Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Dipartimento di Informatica - Scienze e Ingegneria

SOCIALTRUSTR: SOLUZIONE BLOCKCHAIN PER LA TRACCIABILITÀ E VALIDITÀ DELLA DIFFUSIONE DEI CONTENUTI ONLINE

Laurea triennale in Informatica

Relatore:
Dott. Luca Sciullo

Correlatore:
Dott. Federico Montori

Presentata da:
Arto Manuel

II Sessione

Anno Accademico 2022/2023

The computer was born to solve problems that did not exist before.

Il computer è stato creato per risolvere problemi che prima non esistevano.

Bill Gates

Abstract

Nel contesto attuale del panorama digitale la disinformazione rappresenta una sfida crescente, alimentando la polarizzazione delle opinioni e minando la fiducia nelle istituzioni e nei media. L'enorme flusso di dati scambiati ha reso obsolete le soluzione tradizionali di fact-checking spingendo verso l'adozione di approcci innovativi. Questa tesi si propone di affrontare tale problema presentando SocialTrustr, un sistema basato su blockchain progettato per garantire la tracciabilità e validità delle informazioni condivise all'interno di ambienti social. L'utilizzo di una blockchain fornisce caratteristiche chiave come decentralizzazione e immutabilità dei dati, inoltre il sistema permette di valutare i contenuti tramite l'utilizzo di un digital token e di un meccanismo ad incentivi. Queste caratteristiche consentono a ogni utente di trasformarsi in un attivo partecipante del processo di verifica delle informazioni. È stato quindi progettato un sistema di voting decentralizzato in grado di resistere alle manipolazione grazie a un meccanismo di consenso basato sull'entropia per generare ricompense e penalizzazioni incoraggiando comportamenti onesti. L'implementazione pratica ha confermato la resistenza agli attacchi e l'efficacia del meccanismo di incentivazione. SocialTrustr emerge come una solida risposta alle sfide della validazione delle informazioni in ambienti social.

Indice

1 Introduzione **1**

2 Stato dell'arte **3**
2.1 Problema della diffusione dei contenuti 3
2.2 Soluzione esistenti 4
2.3 Background 8
2.3.1 DLT e Blockchain 8
2.3.2 The Graph 11
2.3.3 IPFS 12

3 SocialTrustr **13**
3.1 Storage e ottenimento dei dati 15
3.2 Sistema di Voting 17
3.2.1 Meccanismo di Consenso 18
3.2.2 Valutazione e Affidabilità 22
3.3 Modello predittivo probabilistico 23
3.3.1 Validità di un contenuto 24
3.3.2 Meccanismo di ricompense e penalizzazioni . . . 26

4 Implementazione **31**
4.1 Blockchain Application 32
4.1.1 ContentSharing 32
4.1.2 ContentEvaluation 33
4.1.3 TrustToken 37

4.2 Subgraph Application 39

5 Validazione **45**
5.1 Integrazione su dApp 45
5.1.1 Livechat Integration 45
5.2 Analisi 48
5.2.1 Resistenza agli attacchi di manipolazione del sistema 49
5.2.2 Analisi del meccanismo di tuning tramite entropia 49
5.2.3 Casi d'uso 57

6 Conclusioni e sviluppi futuri **59**

Riferimenti bibliografici **61**

Capitolo 1

Introduzione

Nel panorama digitale odierno, caratterizzato dalla rapida evoluzione delle tecnologie e dalla pervasività dei media digitali, la disinformazione si presenta come un fenomeno sempre più complesso e diffuso[13]. L'enorme flusso di dati scambiati ha reso la disinformazione una delle sfide più insidiose del nostro tempo. Questo fenomeno non solo alimenta la polarizzazione delle opinioni, ma mina anche la fiducia delle persone nei confronti dei media e delle istituzioni.

