Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea in Informatica

PROFITABLE STRATEGIES ON ETHEREUM BLOCKCHAIN

Relatore:
Chiar.mo Prof.
Giancarlo Succi

Presentata da:
Gabriele Bernardi

Sessione III
Anno Accademico 2024/2025

Sommario

Questo progetto mira a continuare lo studio della distribuzione di ricchezza nella blockchain di Ethereum cominciato da Francesco Santilli[1]. Dato che lo studio precedente ha individuato che la distribuzione di ricchezza sulla rete è molto eterogenea, con una piccola percentuale di indirizzi che detiene la maggior parte della ricchezza totale, l'obiettivo principale di questo progetto è individuare tipici parametri delle strategie che questi indirizzi utilizzano per arricchirsi. Per fare ciò, abbiamo analizzato esclusivamente i dati pubblici della blockchain utilizzando l'Intelligenza Artificiale per riconoscere schemi di comportamento complessi. I risultati hanno mostrato che le strategie di accumulo passivo sono predominanti tra gli indirizzi profittevoli. Inoltre, si è scoperto che i piccoli investitori riescono ad avere ritorni superiori rispetto alle grandi entità sulle strategie con un basso numero di transazioni giornaliere. Tuttavia, questa profittabilità maggiore è compensata dalla netta superiorità delle istituzioni nella probabilità di avere profitti anche se di dimensioni relativamente inferiori.

Indice

Elenco delle tabelle

Capitolo 1

Introduzione

Il panorama delle transazioni mondiali si sta evolvendo rendendo i trasferimenti di valori sempre più veloci ed economici. Questa evoluzione è portata avanti dalle blockchain, applicazioni web che permettono di condividere token in modo anonimo, veloce ed economico. Queste sono interamente gestite, in principio, in modo completamente decentralizzato e dunque dovrebbero permettere a qualunque utente di trasferire e detenere in sicurezza i propri token in modo indipendente dagli altri utenti. Le blockchain funzionano grazie a dei token accumulabili spesso convertibili in valute FIAT[2] attraverso Centralized Cryptocurrency Exchanges, piattaforme Peer-to-Peer, ATMs o altre piattaforme che accettano pagamenti in criptovalute[3]. La possibilità di convertire facilmente in valute di corso legale i token, ha iniziato una corsa nell'accumulare più token per poi utilizzarli per le spese ordinarie o come investimento sulla blockchain stessa[4]. Proprio per questa ragione gli hedge fund e i fondi d'investimento hanno cominciato a dedicare sempre più attenzione a questo mondo[5] rendendo estremamente efficienti anche questi mercati[6] una volta trascurati rispetto ai mercati finanziari tradizionali.
Questa tesi ha lo scopo di individuare se esistono cluster di strategie che sono riuscite ad accumulare Ether sulla blockchain di Ethereum e le principali caratteristiche di queste strategie. In particolare la ricerca si concentrerà sul cercare peculiarità comuni che raggruppano queste strategie attraverso l'utilizzo dell'intelligenza artificiale.

Capitolo 2

Cos'è una blockchain

La blockchain è un registro digitale condiviso e sicuro, in cui le informazioni vengono raccolte in blocchi collegati tra loro in ordine cronologico. Una volta aggiunto, un blocco non può essere modificato senza alterare tutti quelli successivi, il che rende il sistema molto resistente alle manomissioni. Questo registro non è controllato da un'unica entità, ma è mantenuto da una rete di computer che collaborano per verificarne e garantirne l'affidabilità, eliminando la necessità di intermediari di fiducia e inserendo il concetto di decentralizzazione nel mondo delle transazioni contemporanee.
La prima blockchain che ha fatto sapere dell'esistenza di questa tecnologia, e tuttora detentrice della più grande capitalizzazione di mercato, è Bitcoin[7]. Bitcoin è una blockchain basata sul sistema di sicurezza Proof of Work[8], sistema estremamente dispendioso e lento se confrontato ai nuovi algoritmi di consensus moderni[9].

2.1 Perché sono nate le blockchain

In un mondo sempre più interconnesso, scambiare denaro tramite il sistema bancario tradizionale può portare diversi svantaggi come commissioni elevate, tempo necessario e soprattutto limiti imposti su importi, orari, zone geografiche... Ad esempio i costi medi per una transazione Retail B2B sono dell'1.5% mentre per le transazioni P2P sono del 2.5%[10]. La situazione peggiora se si studiano i paesi in via di sviluppo come la regione dell'Africa subsahariana o dell'America Latina dove i costi si alzano fino ad una media del 3.9% di commissione per transazione[10]. Le blockchain sono nate per risolvere tutti i problemi di questo tipo: decentralizzare il potere per rendere i trasferimenti sempre a disposizione di tutti senza oligopoli. In particolare le blockchain permettono di inviare token in modo sicuro, efficiente, decentralizzato, economico e anonimo a chiunque abbia accesso alla rete. Ad esempio, il costo medio di transazione sulla blockchain di Bitcoin è sceso fino ad arrivare a $1.24[11]. Con 120 bilioni di Dollari Statunitensi processati tramite SWIFT[12] e una media di 44.8 milioni di transazioni al giorno[13] si ottiene una

media di oltre $7000 scambiati per transazione con una commissione media di oltre i $105. Di conseguenza, Bitcoin, con un algoritmo di validazione comunque costoso rispetto ai nuovi standard, ha un costo che è pari all'1% rispetto ai costi applicati dalle banche tradizionali per trasferimenti internazionali.

2.2 Algoritmi di consenso

2.2.1 Proof of Work

Ma perché gli utenti delle blockchain dovrebbero fidarsi che la transazione vada a buon termine? E soprattutto, chi garantisce che i loro token siano al sicuro? Le blockchain funzionano attraverso Algoritmi di Validazione, ovvero algoritmi di consenso che garantiscono l'integrità, la trasparenza e l'immutabilità delle transazioni senza la necessità di un'autorità centrale[14].
Il Proof of Work è un tipo di Algoritmo di Consenso che richiede ai partecipanti della rete, chiamati miner, di risolvere problemi matematici complessi per validare le transazioni e aggiungere nuovi blocchi alla catena. Ad esempio, nella sua applicazione più famosa, Bitcoin, la Proof of Work consiste nel trovare una stringa che, se unita con i dettagli delle precedenti transazioni, produca un hash SHA-256 specifico. Se un miner trova la stringa richiesta, questo la invia al resto della rete che una volta accertato sia la parola corretta, aggiunge il blocco di transazioni alla catena. Questo procedimento rende sicura la rete in quanto per modificare la storia delle transazioni un attaccante dovrebbe ricalcolare l'hash della transazione già confermata e gli hash di tutte le transazioni precedenti a quella in quanto l'hash della transazione stessa dipende da quelli precedenti. Il ricalcolo di tutti questi codici è computazionalmente elevato e dunque disincentivante per eventuali attaccanti. Al contrario i miner, seppur debbano compiere loro stessi molti calcoli per trovare la stringa corretta, sono incentivati a rendere sicura la rete e verificare le transazioni in quanto per ogni transazione verificata, il miner vincitore, riceve una ricompensa. La ricompensa che riceve il miner deriva direttamente da parte delle commissioni che l'utente della rete deve pagare per inviare una transazione.[8] Altro fattore che rende il Proof of Work un algoritmo di consenso estremamente robusto è la decentralizzazione; anche se uno o più miner si accordassero per cambiare la storia della rete questa, essendo distribuita, non è controllabile da singoli individui e serve la maggioranza della potenza di calcolo dei miner per rimuovere o modificare una transazione, ovvero il 50%+1 di tutti i miner della blockchain. La rete di Bitcoin ha raggiunto i 949 EH/s[15] e servirebbero dunque più di 474.5 EH/s per riscrivere la storia delle transazioni della sua blockchain.

2.2.2 Proof of Stake

Come il Proof of Work, anche la Proof of Stake (PoS)[16] è un algoritmo di consenso la cui base è la decentralizzazione e distribuzione. Questo algoritmo è stato introdotto con l'obiettivo di abbassare gli esorbitanti costi del PoW portando un consumo energetico inferiore e introducendo il concetto di scalabilità delle blockchain. Come detto nella sezione sopra, i miner sono in competizione tra loro per indovinare il prima possibile la stringa corretta per ottenere la ricompensa, nella Proof of Stake, questo concetto di competizione tra miner non esiste. Si ridefinisce l'entità che convalida un blocco di transazioni non parlando più di miner ma di Validatori. I nuovi blocchi vengono confermati selezionando un validatore il quale, confermando il blocco, riceve una ricompensa. La scelta del validatore è però determinata da una componente stocastica influenzata dal numero di criptovaluta che il candidato ha "bloccato" (messo in stake) nella blockchain come collaterale[16]. Questo collaterale funge sia da investimento che da deterrente: se il validatore si comporta onestamente validando transazioni corrette allora questo viene ricompensato con degli interessi sullo stake (e quindi un stake più grande implica un guadagno assoluto maggiore) e delle commissioni; se però il validatore non agisce correttamente, validando transazioni fraudolente, attaccando la rete cercando di cambiarne la storia o semplicemente non convalida più blocchi mentre è ancora un validatore, allora gli vengono imposte delle sanzioni sul capitale che ha bloccato nel protocollo. Queste penalità consistono nel rimuovere una parte o il totale dei token depositati come garanzia[17].

