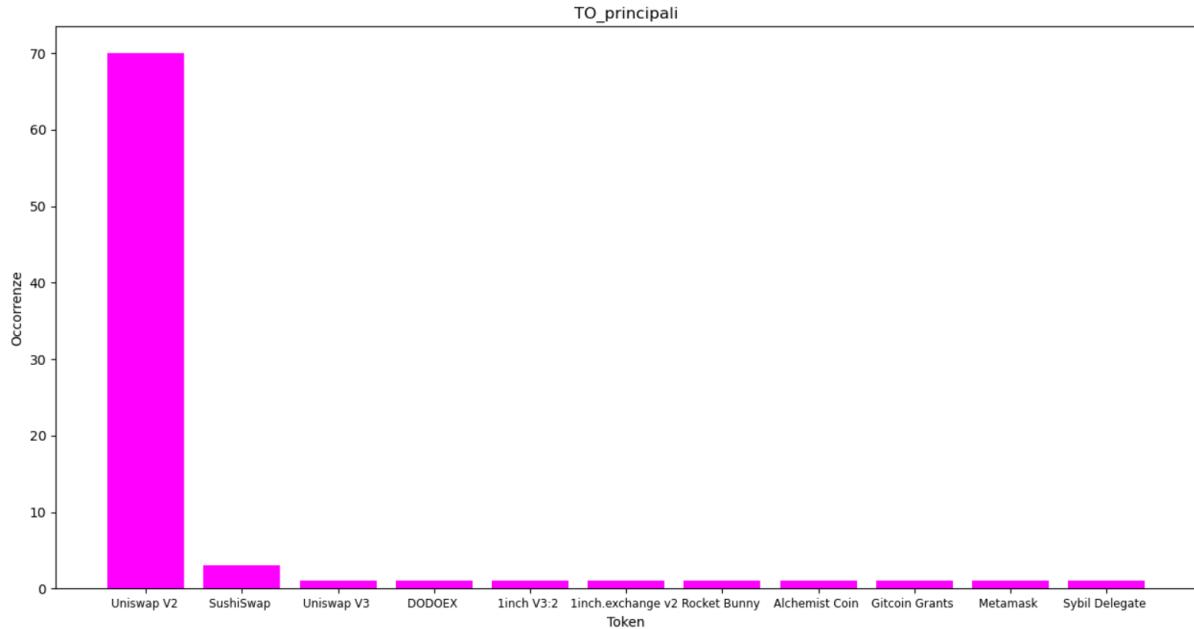


Figura 5.13: Token Transferred TO principali

esecuzione su Ethereum che cerca di incentivare una rete distribuita di computer a gestire uno scambio in cui gli utenti possono acquistare e vendere qualsiasi criptovaluta. Infine, sempre solo nelle transazioni secondarie, è presente per 33 volte il protocollo **Aave**. Quest'ultimo è un protocollo Open Source e non-custodial per guadagnare interessi su depositi e attività di prestito.

5.6.2 Campo To

Permanendo con lo studio dei Tokens Trasferred, abbiamo spostato la nostra attenzione sul campo To. Dopo aver compreso da che protocolli vengono effettuati i trasferimenti, questo nuovo campo ci consente di comprendere verso chi e verso quali protocolli vengono effettuati i trasferimenti di token. La stessa indagine viene nuovamente condotta sia sulle transazioni principali, sia sulle transazioni secondarie, e i risultati vengono raffigurati nelle figure 5.13 e 5.14. Nelle figure 5.13 e 5.14 viene mostrata quindi, una statistica dei protocolli maggiormente presenti come destinatari dei trasferimenti. L'esito ha dimostrato che **Uniswap V2** continua ad essere il più ricorrente anche in questa indagine. Oltre quest'ultimo sono presenti anche nuovi protocolli, mai riscontrati in precedenza. In particolare abbiamo **DODOEX** o DODO, che è un

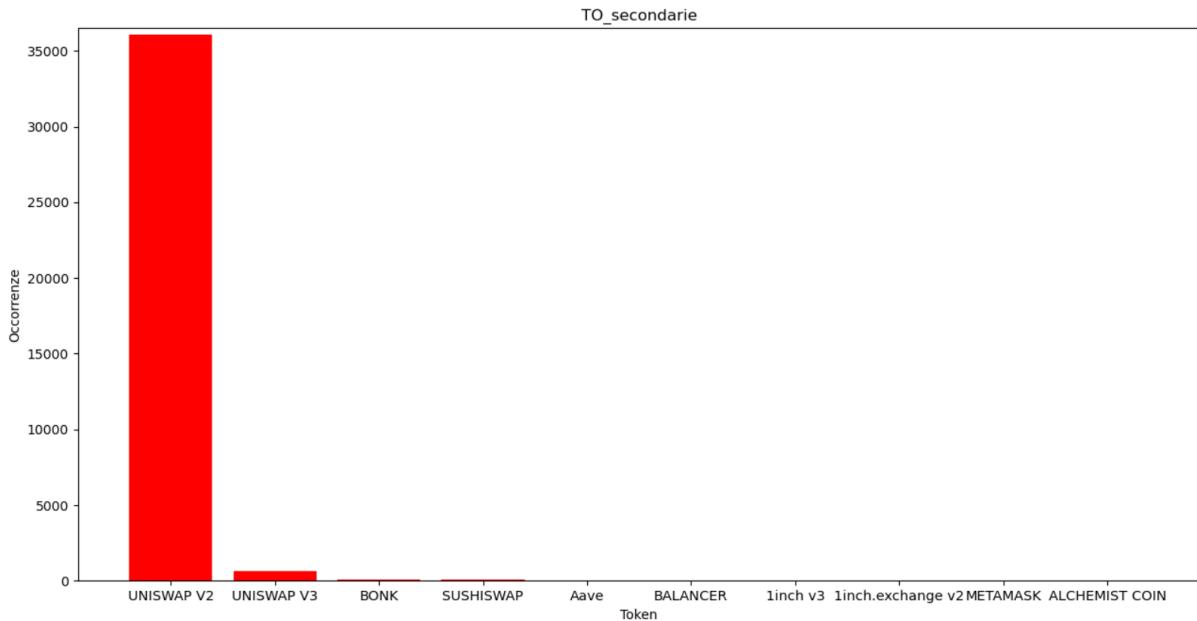


Figura 5.14: Token Transferred TO interne

protocollo DeFi e un fornitore di liquidità on-chain basato su Ethereum. Caratterizzato da un nuovo algoritmo Proactive Market Maker (PMM), il protocollo è governato da un token DODO basato su ERC20. DODO offre pool di liquidità, altamente efficienti in termini di capitale, che riducono le perdite temporanee e lo slippage per i trader. Proseguendo è risultato frequente anche **Rocket Bunny**, che fornisce aggiunte automatiche di liquidità, rendimento composto, offerta deflazionistica, premi di liquidità e protezione dagli shock dei prezzi. In seguito abbiamo riscontrato **Alchemist coin**. Alchemist è una piattaforma unica che offre agli utenti l'accesso a una varietà di piattaforme per guadagnare premi e partecipare ad aste di lancio equo di token. Un'ulteriore scoperta è **Metamask**, ovvero uno strumento che funge da "ponte" tra il proprio browser, la piattaforma Ethereum e le Dapps costruite su tale piattaforma. MetaMask permette agli utenti di utilizzare le Dapps Ethereum direttamente dal loro browser. MetaMask consente ai propri utenti di conservare, scambiare e ricevere token ETH, e più in generale tutti i token basati sulla blockchain Ethereum, cioè token ERC20. Infine c'è anche **Bonk**. Bonk è un token ERC20 decentralizzato creato sulla rete Ethereum ed è un token di utilità che può essere utilizzato per creare NFT. Invece, per quanto riguarda i protocolli già osservati in precedenza tramite lo studio del campo From, ricorrono all'interno del campo To anche **Uniswap V3**, **1inch V3**, **Gitcoin Grants**, **Sybil Delegate**, **Sushiswap** e **1inch.exchange v2**.