La ricerca di strumenti efficaci per tracciare e validare le informazioni risulta quindi urgente, mettendo alla prova le soluzioni convenzionali e spingendo verso l'adozione di nuovi approcci innovativi. Le soluzioni convenzionali si appoggiano spesso all'intervento di esperti o agenzie di fact-checking[31]. Tuttavia tali approcci mostrano limiti evidenti, soprattutto a causa della loro natura centralizzata e esposizione a bias. In aggiunta, la loro lentezza e la scarsa scalabilità di fronte al crescente volume di dati rappresentano ulteriori sfide.

In questo contesto, la tecnologia blockchain emerge come una soluzione promettente per riscrivere le dinamiche della fiducia online. Le caratteristiche di decentralizzazione e immutabilità dei dati offerte dalla blockchain presentano un potenziale rivoluzionario nel contesto della verifica delle informazioni online. Integrare questi principi nella lotta

contro la disinformazione potrebbe non solo migliorare l'efficacia del fact-checking ma anche coinvolgere attivamente la comunità online nella validazione delle informazioni.

In questo lavoro di tesi si propone `SocialTrustr`, un sistema basato su blockchain progettato per affrontare la disinformazione tramite un approccio che sfrutta la teoria dei giochi [18] e sistemi di reputazione decentralizzati. Valorizzando la saggezza della folla [26], SocialTrustr punta a fornire una piattaforma in cui ogni utente contribuisce attivamente alla validazione dei contenuti, essendo incentivato a comportamenti onesti da un meccanismo di consenso basato su ricompense.

Verrà dimostrato come l'approccio decentralizzato e il sistema di incentivi basati sull'entropia, possano costituire un'alternativa valida alle pratiche di fact-checking tradizionali, trasformando ogni utente in un membro attivo della validità dei contenuti online. Il lavoro esplora una prospettiva teorica ben fondata e allo stesso tempo mostra l'applicabilità pratica di SocialTrustr attraverso l'analisi di casi d'uso e lo sviluppo concreto del sistema.

Il documento è strutturato come segue. Nel `capitolo 2` analizzeremo lo stato dell'arte e il background delle tecnologie impiegate nel sistema. Successivamente nel `capitolo 3` verrà presentata la nostra soluzione riguardo alla tracciabilità e validità dei contenuti condivisi in ambienti digitali. Nel `capitolo 4` descriveremo i dettagli relativi all'implementazione, seguiti dal `capitolo 5` in cui verrà analizzato un esempio di caso d'uso del sistema, integrandolo su una app social. Sempre nello stesso capitolo, verranno fornite delle analisi sulla resistenza agli attacchi e sull'efficacia del meccanismo ad incentivi. Infine, nel `capitolo 6` riassumeremo il lavoro svolto analizzando gli obiettivi raggiunti e gli eventuali sviluppi futuri.

Capitolo 2

Stato dell'arte

Nella prima parte di questo capitolo andremo a esplorare lo stato dell'arte per comprendere i vari problemi relativi alla diffusione dei contenuti online e le attuali sfide relative allo sviluppo di soluzioni blockchain per la tracciabilità e validità di tali contenuti.

2.1 Problema della diffusione dei contenuti

Nell'era digitale odierna, la diffusione dei contenuti online è diventata una sfida significativa. La crescente quantità di informazioni generate ogni giorno, unita alla varietà di piattaforme digitali, presenta problemi unici legati all'accessibilità e alla credibilità dei contenuti. Gli utenti vengono inondati quotidianamente con centinaia di contenuti che spesso possono rilevarsi disinformativi o non conformi alla linee guida di una piattaforma[28;25]. La necessità di dare uno strumento direttamente agli utenti per permette di tracciare e limitare la diffusioni di tali contenuti risulta perciò urgente.

Uno dei problemi più rilevanti è sicuramente il fenomeno delle fake news[29]. La velocità di propagazione e la difficoltà nel discernere la veridicità delle notizie contribuiscono a rendere il panorama delle informazioni online un terreno fertile per la diffusione di contenuti inaccu-

rati e fuorvianti. La mancanza di controlli rigorosi e la tendenza delle persone a condividere rapidamente senza verificare contribuiscono ulteriormente all'ampia diffusione di informazioni non verificate.[16] Inoltre l'arrivo di tecnologie avanzate come Natural Language Processing (NLP) e modelli AI come Generative Adversarial Networks (GANs) hanno contribuito allo sviluppo di strumenti in grado di manipolare foto e video in modo irriconoscibile all'occhio umano (i.e. deepfakes).[11] Motivo in più per cui un sistema in grado di garantire l'affidabilità di un contenuto all'interno di una piattaforma risulta necessario al più presto.