2.2.3 Differenze tra PoW e PoS

Le differenze tra Proof of Work e Proof of Stake sono molte ma quelle più significative risiedono nell'efficienza energetica e nel modo in cui la sicurezza è garantita. Se nella Proof of Work la sicurezza viene garantita tramite il costo computazionale ed energetico elevato, la Proof of Stake è basata sul capitale bloccato da ogni validatore. Questo rende la Proof of Stake estremamente più efficiente in quanto per convalidare un blocco di transazioni non è necessario consumare una grande quantità d'energia. Questa differenza emerge chiaramente analizzando la blockchain di Ethereum[2.3], la quale è nata con una tecnologia Proof of Work ma che poi ha cambiato verso l'algoritmo più efficiente. Questo cambio ha ridotto il consumo energetico totale della propria blockchain di oltre il 99.95%[18]. Oltre a differire da un punto di vista della validazione, queste due tecnologie differiscono anche dal punto di vista degli attacchi possibili; se per prendere il controllo di una blockchain basata sul sistema Proof of Work bisogna controllare almeno il 50%+1 della potenza, per attaccare con successo una blockchain che utilizza Proof of Stake si dovrebbe acquistare la maggioranza dei token emessi che, in caso di successo, porterebbe ad una svalutazione di questi e quindi un danno economico per l'attaccante.

2.3 Ethereum

Ethereum è un'applicazione decentralizzata open-source basata sulla tecnologia blockchain che permette la creazione e l'esecuzione di smart contracts e applicazioni decentralizzate[19]. Mentre Bitcoin utilizza un linguaggio progettato per garantire la sicurezza delle transazioni, Ethereum permette l'utilizzo di una macchina virtuale, la Ethereum Virtual Machine (EVM)[20] la quale permette di creare applicazioni eseguibili sulla rete.

2.3.1 Modello account-based

Altra differenza fondamentale che separa Ethereum da Bitcoin è la gestione dello stato. Bitcoin utilizza il modello UTXO (Unspent Transaction Output)[21] dove il saldo è calcolato dalla somma delle transazioni in entrata non ancora spese; Ethereum utilizza invece il modello basato sugli account[19].
Esistono due tipi di account in Ethereum[20]:

- **Externally Owned Accounts**: accounts controllati da chiavi custodite da utenti.
- **Contract Accounts**: accounts controllati da codici ad essi associati (smart contracts) i quali possono interagire con il resto della rete solamente in risposta ad una transazione.

2.3.2 Smart contracts

Ethereum ha introdotto molte innovazioni tra cui l'implementazione degli smart contracts[22]. L'obiettivo di uno smart contract è quello di soddisfare le condizioni imposte nel momento della scrittura dello stesso, eliminando la necessità di un intermediario. Nell'implementazione utilizzata da Ethereum, questi sono script autonomi possedenti un indirizzo specifico ed eseguiti in automatico dalla Ethereum Virtual Machine quando necessario. In particolare, questi sono caricati su un nodo della rete e poi distribuiti all'intera blockchain che dovrà salvarne il codice per poi eseguirlo. Data l'immutabilità della blockchain questi contratti, una volta descritti non possono essere più cambiati garantendone i termini per tutte le parti coinvolte.

2.3.3 Ether e gas

Come descritto in precedenza, sia le blockchain che utilizzano Proof of Work sia quelle che utilizzano Proof of Stake, hanno dei token utilizzati come ricompensa ai miner-validatori per il lavoro svolto nel garantire il corretto funzionamento della rete. Nella blockchain di Ethereum, la criptovaluta nativa è Ether (ETH)[19]. Questa è la valuta necessaria per eseguire una transazione e/o interagire con gli smart contracts, questa commissione si chiama "Gas" ed andrà come ricompensa al validatore (prima del cambio di tecnologia al

miner) che convalida il blocco contentente la transazione. Data la Turing-Completezza della Ethereum Virtual Machine, esiste il rischio che lo smart contract con cui si interagisce entri in un ciclo infinito e dunque, per prevenire ciò, ogni operazione eseguita dalla EVM ha un costo fisso in unità di Gas. Quando l'utente vuole interagire con un'applicazione, specifica il numero massimo di Gas che intende spendere e se l'esecuzione non termina entro questo limite la transazione viene annullata ma la commissione viene comunque pagata ai validatori i quali hanno dovuto comunque eseguire il programma e quindi eseguire un lavoro[20].

2.3.4 The Merge

Come già citato, Ethereum è nata basandosi sulla tecnologia Proof of Work passando poi alla Proof of Stake. Questo passaggio ha preso il nome di "The Merge", riducendo drasticamente il consumo di energia[18] e sostituendo i miner con i validatori.

Capitolo 3

Dati e metodologie

I dati delle transazioni utilizzati sono gli stessi dati impiegati nella ricerca "WEALTH DISTRIBUTION ON ETHEREUM BLOCKCHAIN" di Francesco Santilli[1]. Di seguito viene riportata la metodologia di acquisizione dei dati adottata nel precedente progetto. Le transazioni e i dettagli di queste sono state estratte dal nodo pubblico Ethereum fornito dalla piattaforma "Alchemy". Dato l'elevato numero di blocchi si è deciso di dividere questi ultimi in batch da 100.000 blocchi, ognuno salvato in un file .json. Le transazioni che avvengono tra due nodi pubblici non sono incluse in quanto computazionalmente esigenti. Queste transazioni non sono standard ma sono interazioni avviate da uno smart contract come destinatario un altro smart contract. L'omissione di questi dati potrebbe generare lievi discrepanze nei saldi calcolati rispetto ai saldi reali presenti sulla Blockchain.
Per la presente tesi, si è utilizzata l'infrastruttura High Performance Computing (HPC) dell'università di Bologna sulla quale sono state eseguite sia la fase di pre-elaborazione dei dati[5.3] sia le fasi di pre-training e clustering del modello Deep Embedded Clustering[5.4]. È stata utilizzata una sola GPU NVIDIA L40 per tutti i job anche in caso di interruzione e creazione di un nuovo job. Il workload manager implementato è SLURM con un limite di esecuzione per job pari a 5 giorni consecutivi. Il linguaggio di riferimento per l'intero progetto è stato Python con anche l'utilizzo di SQLite per la gestione efficiente dei dati durante la fase di pre-elaborazione dei dati.

Tabella 3.1: Struttura delle transazioni sulla blockchain di Ethereum

Transaction Hash	l'hash della transazione	(es. 0x10ad32079f4bda63f45650b8f402fb9fd4576ce
Status	lo stato della transazione, indica se la transazione è avvenuta con successo o meno	(es. Success).
Block	numero identificativo del blocco in cui è contenuta la transazione	(es. 23.757.981).
Timestamp	la data precisa di quando la transazione è stata prodotta	(es. 1.698.315.000).
From	l'indirizzo mittente della transazione	(es. 0x885869d5f33fc84962bf87a0CD092814935df4
Interacted With (To)	l'indirizzo destinatario della transazione	(es. 0xdAC17F958D2ee523a2206206994597C13D8
ERC-20 Tokens Transferred	la lista dei token ERC-20 trasferiti nella transazione	(es. 0xdAC17F958D2ee523a2206206994597C13D8
Value	il valore scambiato in Ether	(es. 0.5).
Transaction Fee	La commissione in Ether pagata per effettuare la transazione	(es. 0.00000382131877842).
Gas Price	è il prezzo per unità di lavoro necessaria a convalidare la transazione	(es. 0.06541002 Gwei).

Capitolo 4

Stato dell'arte

L'analisi delle transazioni che avvengono sulle blockchain è un campo di ricerca in rapida espansione dopo che le più grandi criptovalute hanno raggiunto elevate capitalizzazioni superando molti altri asset come l'argento[24]. In questo caso, l'analisi si suddivide in due parti, la prima dedicata all'analisi dei dati on-chain e la seconda dedicata all'utilizzo dell'intelligenza artificiale per studiare proprio questi dati.