Conclusioni Campo From e To

Concludendo questa prima analisi dei Token Trasferred, i risultati mettono in luce che in tutte le nostre transazioni vengono utilizzati i decentralised exchangers e piattaforme di DE-FI. In particolare Uniswap V2 viene riconfermata come la preferita dagli utenti per eseguire le loro operazioni, probabilmente perché quest'ultima mette a disposizione anche pool di coppie di token ERC20/ERC20. Infatti come abbiamo notato nelle nostre ricerche, gli utenti sono soliti effettuare scambi tra diversi token ERC20; e quindi la creazione di queste coppie permette loro facili scambi tra un token ERC20 di input e un token ERC20 di output; al contrario di altri exchanger che impongono restrizioni per le coppie di scambio. Inoltre questo nuovo studio introduce anche una nuova versione di Uniswap, ovvero la V3 che offre maggiori vantaggi per l'aggiunta di liquidità. Nonostante ciò, la versione V2 risulta comunque essere più utilizzata della V3, anche se quest'ultima introduce miglioramenti; ma questo risultato probabilmente è condizionato dal contenuto del nostro dataset D2. Infatti quest'ultimo contiene dati di transazioni e trasferimenti effettuati tra i giorni 11 ottobre 2020 e 5 agosto 2021. Invece, Uniswap V3 è stata annunciata il 23 marzo 2021, e rilasciata su Ethereum il 6 maggio 2021. Infine oltre Uniswap, molti utenti, magari più abili, si interfacciano con nuovi exchanger decentralizzati, che possono fornire ulteriori vantaggi rispetto Uniswap come: un guadagno maggiore di interessi, un mercato più liquido e uno scambi in breve tempo.

5.6.3 Campo For

Gli studi del campo "Tokens Transferred" delle transazioni, vengono terminati analizzando i token "intermedi" di ogni trasferimento. Tramite questa analisi cercheremo di comprendere se sono presenti ulteriori token, oltre i token YUP, che vengono trasferiti nelle transazioni dei nostri utenti.

La statistica dei risultati è riportata graficamente nelle figure 5.16 e 5.15. Tramite queste ultime notiamo che i token maggiormente utilizzati, sia nelle transazioni principali, sia nelle secondarie, sono:

- **UNI-V2** che è un token ERC 20. La chiave di questi token sono gli smart contract, contratti ad esecuzione automatica che permettono sia di creare token, sia di gestire scambi.

- **YUP** è un protocollo di consenso sociale che facilita la misurazione, la cattura e lo scambio di influenza in un'economia basata sull'opinione pseudonima. Identifica i contenuti e distribuisce i premi in base al valore delle opinioni associate a quel contenuto.
- **WETH** è un tipo speciale di token ERC20 che mira a facilitare le operazioni di scambio tra le diverse piattaforme decentralizzate che fanno parte del vasto ecosistema di Ethereum. Questo token funziona come un "convertitore" tra Ether e token.
- **USDT**, ovvero Tether(USDT), è una criptovaluta a moneta stabile che si basa sul dollaro americano (USD), quindi il valore di una moneta Tether corrisponderà al tasso di cambio di un dollaro.
- **USDC** o USD Coin, è una stablecoin ancorata al dollaro USA. Può anche consentire alle aziende di accettare pagamenti in risorse digitali.
- **RARI**, ovvero Rarible è un software che consente a creatori digitali di emettere e vendere risorse crittografiche personalizzate che rappresentano la proprietà del loro lavoro digitale. I token che i creatori generano su Rarible sono conosciuti come token non fungibili (NFT). Ogni NFT è unico e non sono intercambiabili.
- **BASE** consente ai trader di speculare sull'intero settore delle criptovalute con un token. BASE è un asset sintetico progettato per simulare i modelli di mercato del suo asset sottostante: tutte le criptovalute. Ciò consente agli utenti di speculare in modo agnostico su ogni token.
- **SHROOM** è un protocollo DeFi e DAO focalizzato sul conio, lancio e trading di asset in-game. Il token SHROOM ha un valore intrinseco pari a zero, ed è inteso come token di governance. Qualsiasi valore che può maturare per il token è interamente deciso dalla comunità e dai suoi possessori di token.
- **SHIB**, Shiba Inu, è un token progettato per essere un'alternativa compatibile con Ethereum. SHIB è intenzionalmente abbondante, con una fornitura iniziale in circolazione di un quadrilione di monete. L'ecosistema Shiba Inu supporta anche progetti come uno scambio decentralizzato chiamato Shibaswap.