Una soluzione efficace deve necessariamente incorporare un approccio decentralizzato e resiliente agli attacchi. Attribuire la fiducia delle informazioni ad entità centralizzate può risultare una soluzione errata, poiché le organizzazioni possono avere difficoltà a garantire la trasparenza e prevenire le manipolazioni, come il cherry-picking a favore di un particolare partito[1;7;24] . Inoltre, il "fact-checking" richiede molto lavoro e le organizzazioni potrebbero non essere in grado di tenere il passo con la quantità di notizie diffuse o di coprire l'intero ecosistema dei contenuti online. Quando un revisore umano controlla un contenuto, questo si è già propagato a numerose persone[27]. Un altro articolo[10] esamina i recenti sviluppi del fact-checking automatizzato per accelerare il processo. La soluzione è quindi affidarsi ad una valutazione che emerga dalla consapevole partecipazione degli utenti.

2.2 Soluzione esistenti

Numerose proposte sono state avanzate per affrontare nello specifico il problema delle fake news. Alcune si concentrano sulla verifica automatica delle informazioni attraverso l'uso di algoritmi e intelligenza artificiale, mentre altre cercano di coinvolgere direttamente la comunità degli utenti nella validazione delle notizie. Tuttavia, molte di queste soluzioni presentano sfide significative legate alla centralizza-

zione delle autorità di verifica, alla resistenza agli attacchi malevoli e alla necessità di un consenso generalizzato sulla veridicità delle notizie.

Diverse soluzioni propongono l'uso combinato di intelligenza collettiva e classificatori comportamentali per valutare la veridicità delle notizie. Un approccio che ha suscitato interesse è quello basato su blockchain, noto per la sua trasparenza e immutabilità. Tuttavia, molte di queste soluzioni dipendono ancora da una certa centralizzazione nella selezione dei validatori o nella gestione delle informazioni.

In un studio condotto da Paula et al.[9] , è stato analizzato l'utilizzo di soluzione come DLT e blockchain per combattere le disinformazioni digitale. Queste tecnologie garantiscono la provenienza e tracciabilità dei dati fornendo un registro di transazioni immutabile, verificabile e trasparente, creando al contempo una piattaforma sicura peer-to-peer per l'archiviazione e lo scambio di informazioni. Tali caratteristiche (illustrate nella Figura 2.1), insieme all'utilizzo di smart contract, possono svolgere un ruolo efficace nella lotta alle fake news.