4.1 Analisi dei dati on-chain

L'analisi delle transazioni su blockchain è iniziata fin dai primi anni d'utilizzo di Bitcoin con i primi obiettivi posti nel cercare di de-anonimizzare alcuni utenti possessori di quantità particolarmente elevate di Bitcoin. Sebbene si pensi spesso che questo tipo di analisi sia inutile data l'anonimizzazione tramite indirizzo, questo non ha impedito l'analisi attraverso euristiche. Ad esempio, Reid e Harrigan[25] sono stati fra i primi a dimostrare ciò proprio attraverso un'analisi del grafo delle transazioni e euristiche di clustering; in particolare, queste tecniche sono state utili nell'identificare gli indirizzi appartenenti a servizi di wallet custodial o exchange centralizzati. Questo tipo di analisi è più complicata su Ethereum dato l'utilizzo del modello *account-based*[2.3.1] e l'introduzione degli smart contract[2.3.2]. Anche in questo caso però, seppur con un costo computazionale più elevato, ricercatori come Friedhelm Victor[26] sono riusciti a sviluppare euristiche per il clustering degli indirizzi appartenenti a singole entità come gli exchange.

Oltre all'analisi dei dati per la de-anonimizzazione degli utenti la ricerca si è concentrata anche sul capire quali pattern di transazioni permettessero ad un indirizzo di trarre un profitto. Queste ricerche hanno individuato che i profitti più costanti, derivanti dal trading algoritmico, si ottengono quando in uno dei campi delle transazioni è presente un exchange decentralizzato. Philip Daian in "Flash Boys 2.0"[27] ha introdotto il

concetto di *Miner Extractable Value*, mostrando come i bot di arbitraggio ottengono un guadagno utilizzando pratiche come il *front-running* e il *back-running*. Sebbene queste tecniche siano consentite, utilizzate e profittevoli, spesso vanno a discapito dell'utente che inizializza la transazione anche se non impattano in modo sensibile[28]. A differenza di questi studi, la presente tesi mira a identificare gruppi di strategie profittevoli più ampie ed etiche utilizzando tecniche di apprendimento non supervisionato.

4.2 Utilizzo dell'intelligenza artificiale su blockchain

Data l'elevata quantità di dati e la loro difficile comprensione, l'utilizzo di tecniche basate sul Machine Learning è diventato sempre più lo strumento necessario per svolgere una qualsiasi analisi sulle blockchain contemporanee. La maggior parte della letteratura esistente si avvale di approcci che implementano modelli ad *apprendimento supervisionato*[29]. Le tecniche di apprendimento supervisionato permettono di classificare gli indirizzi date delle etichette note; ad esempio Weili Chen[30] ha utilizzato tali algoritmi per identificare schemi Ponzi sulla blockchain di Ethereum.
Tuttavia, in questa tesi non erano disponibili le etichette per ogni account e di conseguenza si è dovuto ricorrere al metodo di *apprendimento non supervisionato*[31]. In particolare, il modello implementato in questa tesi appartiene alla famiglia dei *Deep Clustering*[32]. Tra questi algoritmi è presente anche il *Deep Embedded Clustering*[33] proposto con l'obiettivo di aiutare nel clustering di dataset con relazioni non lineari; questo modello utilizza reti neurali profonde (autoencoder[34]) per imparare rappresentazioni latenti dei dati. Questa tesi applica proprio questo modello sulle transazioni della blockchain di Ethereum per identificare e raggruppare le caratteristiche delle strategie utilizzate dagli indirizzi profittevoli.

Capitolo 5

Analisi sui cluster di strategie profittevoli

Questo capitolo si concentrerà sull'individuazione e analisi dei cluster di strategie profittevoli sulla blockchain di Ethereum attraverso l'utilizzo di modelli Deep Learning[35]. Nello studio è stato utilizzato il modello del Deep Embedded Clustering[33] implementato attraverso tecniche di regolarizzazione per evitare il problema del collasso su un singolo cluster.

5.1 Classici cluster di strategie profittevoli

Nel mondo della finanza classica esistono molti tipi di strategie utilizzate da diversi hedge fund e fondi d'investimento che, seppur radicalmente differenti per quanto concerne i ritorni, hanno tutte caratteristiche comuni. Ad esempio, molti hedge fund utilizzano il Market Making, una strategia che consiste nel fornire liquidità al mercato senza accumulare una grande posizione netta scambiando un gran volume di posizioni.[36] Al contrario, i fondi d'investimento sono solitamente molto più "lenti" nelle scelte operative e preferiscono ridefinire il proprio portafoglio con una frequenza nettamente inferiore rispetto ai precedenti e con una direzionalità molto più decisa.
Si sono dunque identificate le caratteristiche principali di queste strategie affinchè si possano individuare gruppi corrispondenti sulla blockchain di Ethereum. Per identificare queste peculiarità si individuano i cluster di queste strategie nel mondo della finanza classica ed infine se ne estraggono le qualità che li distinguono:

- **Market Making**: Come scritto precedentemente, questa è una strategia che permette di avere bassa esposizione sul sottostante in quanto la posizione netta non è mai elevata. Il guadagno deriva dalla differenza che il Market Maker impone tra i suoi ordini di vendita e quelli di acquisto, ovvero lo spread.

- **Arbitraggio**: Strategia che sfrutta le inefficienze di mercato e consiste nel comprare e vendere contemporaneamente lo stesso asset (o asset perfettamente correlati) quotato su mercati differenti con una differenza di prezzo. Questo permette di ottenere un guadagno quasi privo di rischio ma solitamente non molto elevato data la grande efficienza dei mercati contemporanei.

- **Hedging**: Strategia che mira a ridurre al minimo le perdite piuttosto che massimizzare i profitti. Questa tecnica consiste nel comprare/vendere prodotti derivati[37] per ridurre il rischio di movimenti di prezzo che potrebbero far incorrere il portafoglio in perdite di denaro.

- **High Frequency Trading**: Questa strategia ha la peculiarità di essere accessibile ad un numero ristretto di enti; infatti questa tecnica consiste nell'eseguire un elevatissimo numero di scambi sfruttando algoritmi complessi e tecnologie il più ottimizzate possibile. Questa strategia è anche la più costosa anche da un punto di vista hardware in quanto ogni centimetro di cablaggio tra l'elaboratore e il centro di scambio di dati del mercato di riferimento potrebbe creare latenze importanti.

- **Trend Following**: Strategia estremamente semplice e molto utilizzata, consiste nel seguire il trend principale del prezzo, sia al rialzo che al ribasso, entrando su ritracciamenti, rotture di livelli importanti o altri segnali tecnici.

- **Mean Reversion**: Al contrario della precedente, questa strategia, anch'essa semplice, mira ad entrare nel verso contrario del trend principale assumendo che prima o poi il prezzo torni alla media da cui ha deviato soltanto temporaneamente.

- **Pairs Trading**: Similmente all'arbitraggio anche in questo caso si aprono due posizioni opposte su due asset correlati; tuttavia la relazione tra i due asset non è di equivalenza bensì una relazione statistica. Nell'arbitraggio ad essere coinvolti sono o lo stesso sottostante su due mercati differenti o due sottostanti perfettamente equivalenti (ad esempio il titolo azionario Apple quotato sia sul Nasdaq che sul DAX); nel Pairs Trading invece i sottostanti sono sempre differenti ma con un tasso di correlazione statistica molto elevato (ad esempio due titoli azionari bancari dello stesso paese, settore, dimensione...).

- **Buy and Hold**: Molto probabilmente la strategia d'investimento più comune tra le famiglie e investitori retail ed anche la più semplice tra tutte. La strategia consiste nel comprare uno o più sottostanti con la convinzione che nel lungo periodo il valore di questi aumenterà grazie alla crescita economica generale e/o crescita del valore intrinseco del sottostante.