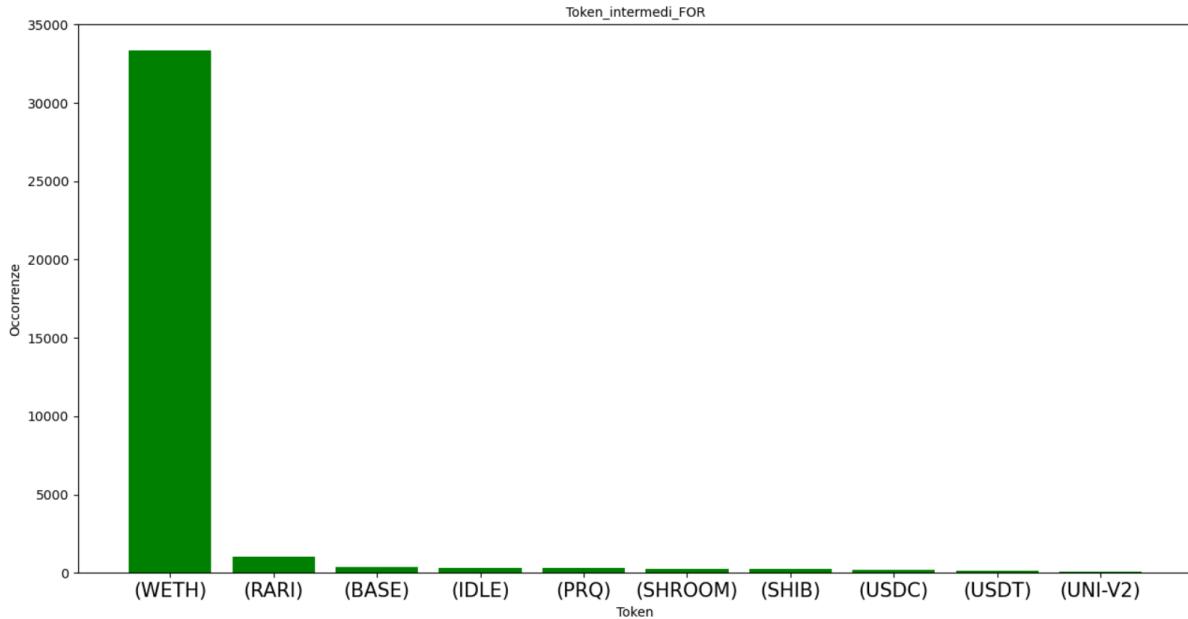


Figura 5.15: Token intermedi FOR interne

- **IDLE** è una piattaforma di rendimento decentralizzata sulla piattaforma Ethereum che utilizza il ribilanciamento automatico tra i protocolli DeFi al fine di ottimizzare il ritorno per i suoi utenti. Gli utenti possono scegliere tra rischio elevato, rendimento elevato (rendimento massimo) o adeguarsi a un rischio inferiore utilizzando la strategia di allocazione RiskAdjusted.
- **PRQ,PARSIQ**, si descrive come una piattaforma di nuova generazione per il monitoraggio e l'intelligenza, che offre strumenti di analisi per la tecnologia blockchain in una miriade di settori. Ciò consente agli utenti di tenere traccia dell'attività di rete in tempo reale, sbloccare nuovi casi d'uso per la loro applicazione e creare notifiche istantanee.

In conclusione attraverso questo studio dei token, abbiamo scoperto che gli utenti di Yup non utilizzano solo i token YUP per effettuare trasferimenti, ma in realtà in molte transazioni prevalgono token differenti. In particolare risalta il token WETH. Questo ultimo compare in 1,333 transazioni principali e in 33,374 transazioni secondarie, e probabilmente è il token più ricorrente poiché permette di effettuare scambi di qualsiasi token. E' addirittura più presente del token YUP stesso. Questo ci fa comprendere che molti utenti di Yup sono interessati a scambi e trasferimenti di token di qualsiasi genere, ed evidentemente sono persone esperte

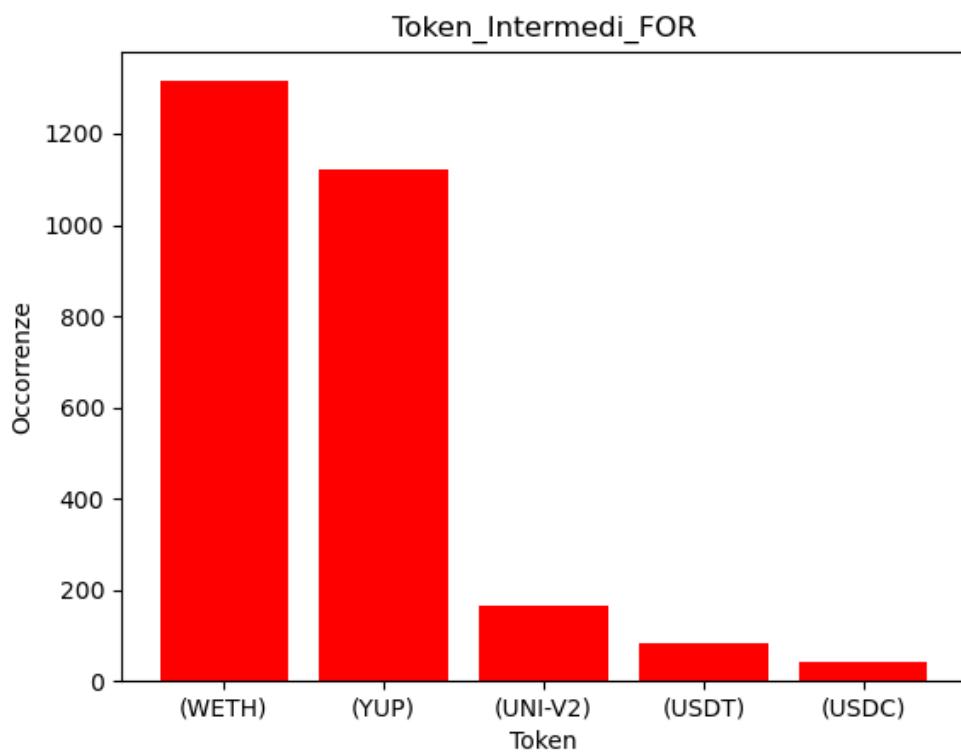


Figura 5.16: Token intermedi FOR principali

di trading e informate sull’andamento dei crypto token tanto da individuare quali sono i migliori su cui puntare. Infine risulta anche comprensibile la scelta di questi utenti di utilizzare piattaforme decentralizzate come Uniswap V2 e V3. Infatti queste ultime sono, per il momento, le uniche che supportano lo scambio di token WETH, che come abbiamo visto è il token maggiormente barattato fra gli utenti.

5.7 Function delle transazioni

Successivamente, abbiamo lavorato sulle funzioni delle transazioni presenti nel nuovo dataset D2. Questo studio è stato eseguito attraverso il campo ”Input Data” delle transazioni. Come spiegato in Sezione 2.3, questo campo contiene dati aggiuntivi per le transazione prese in considerazione, e ci aiuta a comprendere quali sono i metodi maggiormente utilizzati per effettuare le transazioni. Questa indagine ha dimostrato che oltre le funzioni precedentemente riscontrate tramite il dataset D1, ne prevalgono nuove. In particolare la funzione più utilizzata per le transazioni principali, è nuovamente la **swapExactTokensForETH**, ovvero la funzione usata da Uniswap per scambiare token ERC20 in ETH. Per le transazioni secondarie, invece si ha **buyAndFree22457070633**. Questa è una funzione molto particolare che viene fornita dal token LiquidGasToken (LGT). LGT ha come scopo quello di rappresentare il gas di Ethereum sotto forma di un token ERC20 che possa essere scambiato tramite un liquidity pool integrato. Lo scopo principale del token è quello di rendere il gas rimborsabile nel caso in cui una transazione ne richieda meno del previsto, per ridurre i costi di esecuzione su Ethereum. Gli autori del token hanno anche introdotto alcune ottimizzazioni (ad esempio, il numero 22457070633 in coda al nome dell’operazione), per ottimizzare il costo in gas dell’esecuzione delle varie transazioni [17]. Infatti gli utenti devono pagare una commissione, ovvero il gas, nel momento in cui richiedono l’esecuzione di una transazione. Questa ”regola” imposta dalle piattaforme tipo Ethereum, potrebbe essere un fattore limitante nell’incremento dell’utilizzo delle dApp. Invece tramite questa funzione e il token LGT, individuati durante le nostre analisi, l’utente potrebbe pagare meno commissioni quando effettua una transazione, e in questo modo più persone potrebbero essere incentivate ad utilizzare queste piattaforme decentralizzate.