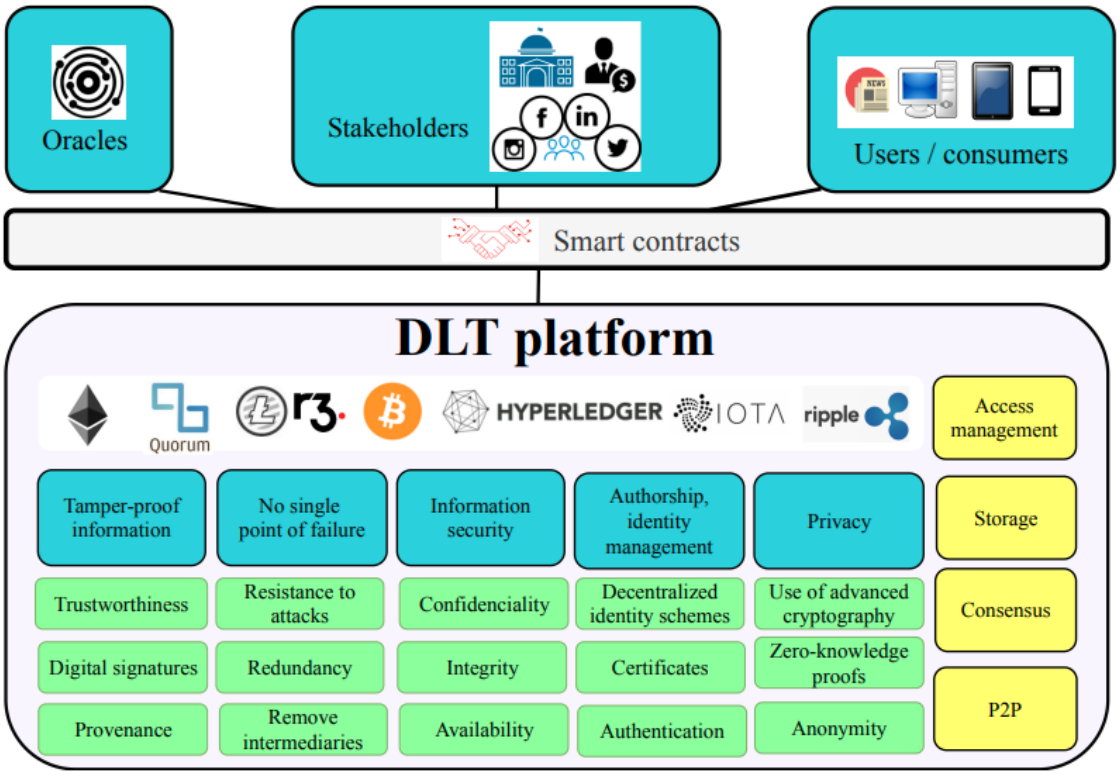

Figura 2.1: Funzionalità DLT e blockchain per combattere la disinformazione digitale [9]

Un approccio interessante è quello proposto di seguito [34], che utilizza un meccanismo di crowdsourcing su blockchain insieme a un sistema di classificazione comportamentale per identificare notizie veritiere o false. Tuttavia, la presenza di un sistema basato su machine learning potrebbe non essere completamente immune da manipolazioni e potrebbe introdurre delle aleatorietà nelle decisioni che svierebbero dal concetto di output deterministico di un sistema blockchain.

Jaroucheh et al. [14] hanno proposto *TRUSTD*: un framework software concettuale che adotta la blockchain e la firma digitale per la verifica dei contenuti online che prevedono l'intervento umano nella revisione delle fake news. Con l'aiuto della firma digitale e della tecnologia blockchain, il loro framework calcola il trust score dei contenuti online in base a un elenco di valutatori che un utente stabilisce e al rispettivo

livello di fiducia nei valutatori. Tuttavia, il trust score di una notizia è generato dall'impostazione dei livelli di fiducia tra utenti e valutatori. Questo approccio manca di oggettività e il trust score perde la sua corrispondenza con l'autenticità della notizia. Ad esempio, il trust score di una stessa notizia verificata dallo stesso gruppo di valutatori porterà a trust scores diversi per utenti che associano livelli di fiducia diversi ai valutatori.

Partendo da *TRUSTD* [14], Chen et al. [6] propone un sistema per avere un valore univoco dell'affidabilità di un determinato valutatore. Implementa inoltre un sistema di prevenzione delle fake news tramite un meccanismo ad incentivi basato sull'entropia dato un pool di voti di utenti fidati. Il suo esperimento si concentra principalmente sull'implementazione pratica e sulle performance dei tempi medi di risposta del sistema, piuttosto che su come e con che efficienza il sistema identifichi le fake news. Inoltre presenta un grosso problema per cui chiunque può diventare un utente fidato semplicemente acquistando dei digital token, e il proprio trust score sarà deciso dalla quantità di questi ultimi rispetto al totale. Questa soluzione può portare ad una maggiore centralizzazione del potere decisionale, in cui chi ha più token ha più influenza.

Un'altra soluzione è quella di *ProBlock* [22], che affronta il problema presentando un modello dinamico con un sistema di voto sicuro basato su blockchain. Utilizza un modello matematico probabilistico per prevedere la veridicità delle notizie in base ai feedback dei revisori. La soluzione si distingue per l'implementazione di una blockchain privata che assicura la privacy dei voti dei revisori. I risultati si mostrano efficaci ma la privatizzazione della blockchain e la scelta privata dei revisori non si integra con il concetto di *open-access* di un sistema blockchain.