5.2 Caratteristiche delle strategie

Attraverso lo studio delle strategie sopra riportate con l'utilizzo dei dati a disposizione sulla blockchain di Ethereum si è dunque deciso di utilizzare le seguenti variabili per il raggruppamento dei vari cluster. Per ogni indirizzo presente sulla rete di Ethereum si è dunque calcolata X_i la i-esima transazione, dove i valori in uscita (incluse le fees) sono indicati con segno negativo, mentre i valori in entrata sono indicati con segno positivo. Con $i = 0, \ldots, n$, si definiscono le seguenti grandezze:

- **Outgoing Value** (valore totale in uscita)

$$\text{OutgoingValue} = -\sum_{i=0}^{n} \min\{X_i,\, 0\}$$

  oppure, equivalentemente,

$$\text{OutgoingValue} = \sum_{i=0}^{n} -\min\{X_i,\, 0\}.$$

- **Incoming Value** (valore totale in entrata)

$$\text{IncomingValue} = \sum_{i=0}^{n} \max\{X_i,\, 0\}.$$

- **Daily Transaction Frequency** (frequenza giornaliera delle transazioni) Sia t_i la data della transazione X_i e D l'insieme dei giorni nel periodo considerato calcolato come la data dell'ultima transazione a cui si sottrae la data della prima transazione. Allora:

$$\text{Freq}(d) = \sum_{i=0}^{n} \mathbf{1}_{\{t_i = d\}},$$

  dove $\mathbf{1}_{\{t_i=d\}}$ vale 1 se la transazione i appartiene al giorno d, altrimenti 0. La frequenza media giornaliera risulta:

$$\overline{\text{Freq}} = \frac{1}{|D|} \sum_{d \in D} \text{Freq}(d) = \frac{1}{|D|} \sum_{i=0}^{n} 1 = \frac{N}{|D|},$$

  dove $N = n + 1$ è il numero totale di transazioni.

- **Net Balance** (saldo netto)

$$\text{NetBalance} = \sum_{i=0}^{n} X_i.$$

  Equivalentemente:

$$\text{NetBalance} = \text{IncomingValue} - \text{OutgoingValue}.$$

- **Total Volume** (volume totale)

$$\text{TotalVolume} = \sum_{i=0}^{n} |X_i| = \sum_{i=0}^{n} (\max\{X_i, 0\} - \min\{X_i, 0\}) = \text{IncomingValue} + \text{OutgoingValue}.$$

- **Total Number of Transactions** (numero totale di transazioni)

$$\text{TotalNumberOfTransaction} = N = \sum_{i=0}^{n} 1 = n + 1.$$

Queste variabili riescono a raccogliere in modo esaustivo tutte le strategie sopra riportate, ad esempio chi sarà un High Frequency Trader avrà un numero molto elevato di *Daily Transaction Frequency* o al contrario i Buy and Holder avranno questo valore estremamente basso e un Net Balance elevato.

5.3 Pre-elaborazione dei dati

Data la breve finestra temporale imposta dalle regole dell'HPC in utilizzo l'elaborazione è stata suddivisa in più job sequenziali, capaci di riprendere l'esecuzione dal punto di interruzione del job precedente. Questo è stato realizzato tramite un ingest in streaming: i file JSON sono stati scorsi riga per riga con parser incrementali in modo da non dover mai tenere in memoria l'intero dataset e poter ripartire velocemente esattamente da dove si è interrotta una precedente esecuzione. Durante questa scansione, ogni transazione viene canonicalizzata ovvero tutte le chiavi che le varie fonti adottano (ad esempio from", "fromAddress" o "sender") sono mappate in un vocabolario unico mentre gli importi sono convertiti in ETH e privati della stringa di unità. Questa canonicalizzazione evita di dover gestire a mano i numerosi alias che popolano i dati grezzi e consente in modo uniforme di calcolare il valore trasferito, le gas fee e i timestamp. Per ogni transazione quindi si producono due delta: uno relativo all'indirizzo mittente e uno per quello destinatario in modo da aggiornare separatamente entrambi i bilanci di uscita e di entrata. I delta vengono accumulati all'interno di un unico store SQLite[23] in modo tale che sia in un formato facilmente trasferibile, versionabile e utilizzabile in ambiente HPC. Lo schema SQL è formato da: il saldo netto, il numero di transazioni, i volumi in entrata e in uscita e i timestamp (primo ed ultimo utilizzo oltre al primo deposito); queste sono tutte le statistiche necessarie per calcolare i valori descritti nella sezione precedente. Con un'unica query "INSERT ... ON CONFLICT" i valori vengono aggregati atomicamente senza dover necessariamente effettuare prima una lettura. Sono poi abilitati il journal WAL e un set di pragmi per rendere più veloce l'esecuzione: durante la fase di ingest la sincronizzazione su disco viene allentata in modo da accelerare la scrittura pur continuando a effettuare checkpoint frequenti con il conteggio delle

transazioni elaborate e del più recente hash elaborato. Per ridurre la latenza verso il database i delta vengono inviati a blocchi: il buffer di inserimento (con ampiezza di almeno 5.000 elementi e comunque proporzionale al numero di transazioni processate per job) consente di bilanciare l'uso di memoria e il throughput. Ogni commit, realizzato dopo circa 20.000 transazioni, salva in tabella "meta" lo stato del job così da facilitare la ripresa del lavoro nel batch successivo senza doppioni. Al termine della fase di ingest gli indirizzi interessati vengono filtrati direttamente utilizzando SQL andando a creare il database sul quale verrà poi eseguito il training del modello. Una volta consolidati i saldi per ogni indirizzo, le features (ad esempio valore totale in ingresso, valore totale in uscita, saldo netto, volume totale e frequenza media giornaliera) vengono scritte in un file memmap features_memmap.dat per evitare di saturare la RAM e permette di trattare milioni di indirizzi scorrendo il disco in modo sequenziale. Per calcolare i parametri di normalizzazione senza scorrere l'intero dataset in memoria (data l'enorme mole di dati) si è adottato il campionamento del reservoir: su un campione di 2×10^6 esempi si estraggono la mediana e l'Intervallo Interquartile (IQR), statistiche più robuste della media e la deviazione standard nell'ambito di distribuzioni a code pesanti. Queste statistiche vengono poi salvate in normalization_params.npz e utilizzate per scalare le caratteristiche; i valori eccedenti il $25 \times$ IQR sono infine limitati a ± 25 in modo da evitare che rarità estreme distorcano il clustering, ovvero si eliminano gli outliers. Questa implementazione permette di utilizzare lo stesso database SQLite in esecuzioni successive (magari senza ri-eseguire l'ingestione) in caso di interruzione, ricominciare da punti di controllo e convertire i dati in una forma compatta in modo tale che sia possibile finire la pre-elaborazione dei dati anche in caso di interruzione per limitazioni derivanti dall'HPC.

5.4 Deep Embedded Clustering su Ethereum

5.4.1 Il modello DEC

Il Deep Embedded Clustering (DEC)[33] è un modello di apprendimento non supervisionato progettato per individuare gruppi di dati con caratteristiche simili grazie all'acquisizione della rappresentazione latente dei dati stessi.
L'obiettivo è trasformare il dataset in ingresso in uno spazio di feature più piccolo, preservando esclusivamente le caratteristiche più significative rendendo più semplice l'individuazione di pattern ricorrenti. Questo processo avviene in due fasi distinte:

- **Inizializzazione dei parametri con autoencoder**: in questa fase il dataset viene elaborato tramite un autoencoder[34], il quale apprende le informazioni essenziali necessarie per la ricostruzione dei dati originali. Ciò consente alla rete neurale di calibrare i pesi in modo da ottenere una rappresentazione robusta delle caratteristiche salienti.

- **Ottimizzazione del clustering**: dall'autoencoder addestrato viene rimossa la componente di ricostruzione (decoder); successivamente, si affina l'addestramento della parte preposta alla codifica (encoder) minimizzando una funzione di costo basata sulla divergenza di Kullback-Leibler (approfondita nella sezione successiva[5.4.2]).

In termini intuitivi, il DEC proietta i punti (rappresentanti i singoli indirizzi Ethereum) in un nuovo spazio dove questi tendono ad aggregarsi attorno a dei centroidi, rendendo i cluster progressivamente più definiti e separati.

5.4.2 Implementazione tecnica con regolarizzazioni

Dopo aver concluso la fase di pre-elaborazione dei dati e aver ottenuto il dataset con solamente gli indirizzi che presentano un bilancio netto positivo si è cominciato con la fase di clustering di questi in base alle caratteristiche riportate nella sezione 5.2.
Data l'assenza di etichette, si è dovuto ricorrere a modelli non supervisionati e data la non linearità delle features, nemmeno una normale analisi regressiva lineare sarebbe stata possibile. Per risolvere questi due problemi e avere comunque un metodo efficiente, si è utilizzato il modello *Deep Embedded Clustering* (DEC)[33] il quale, grazie all'autoencoder[34], consente di individuare le strategie profittevoli grazie alla struttura latente dell'autoencoder, il quale elimina i rumori. Come input del modello, si sono utilizzate le caratteristiche precedentemente specificate in forma vettoriale per avere una gestione più semplice. Per avere una clusterizzazione corretta, si è resa necessaria la normalizzazione dei vettori; questa è avvenuta tramite mediana e intervallo interquartile calcolati su un campione casuale. Si sono preferite queste due metriche in quanto i dati on-chain presentano spesso code pesanti e dunque una standardizzazione classica non è robusta. Inoltre, dato che l'obiettivo della ricerca è quello di individuare cluster di strategie, sono stati troncati gli outlier a $\pm 25 \times \text{IQR}$ per evitare che dominassero la dinamica dell'ottimizzazione.
L'encoder[34] restituisce embedding normalizzati a norma unitaria affinchè l'informazione sia concentrata sulle direzioni dei parametri piuttosto che sulle loro grandezze. Questo permette di rendere l'aggiornamento dei centroidi durante la fase di training più efficiente.