Le altre funzioni che ricorrono maggiormente sono per quanto riguarda le transazioni secondarie:

- **swapExactETHForTokens** che si occupa dello scambio di un numero esatto di ETH con i token.
- **swapExactTokensForTokens** che consiste nello scambiare un numero esatto di token con altri token.
- **swapETHForExactTokens** che permette lo scambio di ETH per un numero esatto di token.
- **swapExactTokensForETH** che effettua lo scambio di un numero esatto di token per degli ETH.
- **swapExactTokensForETHSupportingFeeOnTransferTokens** si occupa di effettuare uno scambio di un numero esatto di token con una quantità di ETH, e in più viene aggiunta una commissione di supporto sui token trasferiti.
- **swapTokensForExactETH** che opera scambiando token con un numero esatto di ETH.
- **approve** è la funzione di approvazione utilizzata per approvare il prelievo di importi dagli indirizzi.
- **multicall** raggruppa diverse chiamate HTTP in una sola. In questo modo, se vogliamo ottenere dati da n diverse richieste, possiamo raggrupparle prima di inviare e inviare una sola richiesta HTTP, migliorando i tempi di risposta e il consumo delle nostre chiamate.
- **atomicMatch** è un metodo che consente gli scambi atomici, ovvero contratti di scambio automatici che consentono a due parti di scambiare token da due blockchain diverse.
- **transfer** usata per trasferire una quantità di token da un indirizzo.
- **depositFor** è il metodo utilizzato per effettuare il deposito di una somma di denaro in modo da poter effettuare successivamente i vari scambi Ether.

- **cancelOrder** che compare come operazione nelle transazioni che si occupa di cancellare una transazione in sospeso.

Per le transazioni principali invece abbiamo:

- **removeLiquidityETHWithPermit** consente la rimozione della liquidità, ma l'utente prima firma l'autorizzazione al contratto del router per spendere i propri token.
- **removeLiquidityETH** che si occupa di rimuovere liquidità da un pool in cui precedentemente era stata aggiunta.
- **addLiquidityETH** che viene utilizzata per aggiungere liquidità ad un pool.
- **swapExactTokensForETH** consente lo scambio di un numero esatto di token con gli ETH.
- **swapExactTokensForTokens** effettua lo scambio di un numero esatto di token con altri token.
- **swapETHForExactTokens** esegue lo swap di una somma di ETH con un numero esatto di token.
- **swapExactETHForTokens** che consiste nello scambiare un numero esatto di ETH con dei token.
- **swapTokensForExactTokens** effettua lo scambio di una certa quantità di token con un'altra quantità di token precisa.
- **sellToUniswap** permette agli utenti di vendere i propri token.
- **transfer** si occupa del trasferimento di token tra diversi indirizzi.

Questo studio ci riconduce agli esiti ottenuti in precedenza con il dataset D1, infatti le operazioni principalmente svolte dagli utenti risultano essere ancora una volta funzioni di scambi di token. In particolare, nel caso di questi nuovi dati presenti nel dataset D2, la funzione di swap maggiormente riscontrata è la **swapExactTokensForETH**. Attraverso questo metodo