Tutte le soluzioni citate finora non fanno riferimento a una modalità di tracciamento delle notizie e della loro diffusione, ma analizzano

solamente la parte relativa alla validazione.

Una soluzione in merito alla tracciabilità delle notizie è quella di Wang et al.[32] dove affrontano la sfida di identificare e tracciare fake news da fonti sconosciute, proponendo un modello di archiviazione sicura dei dati basato sulla blockchain. Utilizza un modello di archiviazione collaborativo on-chain e off-chain per migliorare la tracciabilità dei dati e un algoritmo per garantire la coerenza tra i dati off-chain e quelli del database blockchain. L'approccio collaborativo tra on-chain e off-chain, insieme alla natura decentralizzata e a prova di manomissione della blockchain, mira a garantire la massima sicurezza nell'archiviazione dei dati delle notizie.

Questa analisi critica del panorama attuale sottolinea la necessità di una soluzione avanzata come SocialTrustr, che cerca di combinare l'affidabilità della blockchain con un sistema di valutazione distribuito per affrontare in modo più completo le sfide delle fake news. Nel capitolo successivo, esploreremo in dettaglio la proposta di SocialTrustr, mettendo in luce come affronta e supera le limitazioni delle soluzioni esistenti.

2.3 Background

Questa sezione contiene le conoscenze di base che sono ampiamente utilizzate in tutta la tesi. Vedremo le caratteristiche principali di un sistema basato su blockchain insieme ad alcuni dei suoi concetti fondamentali e alcuni degli strumenti di sviluppo utilizzati per l'implementazione. Infine introdurremo due applicazioni distribuite utilizzate per lo storage e presentazione dei dati.

2.3.1 DLT e Blockchain

Le tecnologie Distributed Ledger (DLT) sono sistemi basati su un registro distribuito, ossia sistemi in cui tutti i nodi di una rete possiedono

la medesima copia di un database che può essere letto e modificato in modo indipendente dai singoli nodi. Blockchain[4] è un tipo di DLT in cui il registro è strutturato come una catena di blocchi contenenti più transazioni e i blocchi sono tra di loro concatenati tramite una firma crittografia chiamata hash. Il registro distribuito peer-to-peer è gestito da una rete di nodi partecipanti che effettuano transazioni senza affidarsi a intermediari terzi. È immutabile, decentralizzata, tracciabile e fornisce un'elevata sicurezza alle transazioni e alle informazioni che detiene. I blocchi sono composti da un body e da un header, un hash del blocco precedente, un timestamp, un nonce, un merkle root e altre informazioni presenti nel body di un blocco. Ogni blocco è collegato a quello precedente tramite l'hash contenuto nell'header. Le blockchain sono classificate in tre gruppi in base alle applicazioni che servono[19]. Questi gruppi sono:

- Public Blockchain (accesso illimitato ai dati, pubblica, chiunque può accedervi)
- Private Blockchain (controllata da un'autorità, solo i membri di una particolare organizzazione possono accedervi)
- Consortium Blockchain (solo le organizzazioni all'interno del sistema possono scrivere, leggere, inviare e registrare congiuntamente i dati delle transazioni in una blockchain amministrata da diverse organizzazioni, dove ogni organizzazione può avere più nodi).

Meccanismi di Consenso

I meccanismi di consenso sono l'insieme completo di idee, protocolli e incentivi che consentono a una serie distribuita di nodi di acconsentire sullo stato di una blockchain e sulle validità delle sue transazioni. Un meccanismo di consenso si divide in due protocolli: **chain selection** (*i.e. Longest Chain Rule utilizzata da Bitcoin*) e **Sybil resistence** (*i.e. Proof of Work, Proof of Stake*). Il Sybil attack riguarda l'abilità

di un singolo utente di controllare un numero sproporzionato di nodi fittizi in un sistema per violare la blockchain. I meccanismi di Sybil-resistance mirano a prevenire o limitare tale attacco.