Il pre-training del modello è stato effettuato seguendo quanto descritto nelle note descrittive del DEC[33] ovvero utilizzando una funzione di loss che corrisponde allo scarto quadratico medio tra input e la ricostruzione proveniente dell'autoencoder stesso. Durante la fase di pre-training si è utilizzato un Dropout del 5% e un gradient clipping i quali mantengono la fase iniziale stabile. Rimanendo nella fase d'apprendimento, si è utilizzata una riduzione del learning rate una volta che si raggiungeva un plateau, permettendo un pre-training più fine in prossimità della convergenza. In questa fase il modello ha dunque ottenuto i vari pesi necessari sui neuroni affinchè possa imparare come classificare le transazioni nella fase di training vera e propria.

Mantenendo il decoder attivo con un peso ridotto sulla loss, si è dunque passati alla fase di training o clustering. I centroidi sono stati inizializzati tramite Mini-Batch K-Means sugli embedding[38] così da poter evitare di caricare in memoria l'intero dataset ma caricare campioni (fino ad un milione di indirizzi) senza saturare la memoria a disposizione. Nel caso in cui almeno uno dei cluster risultasse troppo popolato (oltre al 92% del campione presente in un cluster) alla fine del clustering, si riesegue un nuovo K-Means sugli embedding per riallineare i centroidi.
Durante la fase di training vera e propria, il cuore della funzione di loss corrisponde alla divergenza Kullback-Leibler[39], questa metrica è particolarmente adatta per l'analisi in questione dato che restituisce un numero sempre maggiore o uguale a 0 dove 0 indica due distribuzioni identiche. Infatti questa misura quantifica quanto sono simili due distribuzioni di probabilità; la distribuzione bersaglio viene ottenuta elevando q (dove q rappresenta la distribuzione di probabilità per ogni indirizzo di appartenere ad un determinato cluster) alla potenza γ e normalizzando per le frequenze di cluster, in questo modo gli indirizzi assegnati con maggiore confidenza influenzano di più l'aggiornamento. Anche se è la principale metrica che il modello punta a minimizzare durante il training, questa non è la sola in quanto viene affiancata anche dai parametri di regolarizzazione descritti in seguito.
Il parametro γ, che controlla l'affinamento dei pesi, è stato reso dipendente dall'epoca di training in cui ci si trova, passando da 1.0 a 1.6 per evitare aggiornamenti troppo aggressivi nelle prime fasi. In particolare, γ controlla quanto "raggruppare" un determinato cluster, ovvero avvicinare gli indirizzi associati ad un centroide affinchè questo risulti più *sharp*. Il passaggio da un γ di 1.0 a 1.6 permette di ottenere una distribuzione il più simile possibile all'inizio del training, ovvero nella fase dove il campione è più rumoroso, e man mano che le epoche avanzano si comincia ad aumentare il parametro per effettivamente rendere i cluster più separati tra loro e associare con più sicurezza un indirizzo ad un cluster specifico.

La divergenza Kullback-Leibler, il controllo graduale di γ e il K-Means dovrebbero permettere un clustering che eviti il collasso su un solo gruppo (centroide), tuttavia, per migliorare ulteriormente la stabilità si è deciso di applicare un insieme di regolarizzazioni ispirate ai lavori su DeepCluster[40] le quali completano la funzione di loss del training:

- Penalizzazione dell'entropia negativa di q per favorire assegnazioni meno concentrate nella fase iniziale di training.
- Vincolo sulla dimensione media dei singoli cluster. All'inizio della fase di training il modello imposta questo vincolo in modo "debole" permettendo a questo di esplorare anche configurazioni sbilanciate ma, con il passare delle epoche, questo vincolo diventa sempre più forte imponendo un bilanciamento più stretto; questo permette che i cluster non muoiano o dominino.

- Applicazione di una softmax sulle distanze normalizzate tra i punti (indirizzi) e i centroidi rendendo più uniforme la distribuzione di probabilità dei punti sui centroidi cosicchè ognuno riceva probabilità comparabili e nessuno viene sovraccaricato.
- Penalizzazione sull'inverso delle distanze tra i vari centroidi per definire ancora di più la differenza e ridurre sovrapposizioni.

5.5 Metodi per l'analisi dei cluster

Terminata la fase di addestramento del DEC, i pesi vengono congelati e con questi pesi, per ciascun indirizzo, il modello attribuisce a ogni indirizzo una distribuzione di cluster di appartenenza. Una volta ottenuta questa distribuzione (ovvero il vettore q), si identifica il valore massimo e si assegna a quello specifico indirizzo il cluster corrispondente a quel valore. In particolare, ogni campo di q è formato dalla distribuzione t di Student con moda corrispondente ai centroidi. In base alla dimensione del campione, vengono assegnati i cluster di appartenenza per ogni indirizzo in blocchi variabili da 50 a 200 mini-batch in modo tale da utilizzare la potenza della Scheda Grafica NVIDIA L40 a disposizione. Date le 10 strategie individuate nella sezione 5.1 si è deciso di inserire un massimo di 15 cluster per avere un margine del 50% in caso il modello avesse individuato più di 10 strategie con metriche che indicavano i centroidi di queste molto distanti e le loro distribuzioni poco sovrapposte. Questo numero è stato successivamente portato a 10 date le scarse performance nell'individuare molte strategie; queste sono state rilevate attraverso calcoli descritti successivamente in questa sezione.

Nella fase di valutazione del campione, se al termine delle epoche di clustering viene rilevato che si ha un cluster che raggruppa almeno il 92% di tutti gli indirizzi allora gli embedding congelati vengono riallenati con un nuovo Mini-Batch K-Means su 1 milione di indirizzi; si è scelto 1 milione di indirizzi come campione il quale corrisponde a circa 16 GB di spazio occupato, abbondantemente al di sotto dei 48 GB disponibili durante l'analisi. Ulteriore fattore che aiuta a prevenire il collasso su un singolo cluster sono le già citate regolarizzazioni utilizzate nella fase di Training dell'autoencoder.

Terminata la fase di assegnamento degli indirizzi si calcolano le metriche quantitative utilizzate per verificare la qualità globale dei cluster presenti. In particolare, vengono calcolate le seguenti misure:

- Entropia
- Numero effettivo di cluster
- Coefficiente di Gini

Grazie al Numero effettivo di cluster, si riesce a capire quanti cluster effettivamente il modello è riuscito a trovare negli indirizzi dati in input. Le altre due metriche aiutano invece a capire se i vincoli mutuati da Caron[40] sono in grado di produrre delle partizioni equilibrate o meno; un'entropia vicina a $\ln(10) \approx 2.302$ indica che i cluster sono quasi uniformi (ovvero gli indirizzi sono distribuiti equamente tra i vari cluster) e un coefficiente di Gini vicino a 0 indica che nessun comportamento domina. Per invece fare inferenza statistica sui cluster si utilizza un campione di 500.000 indirizzi per contenere i tempi d'esecuzione mantenendo comunque un numero statisticamente più che adeguato dato il campione di 7.700.724. Su questo campione vengono calcolate le seguenti misure:

- Silhouette: la Silhouette è una misura che può assumere valori compresi tra -1 e +1, questa viene calcolata per tutti i punti del campione e indica quanto il punto è vicino al proprio centroide. In particolare, un valore vicino a 1 indica che gli indirizzi appartenenti ad un determinato cluster, sono più vicini al centroide del proprio cluster piuttosto che ad un altro cluster, ovvero il cluster non si sovrappone con gli altri gruppi. Più questa misura decresce più gli indirizzi presenti nell'insieme sono vicini ai centroidi di altri cluster.
- Calinski-Harabasz[41]
- Davies-Bouldin[42]

Queste misure vengono calcolate sugli embedding normalizzati del DEC.

Per avere un'interpretazione economica dei cluster, i cluster sono stati de-normalizzati utilizzando le mediane e gli IQR, salvati durante la fase di pre-elaborazione dei dati, calcolando per ogni cluster la sua media, deviazione standard, minimo e massimo. Grazie a questi risultati è ora possibile studiare i comportamenti degli indirizzi nei vari cluster.