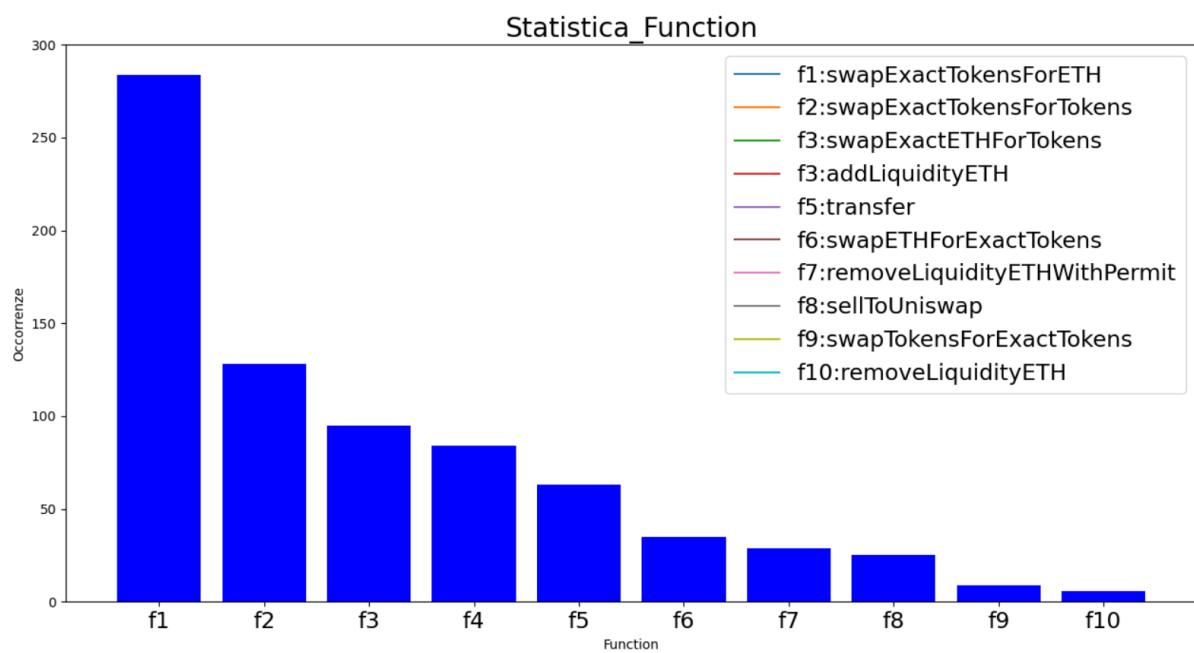


Figura 5.17: Function principali

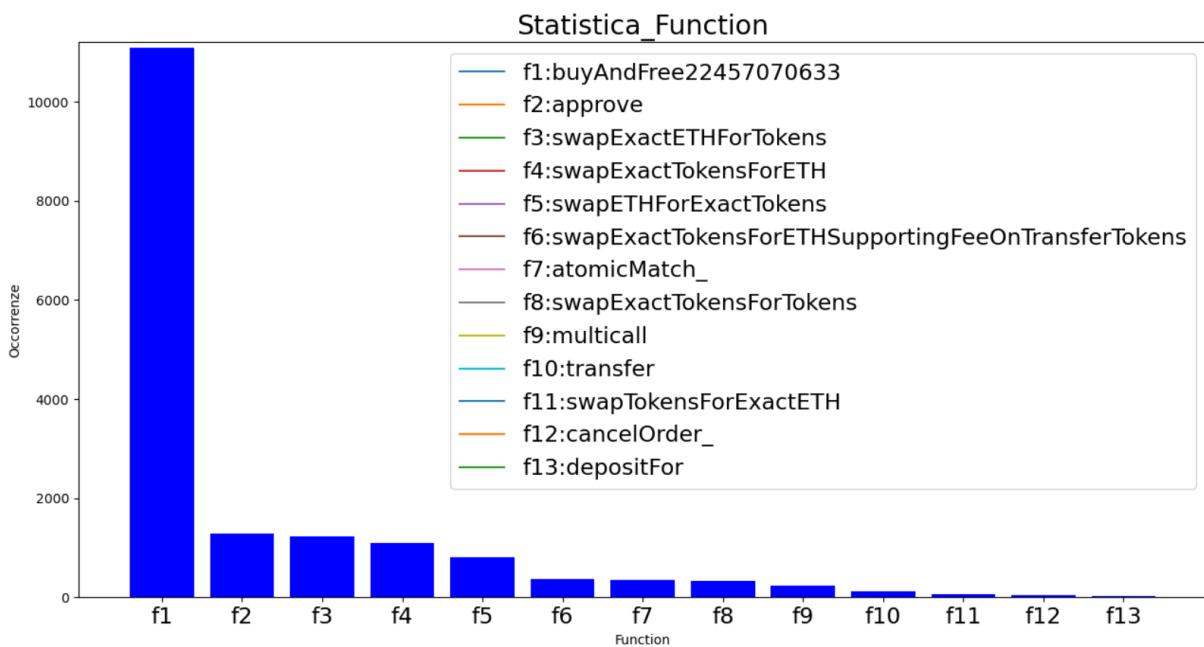


Figura 5.18: Function interne

gli utenti barattano un determinato numero di token in cambio di ETH. In questo caso gli utenti preferiscono stabilire loro stessi la quantità di token da trasferire. Inoltre le altre funzioni presenti risultano essere nuovamente funzioni di swap e di transfer, oppure funzioni utilizzate per investire denaro in liquidity pool. Tutte queste funzioni continuano a corrispondere alle funzioni standard dei token ERC20. Infatti sia i token YUP, sia i token WETH maggiormente ricorrenti sono token di tipo ERC20. Infine oltre queste funzioni prevalgono, anche se con frequenza minore, nuovi metodi utilizzati anche con differenti token presenti nelle nostre transazioni, come il token RARI, USDT e così via. Queste funzioni e questi token compaiono tra i nostri dati poiché gli utenti esplorano anche con differenti tipi di exchanger.

5.8 Address FROM e TO delle transazioni

Infine abbiamo condotto un'ultima analisi delle transazioni presenti nel secondo dataset. Questa indagine è stata eseguita analizzando i campi "From" e "To" delle transazioni. Come descritto in Sezione 2.3 il primo campo identifica l'indirizzo che richiede l'esecuzione della transazione, il secondo campo rappresenta il destinatario della transazione. Per le transazioni principali l'indirizzo che effettua il numero più elevato di transazioni è

0x44b8fce3beba8eec1fa3542d2490e67c022d477, ovvero l'indirizzo dell'account che ha creato lo smart contract di Yup. Mentre l'indirizzo che riceve il numero maggiore di transazioni corrisponde al contratto **Uniswap V2: Router 02**⁸, ovvero il router per Uniswap V2. Per quanto riguarda invece le transazioni secondarie ricorre maggiormente come indirizzo mittente

0x86254cb5a96c161e503d3255d67fdddec056fefef. Quest'ultimo effettua principalmente operazioni dirette verso un destinatario rappresentato dal contratto **0xC65433d2a598c323e04fD143566b08609Af008DC**. Invece il destinatario ricorrente più volte corrisponde all'utente rappresentato dal contratto

Contract 0xc6e6dd6a0c61651d1bc055dc1c22418a729d41bb. I mittenti di queste transazioni risultano essere account di proprietà esterna (descritti nel Capitolo 2), ovvero controllati dagli utenti, mentre i destinatari sono account contratti (presenti nel Capitolo 2) non controllati

⁸Contract 0x7a250d5630b4cf539739df2c5dacb4c659f2488d

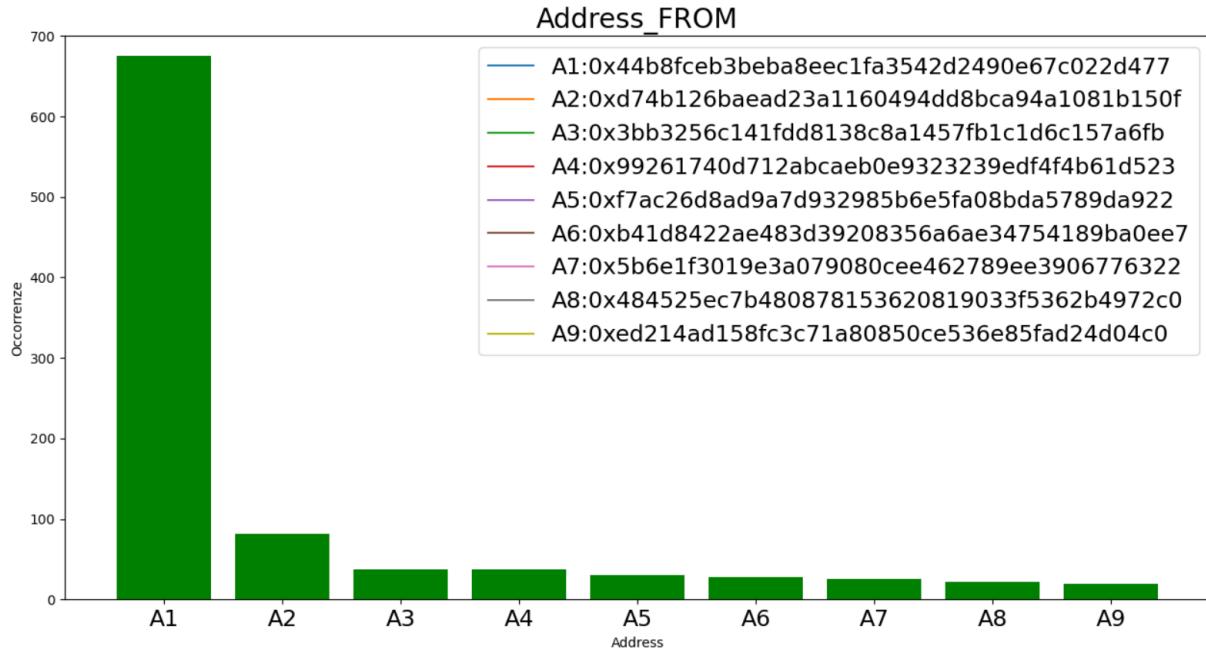


Figura 5.19: Address From principali

da utenti. Questi mittenti effettuano quindi una funzione definita sul contratto con cui interagiscono. Il destinatario che riceve il numero maggiore di transazioni è **Uniswap V2: Router 02**. Le analisi effettuate precedentemente hanno rilevato che il contratto di Uniswap V2 mette a disposizione le funzione di scambio e trasferimenti di token, e di aggiunta e rimozione di liquidità. Quindi il mittente che interagisce con questo account contratto è un utente che utilizza la piattaforma per eseguire operazioni di trading. Per quanto riguarda il mittente ricorrente nelle transazioni secondarie, effettua le funzioni messe a disposizione dal contratto con cui interagisce, ovvero il contratto **0xC65433d2a598c323e04fD143566b08609Af008DC**. Queste funzioni sono tutte funzioni di scambio, in particolare o di scambio di ETH per una quantità di un determinato token, spesso differente dai precedenti, oppure uno scambio di token per ETH. Quindi possiamo concludere che questo secondo mittente è un utente che sembra essere interessato in particolar modo all’andamento dei crypto token.