Ethereum e Smart Contract

Ethereum è una piattaforma blockchain sviluppata per consentire la creazione e l'esecuzione di applicazioni decentralizzate. Queste dApps sono costruite utilizzando gli Smart Contract, ovvero contratti in cui i termini dell'accordo tra acquirente e venditore sono scritti direttamente in righe di codice che vengono eseguite al determinarsi di specifiche condizioni. L'uso degli smart contract consente l'automazione di processi complessi e può contribuire a ridurre i costi e i rischi associati ai contratti tradizionali.

Solidity

Solidity è il linguaggio di programmazione orientato allo sviluppo di smart contract ed è comunemente utilizzato per sviluppare applicazioni su piattaforme blockchain come Ethereum. È un linguaggio di alto livello progettato per essere facile da scrivere e compilare fino al low-level code che può essere eseguito sulla macchina virtuale di Ethereum (EVM). Solidity consente agli sviluppatori di creare contratti intelligenti funzionali e sicuri.

ERC20 Token

Gli ERC-20 sono standard tecnici utilizzati per implementare digital token su blockchain Ethereum. Questi standard definiscono le funzioni di base che un token deve supportare, consentendo l'interoperabilità tra diverse applicazioni e servizi che utilizzano token.

Foundry

Foundry è un framework per lo sviluppo di applicazioni blockchain che semplifica la creazione e la gestione di contratti intelligenti. Fornisce strumenti per la compilazione, il debug e il deploy di contratti su varie reti blockchain.[8]

2.3.2 The Graph

The Graph è un protocollo decentralizzato per l'indicizzazione e l'ottenimento di dati blockchain. Offre un modo efficiente per recuperare dati specifici dalla blockchain. Questo protocollo semplifica notevolmente l'accesso ai dati per le applicazioni decentralizzate e permette di effettuare delle query su dati difficili da ottenere direttamente.[2]

Subgraph

Un subgraph è un'API personalizzata costruita sui dati della blockchain. I subgraph vengono interrogati utilizzando il linguaggio di interrogazione GraphQL e vengono distribuiti a un Graph Node utilizzando la Graph CLI. Una volta distribuiti e pubblicati sulla rete decentralizzata di The Graph, gli indicizzatori elaborano i subgraph e li rendono disponibili per essere interrogati dai consumatori di subgraph[2].

GraphQL

GraphQL è un linguaggio di interrogazioni e manipolazione dei dati open-source per API che consente di specificare la struttura dei dati richiesti. Rispetto alle API REST tradizionali, offre un controllo più granulare sui dati restituiti, evitando sovraccarichi inutili.[20]

2.3.3 IPFS

InterPlanetary File System (IPFS) è un insieme di protocolli peer-to-peer per l'indirizzamento, l'instradamento e il trasferimento di dati indicizzati per contenuto in un file system distribuito. Utilizza un sistema di hash crittografico per identificare i contenuti, rendendo la memorizzazione e il recupero dei dati più sicuri e efficienti. [3]

Capitolo 3

SocialTrustr

La nostra soluzione fornisce un sistema basato su blockchain finalizzato a fornire la tracciabilità e validità dei contenuti condivisi all'interno di ambienti social. Permette inoltre di valutare i contenuti tramite l'utilizzo di token e di un meccanismo ad incentivi per ridurre la diffusione delle fake news.

Il sistema potrà essere integrato su qualunque applicazione social che gestisca degli utenti con wallet e l'invio/inoltro di contenuti digitali. Si propone come una libreria a tutti gli effetti che sviluppatori possono integrare all'interno delle loro app, inoltre risulta agnostico sul tipo di contenuto delle informazioni scambiate.

SocialTrustr è progettato per incentivare l'onestà online attraverso la retribuzione in token agli utenti che pubblicano contenuti onesti e valutano in modo onesto. Al contrario, comportamenti disonesti comportano la perdita di token e affidabilità.