5.6 Significato dei cluster

I cluster ottenuti come output dal modello rappresentano dei sottoinsiemi degli indirizzi del campione sottoposto al DEC[5.4]. Ogni gruppo contiene al suo interno gli indirizzi che condividono delle caratteristiche[5.2] simili nelle loro transazioni. Questi cluster vengono creati scegliendo, per ogni indirizzo, il centroide più vicino come descritto nella sezione precedente[5.5]. Quindi, ogni cluster rappresenta l'insieme di indirizzi aventi le caratteristiche che il modello ha identificato come più simili al centroide corrispondente del gruppo.

Si noti che non esiste un cluster le cui specifiche portino ad una perdita di Ether in quanto il campione in ingresso al modello contiene solamente gli indirizzi profittevoli, ovvero con un *Net Balance* positivo. Questi indirizzi sono ottenuti grazie alla fase di

pre-elaborazione[5.3] delle transazioni e filtrati grazie a SQLite.
In generale, ogni indirizzo di ogni cluster ha generato un profitto, l'obiettivo di questa ricerca era individuare, se possibile, le caratteristiche più comuni di questi gruppi.

Capitolo 6

Risultati dell'analisi

6.1 Metriche globali

Il modello, su 7.700.724 di indirizzi che hanno presentato un bilancio finale maggiore del valore della prima transazione in entrata, ha individuato 10 cluster distinti. La **Silhouette** globale di **0.397** indica dei cluster mediamente moderatamente definiti e infatti, come descritto in seguito, solamente 3 cluster hanno una Silhouette abbastanza elevata da poter confermare la presenza di un gruppo ben definito. L'**entropia** ottenuta è pari a **1.754**, al di sotto dell'entropia massima teorica di $\ln(10) \approx 2.302$ indicando che la distribuzione degli indirizzi non è uniforme tra i vari cluster. Questa disuguaglianza tra i gruppi non indica obbligatoriamente un assegnamento scorretto degli indirizzi ma solamente che sono presenti caratteristiche comuni a più indirizzi (e dunque gruppi più grandi) mentre caratteristiche più rare raggruppano meno portafogli. Questa eterogeneità della distribuzione degli indirizzi è confermata anche dal **coefficiente di Gini** il quale si attesta a **0.776**, molto vicino a 1. Un valore vicino a 1 indica che pochi gruppi contengono la maggior parte degli indirizzi mentre i restanti ne contengono meno. Queste ultime due metriche già mostrano come è presente una forte preferenza per determinati tipi di comportamenti simili tra loro, come mostrato dal Cluster 5 il quale da solo raccoglie circa il 40% del campione.

6.2 Analisi intra-cluster

L'analisi dei dati per i cluster singoli ha mostrato subito come 3 cluster (5, 6 e 9) raccolgono più del 68% degli indirizzi profittevoli su Ethereum i quali presentano inoltre una Silhouette significativamente più alta della media indicando delle strategie ben definite e distinte dalle altre.

Tabella 6.1: Riepilogo di tutti i cluster

Cluster	Count	Silhouette	In (ETH)	Out (ETH)	Net (ETH)	Freq/Day	Tot Tx
0	1,734	-0.506	0.41	0.07	0.31	1.35	6.0
1	101,929	-0.470	1.66	0.00	1.66	0.11	9.2
2	539,064	0.088	2.32	0.01	2.12	17.84	8.0
3	815,776	-0.123	7.00	0.02	6.98	0.21	10.6
4	570,829	-0.240	0.62	0.00	0.60	1.31	17.4
5	**3,069,814**	**0.755**	**0.26**	**0.00**	**0.25**	**0.16**	**4.0**
6	**1,245,766**	**0.353**	**13.12**	**0.20**	**6.71**	**0.14**	**19.5**
7	589	-0.199	3.58	0.10	3.46	0.15	10.2
8	429,292	-0.019	16.73	0.17	4.46	2.91	29.6
9	**925,931**	**0.590**	**0.73**	**0.02**	**0.56**	**29.59**	**3.0**

6.2.1 Cluster 5: Utenti retail

Il cluster 5 è verosimilmente il più "importante" dei 10 individuati, questo gruppo ha una Silhouette di 0.755, la più alta tra tutti, la quale indica una forte coesione e definizione della distribuzione. Questo insieme raccoglie 3.069.814 indirizzi ovvero quasi il 40% di tutto il campione.
Guardando ai dati presenti nella tabella 6.2 si notano subito le due feature con il rapporto $\frac{FeatureMean}{FeatureVariance}$ più alto sono **Outgoing Value** e **Tx Frequency (per day)**. Queste due caratteristiche sono infatti quelle che caratterizzano questo cluster indicando poche transazioni e soprattutto pochi Ether in uscita dal portafoglio. Il Net Value positivo e quasi identico all'Incoming Value indica che questi indirizzi hanno ricevuto una piccola somma di Ether (Incoming Value medio di 0.258) e che hanno preferito mantenere il proprio saldo invariato oppure sono account abbandonati. Questa tipologia di utilizzo è facilmente riconducibile ad una strategia di tipo **Buy and Hold** per piccoli investitori o di utenti che hanno voluto utilizzare la blockchain abbandonandone l'uso dopo poche transazioni. In caso fosse corretta l'ipotesi dell'utilizzo della strategia Buy and Hold allora gli utenti riescono ad arricchirsi con l'aumento di prezzo di Ether nel tempo. In particolare, nei mercati finanziari tradizionali i prezzi dei maggiori indici globali tendono a crescere esponenzialmente[43], peculiarità che si è notata anche con i token delle blockchain maggiori[44]. Questo tipo di crescita implica che più tempo si detiene un asset e più questo porta guadagni sempre maggiori. Questo è confermato dallo studio condotto da Lukáš Pichl e Taisei Kaizoji[45] i quali hanno stimato un ritorno giornaliero di Bitcoin (contro dollari statunitensi) dello 0,328%. Un utente che detiene 1 Bitcoin avrebbe dunque guadagnato dopo un anno circa il 230,44% in dollari mentre se avesse tenuto quel singolo token per 5 anni avrebbe avuto un rendimento del 39.294%. Quindi il fattore dominante di questa strategia risiede nel tempo di detenzione del token piuttosto che nella precisione degli ingressi.

Tabella 6.2: Statistiche del cluster 5

Feature	**Valore medio**	**Varianza**
Outgoing Value (ETH)	0.0012	0.000027
Incoming Value (ETH)	0.258	0.228
Tx Frequency (per day)	0.157	0.064
Net Value (ETH)	0.254	0.229
Total Transactions	4.00	4.79

6.2.2 Cluster 6: Whales

Il Cluster 6 raggruppa il 16.17% degli indirizzi (1.245.766) del campione rendendolo così il gruppo più grande dopo il Cluster 5. La sua Silhouette di 0.353, sebbene considerata non ottimale, conferma la struttura della distribuzione.
I dati mostrano che questo è un gruppo estremamente eterogeneo, date le grandi varianze che dominano tutte le features ad eccezione dell'*Outgoing Value*, ma che mostra comunque una netta superiorità rispetto agli altri gruppi quando si analizza il *Net Value*. In particolare, il Cluster 6 ha il secondo valore più grande di *Net Value* secondo solo al Cluster 3 con un distacco di 0.27 ETH e davanti al cluster 8 di 2.25 ETH. Come descritto in seguito, il Cluster 3 fa parte dei cluster individuati con Silhouette negativa e quindi senza una struttura ben identificata, il che mette ulteriormente in risalto il Cluster 6. Alti volumi di Ether in ingresso 13.12 ETH sono accompagnati da una varianza di queste features estremamente elevata pari a 331.98 ETH. Statisticamente, una media elevata, una varianza molto elevata e un limite inferiore (in questo caso limite imposto a 0 dato che l'indirizzo per essere profittevole deve avere un *Incoming Value* ¿ 0) portano ad una distribuzione asimmetrica positiva. Questo significa che la gran parte dei portafogli si trova al di sotto della media ma questa è spinta al rialzo dalla presenza di entità con valori d'ingresso estremamente elevati. Questi grandi volumi in ingresso, assieme ad una *Tx Frequency (per day)* bassa (considerando anche gli estremi suggeriti dalla varianza) e un *Outgoing Value* decisamente inferiore rispetto all'*Incoming Value* sono tipiche caratteristiche delle cosiddette **Whales**[46]. Date le grandi varianze il gruppo è decisamente molto eterogeneo ma ciò che accomuna gli indirizzi di cui ne fanno parte non è l'entità dei volumi ma l'accumulo lento e costante a lungo termine. Anche questo comportamento è tipico delle strategie **Buy and Hold** dove però, a differenza del Cluster 5[6.2.1], ad effettuare le transazioni in questo gruppo sono entità che generalmente detengono molti più Ether. Anche in questo cluster, come in quello precedente[6.2.1], poichè la strategia utilizzata è verosimilmente la stessa, questa ha le stesse caratteristiche descritte precedentemente ovvero guadagni contenuti (relativamente al capitale investito) nelle prime fasi dell'investimento ma crescenti con l'avanzare del tempo. Una differenza tra questo cluster e quello precedente risiede nei guadagni assoluti in quanto anche una piccola variazione di prezzo di Ether può portare grandi profitti per questi investitori.