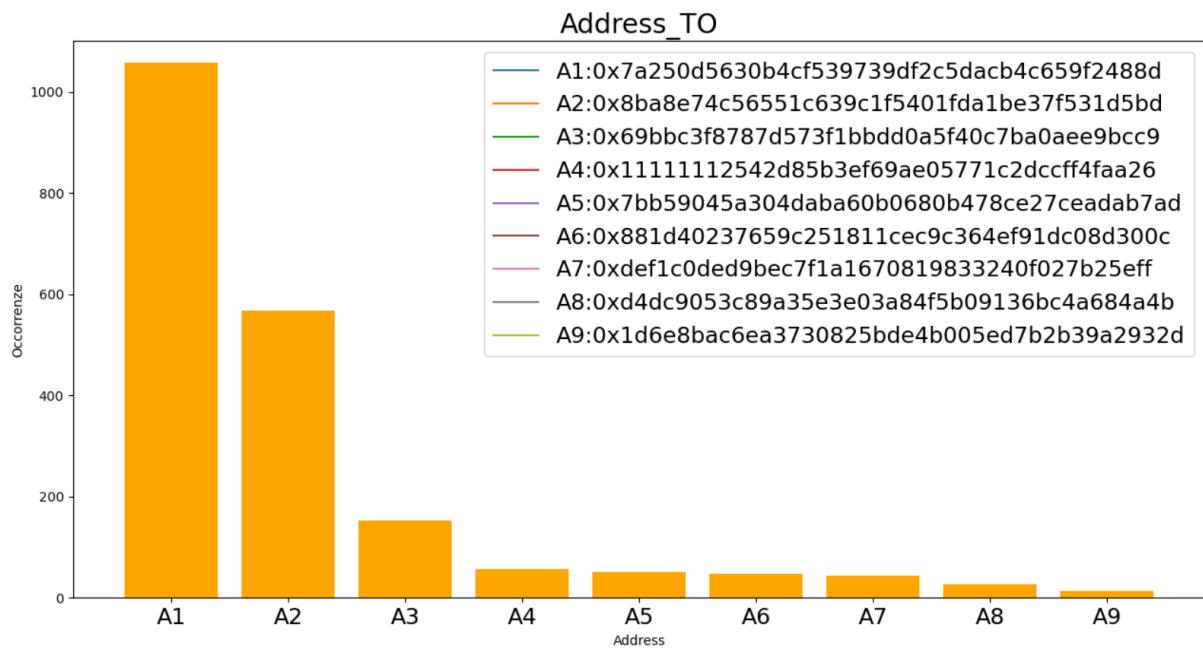


Figura 5.20: Address TO principali

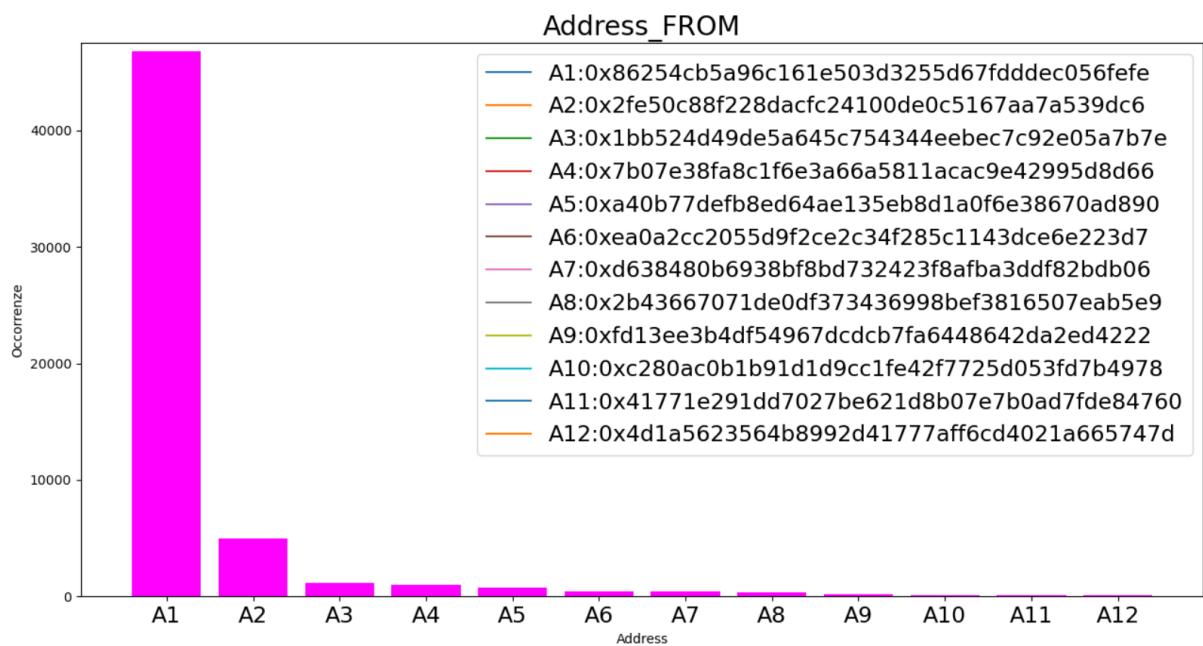


Figura 5.21: Address From interne

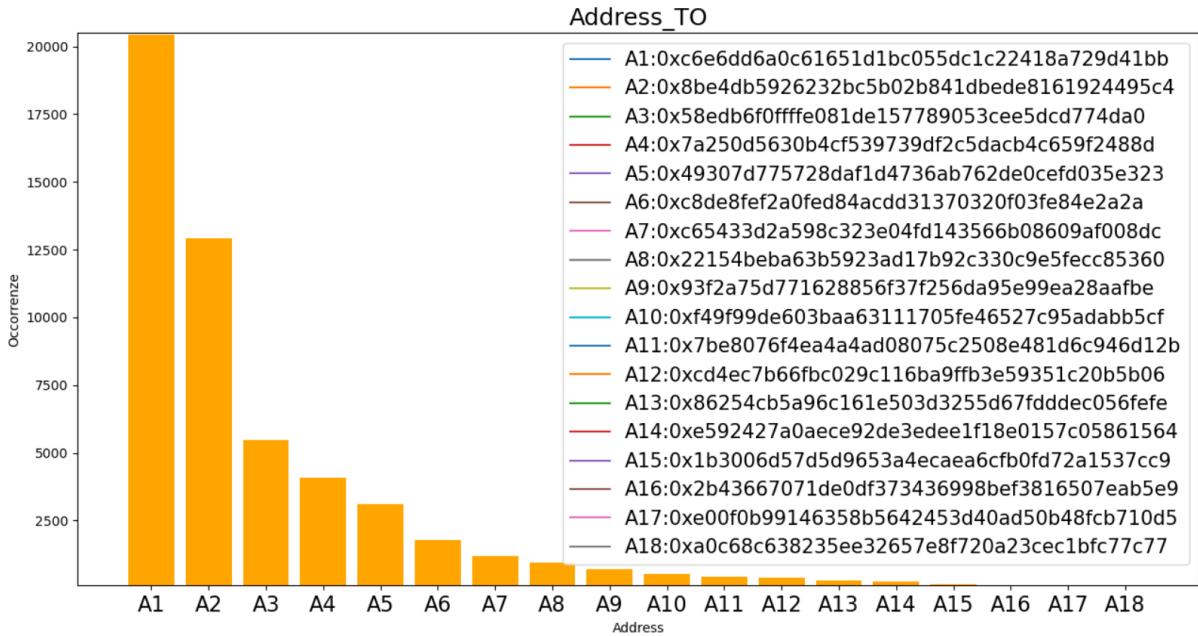


Figura 5.22: Address To interne

5.9 Tipi delle transazioni

Lo studio dei tipi di transazione è basato sulla ricerca della tipologia, riportata nel campo **Txn Type** descritto nella Sezione 2.3, di circa 60,000 transazioni. Il risultato di questa analisi ha evidenziato che in quasi tutte le 57,457 transazioni analizzate il campo "type" non viene specificato e che chiameremo "NO Type", mentre nelle restanti i tipi utilizzati sono **0 (Legacy)** in 207 transazioni e **2 (EIP - 1559)** in 212 transazioni. Questi due valori sono molto simili tra di loro, e le transazioni EIP-1559 corrispondono a quelle eseguite in agosto. Infatti questa tipologia viene introdotta da agosto 2021, mentre prima veniva utilizzata solo la tipologia standard. Di seguito possiamo affermare che le 212 transazioni con tipo 2 (EIP-1559), delineato nel Capitolo 2, sono tutte transazioni eseguite in agosto 2021, subito dopo l'uscita di questa nuova tipologia e ad agosto sono maggiormente eseguite dagli utenti rispetto alle 0 (Legacy). Probabilmente questo incremento delle transazioni 2 (EIP-1559) è dovuto anche al vantaggio che gli utenti ricavano riguardo il pagamento delle tasse ai miner, come spiegato nel Capitolo 2. Mentre per le restanti transazioni, sappiamo con certezza che 207 sono di tipo 0 (Legacy), descritto nel Capitolo 2. Ma considerando che le 2 (EIP-1559) sono state rilasciate il 4 agosto 2021, possiamo affermare che le restanti transazioni effettuate prima della data indicata, e segnate come "NO

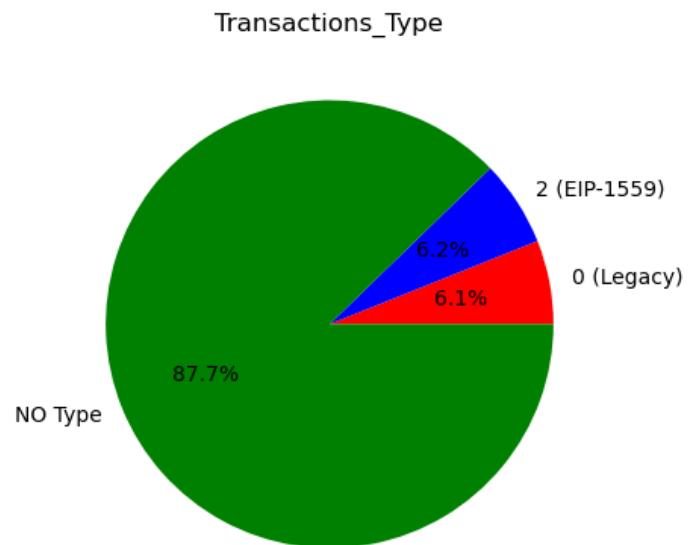


Figura 5.23: Transactions Type

Type”, sono considerabili come transazioni di tipo 0 (Legacy). Infatti, il formato delle transazioni ”NO Type” e di quelle che con certezza sappiamo essere 0 (Legacy), è identico. Questo ci ha portato a concludere che queste ultime, avendo lo stesso formato e di conseguenza le stesse caratteristiche e lo stesso modo di operare, sono tutte transazioni 0 (Legacy) e di conseguenza gli utenti che hanno effettuato operazioni tra dicembre 2020 e dicembre 2021 hanno eseguito soprattutto transazioni di tipi 0 (Legacy).

Capitolo 6

Conclusioni e Lavoro Futuri

I social media ricoprono ormai un ruolo fondamentale nella vita di tutti i giorni. Negli anni, hanno rivoluzionato il modo di comunicare tra le persone. Tuttavia, il loro ampio utilizzo ha portato a diversi tipi di problemi, legati principalmente alla privacy degli utenti e alla libertà di pensiero. Grazie alle piattaforme distribuite, e principalmente alla tecnologia blockchain, i social media hanno subito una rivoluzione che ha portato ad una nuova generazione di piattaforme in grado di venire incontro agli utenti sotto molteplici punti di vista. In particolare, le cosiddette Blockchain Online Social Media (BOSMs), piattaforme parzialmente o completamente decentralizzate basate su blockchain, hanno introdotto nuove tecniche per la gestione della privacy degli utenti o per una ridistribuzione più equa delle risorse economiche generate nella piattaforma. Parallelamente allo sviluppo di queste blockchain, è in crescita anche il campo dell'economia digitale, infatti molte persone iniziano ad addentrarsi in quest'area per usufruire di operazioni economiche maggiormente vantaggiose e per effettuare investimenti traendo profitti. Un utente, tramite queste piattaforme, è libero di comprare o vendere senza essere mediato da alcun terzo fiduciario, come ad esempio un istituto bancario. Infine sia le attività di compravendita, che gli accordi commerciali passano per la scrittura di un contratto informatico la cui natura è assolutamente non modificabile e condizionata nel tempo alle regole stabilite nel contratto stesso. Questo aspetto non va sottovalutato in quanto va a garantire equità e sicurezza tra le parti in causa, consentendo di implementare nuove opportunità di business in maniera assolutamente creativa.

Grazie alle sue innovative idee e alla sua possibilità di integrarsi con altre piattaforme già

esistenti, una BOSMs ad oggi molto utilizzata è Yup, basata principalmente sulla blockchain EOS. Yup permette agli utenti di guadagnare criptomoneta, chiamata YUP, tramite azioni sociali, ad esempio creazione e curazione di contenuti, ma anche di trasferire i propri token YUP su una blockchain differente da EOS. Infatti, con l'avvento del canale di comunicazione instaurato con il bridge di Yup, gli utenti sono liberi di spostare i propri token YUP sulla blockchain di Ethereum e utilizzare i propri fondi per effettuare operazioni tutt'altro che correlate all'ambito sociale. Probabilmente, molte persone iniziano ad usufruire di questa piattaforma spinti dall'idea di un guadagno maggiore, ricavabile tramite i mercati finanziari presenti sulle blockchain. Difatti queste ultime, permettono agli account di acquistare, scambiare e vendere token differenti dal token YUP, consentendo così agli utenti di interfacciarsi con differenti criptovalute, magari più proficue di YUP.

Attraverso la nostra tesi abbiamo cercato di comprendere quanto l'economia digitale stia prevalendo sulle attività sociali che dovrebbero rappresentare il fulcro di queste piattaforme. Questo tipo di analisi, ci ha permesso di valutare quanto le persone che stanno attualmente utilizzando Yup sono coinvolte nel mondo cripto, e quanto l'economia possa impattare sulle attività sociali svolte su queste piattaforme. Spostandoci quindi sull'ambito "economico" di questa dApp, ci siamo concentrati sul modo in cui gli utenti adoperano i token posseduti rappresentanti i profitti che hanno tratto dall'utilizzo di Yup. In particolare facciamo riferimento a coloro che hanno trasferito questi token su Ethereum.