Tabella 6.3: Statistiche del cluster 6

extbfFeature	**Valore medio**	**Varianza**
Outgoing Value (ETH)	0.203	0.0089
Incoming Value (ETH)	13.12	331.98
Tx Frequency (per day)	0.141	0.207
Net Value (ETH)	6.707	109.08
Total Transactions	19.52	611.99

6.2.3 Cluster 9: Utenti "flash"

Il Cluster 9 è il terzo insieme più popoloso e rappresenta il 12.02% del campione (925.931 indirizzi). La Silhouette di 0.59 indica una buona separazione dagli altri gruppi e una struttura ben definita.
La caratteristica più interessante di questo gruppo si trova nell'elevato numero di transazioni rapportato ad una varianza decisamente più bassa. I valori in ingresso sono mediamente bassi ma in questo caso si ha una varianza estremamente elevata come per il numero totale di transazioni e per il *Net Value*. Questi dati indicano che questi sono indirizzi che utilizzano lo stesso pattern: grande numero di transazioni in brevissimo tempo e non utilizzare più l'indirizzo. Questo comportamento può essere ricondotto a svariati motivi:

- Indirizzi "Usa e Getta"
- Cacciatori di Airdrop[47]
- Trader

Il terzo metodo d'utilizzo raccoglie moltissime opzioni e dunque si elencano tutte quelle possibili in seguito. Ciò che accomuna tutte queste strategie risiede nella loro velocità, non sono riconducibili a strategie *Buy and Hold* o *High Frequency Trading* in quanto la prima ha una frequenza di transazioni nettamente inferiore mentre la seconda nettamente superiore. Altro fattore da tenere in considerazione è il numero ridotto di *Total Transactions* se confrontato con la media di *Tx Frequency (per day)*, questo potrebbe indicare che i Trader in questione si sono accontentati di molti profitti (i valori in ingresso sono mediamente quasi 43 volte più elevati dei valori in uscita) in breve tempo per poi non utilizzare più il portafoglio. Questo rapporto tra valori in ingresso e valori in uscita è cruciale per capire quanto un trader è profittevole; nelle piattaforme decentralizzate, per poter aprire una posizione speculativa bisogna depositare un collaterale che viene poi restituito una volta chiusa la posizione (se la posizione viene liquidata il collaterale viene trattenuto dalla piattaforma)[48]. I valori uscenti (contenuti) se confrontati ai valori in ingresso (molto più elevati) fanno pensare proprio ad un comportamento di questo tipo:

il trader deposita il collaterale che viene utilizzato come garanzia per il margine di una posizione speculativa che viene poi chiusa in positivo. Notare come il *Net Value* è di 0.563, più del 20% inferiore rispetto alla differenza tra *Incoming Value* e *Outgoing Value* indicando che la percentuale di posizioni vincenti è l'80% anche considerando solamente gli indirizzi con *Net Value* positivo. Questo sottolinea la difficoltà nell'avere una strategia di trading profittevole per un periodo prolungato. Di seguito le strategie di Day-Trading compatibili con i dati del Cluster:

- Arbitraggio
- Trend Following
- Mean Reversion
- Pairs Trading

Il profilo dei guadagni in questo cluster è opposto a quello dei precedenti[6.2.16.2.2]: i guadagni sono moderati ma si ottengono subito. Il profitto non deriva dall'interesse composto, come nel caso del *Buy and Hold*, bensì dallo sfruttamento di opportunità che esistono solo per un periodo limitato di tempo. Una volta individuato un possibile ingresso nel mercato, questo viene preso e chiuso il prima possibile per non aumentare troppo la propria esposizione al sottostante. Con tale attività i guadagni possono essere elevati sin da subito ma battere l'indice di riferimento è molto difficile nel lungo periodo e quindi anche i guadagni sono inferiori rispetto ad un *Buy and Hold*[49].

Tabella 6.4: Statistiche del cluster 9

extbfFeature	**Valore medio**	**Varianza**
Outgoing Value (ETH)	0.017	0.0034
Incoming Value (ETH)	0.728	8.68
Tx Frequency (per day)	29.59	6.26
Net Value (ETH)	0.563	6.84
Total Transactions	2.97	18.38

6.2.4 Altri cluster

I cluster rimanenti mostrano valori di Silhouette molto vicini allo 0 o negativi, indicando che gli indirizzi campionati per i cluster sono mediamente più vicini ai centroidi di altri gruppi rispetto al proprio. Questo implica una separazione tra i cluster meno netta e più incerta.

- **Cluster 2**: Con una frequenza media di 17.8 transazioni al giorno e un volume totale di circa 2.3 Ether questo cluster raccoglie il 7% del campione e potrebbe

rappresentare bot di trading a bassa frequenza. Tuttavia la Silhouette bassa (0.088) indica che non è possibile eseguire un'analisi certa data la grande sovrapposizione con altri gruppi.

- **Cluster 8**: Con una frequenza di transazioni non elevata di 2.91 e un volume in ingresso di quasi 17 Ether questo gruppo potrebbe rappresentare wallet di exchange. La Silhouette negativa rende però tale analisi poco sicura.

- **Cluster 0, 1, 3, 4, 7**: Questi cluster mostrano tutti Silhouette fortemente negativa indicando che sono probabilmente rumore di fondo.

Capitolo 7

Conclusioni

L'utilizzo del Deep Embedded Clustering per il raggruppamento di indirizzi profittevoli con comportamenti simili ha evidenziato dinamiche che divergono parzialmente dai risultati dello studio precedente[1]. Tale ricerca evidenziava una forte concentrazione di ricchezza in pochi indirizzi mentre la grande maggioranza dei portafogli deteneva una piccola somma di Ether. Al contrario, l'analisi eseguita attraverso il clustering delle strategie mostra come gli indirizzi di utenti retail profittevoli siano significativamente più redditizi di quelli istituzionali. In particolare, il cluster $5^{6.2.1}$ (identificato come l'insieme degli utenti retail i quali applicano strategia Buy and Hold) presenta un rapporto medio tra i valori in ingresso e valori in uscita di Ether di oltre 215,43; al contrario, gli indirizzi appartenenti al cluster $6^{6.2.2}$, pur muovendo volumi di ordini di grandezza superiori, hanno un rapporto di profittabilità di circa 64,78. Questa differenza è elevata seppur entrambi i gruppi sembrino utilizzare la stessa strategia del Buy and Hold, probabilmente dovuto a maggiore flessibilità nelle decisioni dei punti e tempi d'ingresso.

L'analisi non rivela solamente che gli utenti retail profittevoli sono più efficienti rispetto alle grandi entità ma anche una grande disparità all'interno dei retail stessi. Infatti, se unissimo gli indirizzi presenti sia nel cluster 5 che nel cluster 9 otterremmo circa il 52% di tutto il campione. Secondo un recente studio, gli indirizzi totali utilizzati da utenti retail sarebbero circa il 95%[50] di tutti gli indirizzi presenti sull'intera blockchain di Ethereum per un totale di circa 331.500.000 account retail. La somma di tutti gli account presenti nei due cluster precedentemente menzionati è di 3.995.745 ovvero solamente l'1,27% dell'intera popolazione retail. Anche considerando lo scenario migliore e assumendo che l'interezza dei cluster 0, 1, 2, 3, 4 e 7 sia popolata da soli utenti retail, la popolazione totale salirebbe a 6.025.666 ovvero circa l'1.82%.
Al contrario, le whales, che rappresentano circa il 5%[50] di tutti gli utenti della blockchain, popolano verosimilmente l'interezza dei cluster 6 e 8 con un totale di 1.675.058 ovvero circa lo 0.48%.

Di conseguenza, circa il 9,6% dei grandi indirizzi è profittevole mentre la percentuale si riduce a solamente il 1.92% per quanto riguarda gli utenti retail.
In conclusione, sebbene i retail profittevoli abbiano mediamente un fattore di profitto oltre 3 volte maggiore rispetto agli account istituzionali, questi ultimi hanno una probabilità mediamente 5 volte maggiore nell'essere profittevoli.

Confrontando ora i due cluster maggiori, questi racchiudono i due tipi di utenti: retail e istituzionali. Entrambi i gruppi sono caratterizzati da una bassa frequenza di transazioni giornaliere, tipica caratteristica di strategie passive come il *Buy and Hold*. Questo conferma che, come nel mondo della finanza tradizionale, anche nelle criptovalute, un'eccessiva attività di trading è causa di performance inferiori[51]. Ciò emerge chiaramente confrontando il cluster 5 col cluster 9: entrambi hanno un volume di scambi limitato ma gli indirizzi contenuti all'interno del primo gruppo sono circa 3,32 volte quelli contenuti nel secondo. Ciò implica che gli utenti con bassa frequenza hanno una probabilità più elevata di essere profittevoli rispetto a quelli che operano più spesso, anche con capitali simili. Inoltre, anche il coefficiente di redditività tra questi due gruppi è molto differente in quanto il cluster 5 ha un coefficiente di circa 215,43 mentre il cluster 9 di circa 43,65, quasi 5 volte più piccolo. Questa grande disparità è credibilmente ampliata dalle commissioni presenti su Ethereum, queste infatti, pur essendo contenute, riducono significativamente profitti in caso di operatività frequente come nel cluster 9.

Infine, in questo studio la definizione di "profitto" si riferisce all'accumulo di Ether e non all'accumulo di Ether convertiti in valute FIAT[2]. L'elevata percentuale che le strategie passive occupano nell'insieme degli indirizzi profittevoli (solamente il cluster 5 e il cluster 6 sommati raccolgono oltre il 56% di tutto il campione) conferma che tra questi attori Ether è considerato una riserva di valore o asset d'investimento con un orizzonte di lungo periodo piuttosto che come asset speculativo. Ciò suggerisce una certa parità di opportunità nell'accumulazione di questo token in quanto dimostra come i retail siano più efficienti nell'accumulare con bassa operatività rispetto a grandi operatori, scenario completamente differente rispetto alla finanza tradizionale dove i grandi gestori di fondi e hedge fund dominano la scena dei mercati mondiali[52].

In sintesi, l'analisi conferma che la maggior parte degli indirizzi profittevoli utilizza strategie basate sulla pazienza (dove il guadagno dipende da quanto tempo si detiene il token) rispetto ad un'operatività più attiva come il trading, che pur essendo presente, rappresenta una minoranza con un coefficiente di profitto inferiore.

Bibliografia

[1] Francesco Santilli. *WEALTH DISTRIBUTION ON ETHEREUM BLOCKCHAIN*. 2024.

[2] Naresh Aggarwal. *Central Bank Digital Currencies - What is all the fuss?*

[3] Angelo Corelli. *Cryptocurrencies and Exchange Rates: A Relationship and Causality Analysis*. 2018.

[4] Andrea GRIFONI. *Improving the digital financial literacy of crypto-asset users*. 2025

[5] Alternative Investment Management Association. *Press release: More than a third of traditional hedge funds now invest in digital assets, nearly double a year ago: Global Crypto Hedge Fund Report 2022*. 2022

[6] Emilio Barucci, Giancarlo Giuffra Moncayo, Daniele Marazzina. *Market impact and efficiency in cryptoassets markets*. 2023.

[7] Julius Mansa. *What Is Bitcoin? How to Buy, Mine, and Use It*. 2025.

[8] Scott Nevil. *Understanding Proof of Work (PoW) in blockchain: Key Mechanism Explained*. 2025.

[9] Moritz Platt, Johannes Sedlmeir, Daniel Platt, Paolo Tasca, Jiahua Xu, Nikhil Vadgama, Juan Ignacio Ibañez. *The Energy Footprint of blockchain Consensus Mechanisms Beyond Proof-of-Work*. 2022.

[10] Financial Stability Board. *Annual Progress Report on Meeting the Targets for Cross-border Payments*. 2023.

[11] Archana Jain, Chinmay Jain, Karolina Krystyniak. *blockchain transaction fee and Ethereum Merge*. 2023.

[12] Barry Eldan, Kathleen Kinder. *XRP vs. SWIFT Statistics 2025: Transaction Speed, Fees and Adoption*. 2025.

[13] Zennon Kapron. *Why SWIFT Remains Indispensable For Cross-Border Payments*. 2023.

[14] Ankit Kumar Jain, Nishant Gupta, Brij B. Gupta. *A survey on scalable consensus algorithms for blockchain technology*. 2024.

[15] Will Canny. *Bitcoin Network Hashrate Returned to All-Time Highs in August: JPMorgan*. 2025.

[16] Sunny King, Scott Nadal. *PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake*. 2012.

[17] Vitalik Buterin, Virgil Griffith. *Casper the Friendly Finality Gadget*. 2017.

[18] Carl Beekhuizen. *Ethereum's energy usage will decrease by 99.95%*. Ethereum Foundation Blog, 2021.

[19] Vitalik Buterin. *Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform*. 2013.

[20] Gavin Wood. *Ethereum: A Secure Decentralised Generalised Transaction Ledger*. 2014.

[21] Satoshi Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. 2008.

[22] Nick Szabo. *Smart Contracts*. 1994.

[23] Kevin P. Gaffney, Martin Prammer, Larry Brasfield, D. Richard Hipp, Dan Kennedy, Jignesh M. Patel. *SQLite: Past, Present, and Future*. 2018.

[24] Navdeep Singh. *At $1.7 trn m-cap, Bitcoin beats silver to become 8th largest asset in world*. 2024.

[25] Fergal Reid, Martin Harrigan. *An analysis of anonymity in the bitcoin system*. 2013.

[26] Friedhelm Victor. *Address Clustering Heuristics for Ethereum*. 2020.

[27] Philip Daian et al. *Flash Boys 2.0: Frontrunning, Transaction Reordering, and Consensus Instability in Decentralized Exchanges*. 2019.

[28] Kaihua Qin, Liyi Zhou, Arthur Gervais. *Quantifying Blockchain Extractable Value: How dark is the forest?*. 2021.

[29] Tammy Jiang, Jaimie L Gradus, Anthony J Rosellini. *Supervised machine learning: A brief primer*. 2021.

[30] Weili Chen et al. *Detect Ponzi Schemes on Ethereum: Towards Healthier Blockchain Technology*. 2018.

[31] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. 2006.

[32] Yazhou Ren, Jingyu Pu, Zhimeng Yang, Jie Xu, Guofeng Li, Xiaorong Pu. *Deep Clustering: A Comprehensive Survey*. 2024.

[33] Junyuan Xie, Ross Girshick, Ali Farhadi. *Unsupervised Deep Embedding for Clustering Analysis*. 2016.

[34] Xifeng Guo et al. *Deep Clustering with Convolutional Autoencoders*. 2017.

[35] Shi Dong, Ping Wang, Khushnood Abbas. *A survey on deep learning and its applications*. 2021.

[36] Tanmoy Chakraborty, Michael Kearns. *Market Making and Mean Reversion*. 2011.

[37] Shawkat Hammoudeh, Michael McAleer. *Risk management and financial derivatives: An overview*. 2013.

[38] David Sculley. *Web-Scale K-Means Clustering*. 2010.

[39] Solomon Kullback, Richard A. Leibler. *On Information and Sufficiency*. 1951.

[40] Mathilde Caron, Piotr Bojanowski, Armand Joulin, Matthijs Douze. *Deep Clustering for Unsupervised Learning of Visual Features*. 2018.

[41] Tadeusz Caliński, Jerzy Harabasz. *A dendrite method for cluster analysis*. 1972.

[42] Davies Leonard Davies, Donald Bouldin. *A Cluster Separation Measure*. 1979.

[43] T. Kaizoji1 e D. Sornett. *Market Bubbles and Crashes*. 2017.

[44] Elie Bouri, Soufiane Benbachir, Marwane El Alaoui. *How Bitcoin market trends affect major cryptocurrencies?*. 2025.

[45] Lukáš Pichl e Taisei Kaizoji. *Volatility Analysis of Bitcoin Price Time Series*. 2017.

[46] Caroline Banton. *Understanding Crypto Whales: Impact on Market Liquidity and Price*. 2025.

[47] Martin Harrigan, Lei Shi e Jacob Illum. *Airdrops and Privacy: A Case Study in Cross-blockchain Analysis*. 2018.

[48] Lioba Heimbach e Wenqian Huang. *DeFi Leverage*. 2024.

[49] S&P Dow Jones Indeces. *SPIVA*. 2025.

[50] Tom Celig, Tim Alvaro Ockenga e Detlef Schoder. *Distributional equality in Ethereum? On-chain analysis of Ether supply distribution and supply dynamics*. 2025.

[51] Brad M. Barber, Terrance Odean. *Trading Is Hazardous to Your Wealth: The Common Stock Investment Performance of Individual Investors*. 2000.

[52] De La Cruz, A., A. Medina and Y. Tang. *Owners of the World's Listed Companies*. 2019.