Per lo svolgimento della tesi, abbiamo iniziato da una fase di download dei dati, effettuata tramite scraping del blockchain explorer Etherscan. I dataset ottenuti sono due, il dataset *D1* e il dataset *D2*. Il dataset *D1* contiene informazioni di trasferimenti di token YUP effettuate dal *16 ottobre 2020 al 26 dicembre 2021*, per un totale di 5,247 trasferimenti. Il dataset *D2* contiene invece i trasferimenti di token ERC20 di tutti gli utenti individuati nel dataset *D1*, nel periodo dal *11 ottobre 2020 al 5 agosto 2021* e contiene in totale 59,591 transazioni. Le nostre analisi hanno evidenziato che le attività svolte frequentemente dagli utenti e le scelte intraprese per lo svolgimento di queste, vengono eseguite soprattutto per scopi economici. In particolare, queste indagini hanno mostrato che gli utenti trasferiscono le loro ricompense su Ethereum per effettuare operazioni di trading e staking online. Abbiamo inoltre scoperto che il DEX preferito da questi utenti è Uniswap V2, ma in realtà alcuni account decidono di interfacciarsi con nuovi

exchangers decentralizzati per ottenere differenti vantaggi. Di conseguenza abbiamo notato che molti utenti sembrano essere informati sull’andamento delle criptovalute, tanto da investire ed effettuare scambi con token diversi dal token YUP, ma comunque affini allo standard ERC20. Durante queste analisi è stata rilevata anche la problematica MEV, presente sulla blockchain di Ethereum da ormai oltre un anno. Nello specifico si è mostrato che alcune transazioni YUP analizzate e presenti su questa blockchain, sono soggette a questa problematica tramite attacchi di tipo *sandwich*. Questa problematica rilevata segnala quanto l’architettura alla base della DeFi è ancora lontana da essere completamente esente da rischi.

6.1 Lavori futuri

Questo lavoro di tesi ha evidenziato diversi aspetti interessanti che potranno essere approfonditi in futuro. Di seguito elenchiamo alcuni possibili lavori futuri da poter condurre:

- **Analisi sull’identità degli utenti:** Le blockchain come Ethereum assicurano la pseudonimizzazione, ovvero i dati personali non vengono attribuiti direttamente ad una persona, ma ad uno pseudonimo (ovvero la sua chiave pubblica). Ma dApp come Yup permettono ad un utente di collegarsi tramite l’account dei social media che già utilizza, come avviene con Twitter. Questo ultimo potrebbe contenere informazioni che associate ai dati pseudonominizzati potrebbero ricondurre all’identità di una persona.
- **Trasferimento Ether:** Attraverso il campo *Interacted With (To)* e i *TRANSFER* allegati con esso, è possibile individuare i trasferimenti di Ether, se presenti, che avvengono tra gli utenti di Yup. Inoltre attraverso i *TRANSFER* è possibile analizzare i passaggi intermedi effettuati per concludere un trasferimento e scoprire eventuali intermediari. Queste informazioni potrebbero essere utili per comprendere la quantità di Ether che viene scambiata tra due account, se ci sono account che spesso effettuano questi trasferimenti verso stessi account e le motivazioni per cui vengono effettuati, ad esempio per pagamenti tra individui.
- **Individuazione Flashbot:** come notato in questa tesi, le transazioni YUP sono soggette alla problematica MEV. In particolare le nostre transazioni sono state vittime di attac-

chi sandwich. Flashbot rappresenta un'organizzazione di ricerca che cerca di mitigare gli attacchi MEV. Si potrebbe cercare di individuare la presenza di transazioni YUP soggette ai flashbot e confrontarle con quelle soggette ad attacchi MEV deterioranti per la blockchain.

Bibliografia

- [1] Guillermo Angeris, Hsien-Tang Kao, Rei Chiang, Charlie Noyes, and Tarun Chitra. An analysis of uniswap markets. *arXiv preprint arXiv:1911.03380*, 2019.
- [2] Andreas M Antonopoulos and Gavin Wood. *Mastering ethereum: building smart contracts and dapps*. O'reilly Media, 2018.
- [3] Bill Bejeck. *Getting Started with Google Guava*. Packt Publishing Ltd, 2013.
- [4] Andreas Bruns, Andreas Kornstadt, and Dennis Wichmann. Web application tests with selenium. *IEEE software*, 26(5):88–91, 2009.
- [5] Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 910–927. IEEE, 2020.
- [6] Chris Dannen. *Introducing Ethereum and solidity*, volume 1. Springer, 2017.
- [7] Anwitaman Datta, Sonja Buchegger, Le-Hung Vu, Thorsten Strufe, and Krzysztof Rzadca. Decentralized online social networks. In *Handbook of social network technologies and applications*, pages 349–378. Springer, 2010.
- [8] Daniel Conte de Leon, Antonius Q Stalick, Ananth A Jillepalli, Michael A Haney, and Frederick T Sheldon. Blockchain: properties and misconceptions. *Asia Pacific Journal of Innovation and Entrepreneurship*, 2017.
- [9] Massimo Di Pierro. What is the blockchain? *Computing in Science & Engineering*, 19(5):92–95, 2017.

- [10] Barbara Guidi, Andrea Michienzi, and Laura Ricci. A graph-based socioeconomic analysis of steemit. *IEEE Transactions on Computational Social Systems*, 8(2):365–376, 2020.
- [11] Barbara Guidi, Andrea Michienzi, and Laura Ricci. Steem blockchain: Mining the inner structure of the graph. *IEEE Access*, 8:210251–210266, 2020.
- [12] Everett Hildenbrandt, Manasvi Saxena, Nishant Rodrigues, Xiaoran Zhu, Philip Daian, Dwight Guth, Brandon Moore, Daejun Park, Yi Zhang, Andrei Stefanescu, et al. Kevm: A complete formal semantics of the ethereum virtual machine. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 204–217. IEEE, 2018.
- [13] John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(03):90–95, 2007.
- [14] Aljosha Judmayer, Nicholas Stifter, Philipp Schindler, and Edgar Weippl. Estimating (miner) extractable value is hard, let’s go shopping! *Cryptology ePrint Archive*, 2021.
- [15] Xing Liu, Bahar Farahani, and Farshad Firouzi. Distributed ledger technology. In *Intelligent Internet of Things*, pages 393–431. Springer, 2020.
- [16] Stefan Loesch, Nate Hindman, Mark B Richardson, and Nicholas Welch. Impermanent loss in uniswap v3. *arXiv preprint arXiv:2111.09192*, 2021.
- [17] Matthias Nadler. A quantitative analysis of the ethereum fee market: How storing gas can result in more predictable prices, 2020.
- [18] Felipe Pezoa, Juan L Reutter, Fernando Suarez, Martín Ugarte, and Domagoj Vrgoč. Foundations of json schema. In *Proceedings of the 25th International Conference on World Wide Web*, pages 263–273, 2016.
- [19] Shahar Somin, Goren Gordon, and Yaniv Altshuler. Network analysis of erc20 tokens trading on ethereum blockchain. In *International Conference on Complex Systems*, pages 439–450. Springer, 2018.

- [20] Friedhelm Victor and Bianca Katharina Lüders. Measuring ethereum-based erc20 token networks. In *International Conference on Financial Cryptography and Data Security*, pages 113–129. Springer, 2019.
- [21] Yang Xiao, Ning Zhang, Wenjing Lou, and Y Thomas Hou. A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys & Tutorials*, 22(2):1432–1465, 2020.
- [22] Weilin Zheng, Zibin Zheng, Hong-Ning Dai, Xu Chen, and Peilin Zheng. Xblock-eos: Extracting and exploring blockchain data from eosio. *Information Processing & Management*, 58(3):102477, 2021.