Per prevenire manipolazioni, SocialTrustr utilizza un digital token e meccanismi di sybil-resistance. I dati acquisiti includono informazioni sui contenuti e sulla loro diffusione, nonché le valutazioni degli utenti. La validità di un contenuto è calcolata mediante un'analisi probabilistica.

Le motivazioni dietro l'implementazione di SocialTrustr risiedono nel monitorare la diffusione dei contenuti, prevenire la diffusione di fake news, tracciare l'affidabilità degli utenti e garantire l'immutabilità e trasparenza dei dati attraverso l'utilizzo della blockchain.

Inoltre SocialTrustr mette il potere nelle mani degli utenti, eliminando la necessità di affidarsi ad un'autorità centrale per certificare la validità di un contenuto. Infatti un concetto fondamentale analizzato per lo sviluppo del sistema è quello del *Widsom of the crowd*[26], teoria sociologica e statistica secondo la quale la media delle valutazioni date da una massa di individui inesperti indipendenti sarebbe in grado di fornire una risposta almeno eguale, o anche superiore, a quella di un qualsiasi esperto. Questa teoria trova un'applicazione intensa su Internet, per esempio `Stack Exchange`, `Reddit`, `Wikipedia` e moltri altri si fondano su di essa e puntano sul contenuto generato dagli utenti ovvero sulla cosiddetta intelligenza collettiva.

Il sistema si distingue in tre componenti principali:

1. Smart Contract e applicazioni distribuite, per lo storage e l'ottenimento dei dati immutabili;
2. Sistema di voting, basato su un digital token e un meccanismo di consenso per incentivare ad agire onestamente;
3. Modello predittivo probabilistico, per determinare la validità dei contenuti e effettuare il tuning dei token e dell'affidabilità degli utenti

Nota

È interessante notare come il sistema, pur essendo originariamente concepito per monitorare e validare notizie al fine di prevenire la diffusione di fake news, si sia evoluto in una piattaforma estremamente flessibile e adattabile. La completa astrazione del sistema riguardo al contenuto consente agli utenti di utilizzarlo in svariati contesti oltre

alla verifica delle notizie. Ad esempio, gli utenti possono sfruttare la piattaforma per segnalare contenuti che non rispettano le linee guida stabilite o esprimere opinioni su argomenti rilevanti alla comunità, quali sondaggi o discussioni tematiche. La decentralizzazione del potere decisionale consente agli utenti di contribuire alla definizione delle regole e delle dinamiche della comunità, creando un ambiente digitale che riflette le preferenze e i valori condivisi

3.1 Storage e ottenimento dei dati

Questo primo componente di SocialTrustr ha come obiettivo quello di fornire un meccanismo per tenere traccia della creazione e della condivisione dei contenuti online, insieme alle relative votazioni, e di gestire la quantità di token e affidabilità attribuita a ogni utente.

Le funzionalità del componente sono qui riassunte:

1. Salvataggio di contenuti creati o inoltrati;
2. Ottenere una gerarchia ad albero relativa alla condivisione di un contenuto;
3. Salvataggio delle valutazioni;
4. Gestione dell'affidabilità e della quantità di token di un utente;
5. Fornire un meccanismo di non ripudiabilità dei contenuti salvati;

Salvare i dati su blockchain permette una registrazione sicura garantendo immutabilità e trasparenza, attributi che in un sistema di validazione risultano essenziali. Queste funzionalità portano però un costo, infatti salvare troppi dati su blockchain può risultare critico in termini di efficienza e scalabilità.

Proprio per questo sono nate soluzioni come The Graph[2] che permettono di indicizzare i dati su blockchain creando delle entità partendo da eventi emessi dagli smart contract, fornendo poi una modalità di

accesso ai dati tramite GraphQL[20]. Questa soluzione ci permette quindi di spostare la logica della diffusione di un contenuto su The-Graph, creando delle entità analoghe a quello on-chain aggiungendo una reference all'id del contenuto padre e una lista di reference agli id dei contenuti figli inoltrati.

Un altro problema affrontato riguarda il salvataggio del contenuto effettivo. Per gli stessi motivi elencati precedentemente il salvataggio di dati on-chain risulta dispendioso, bisogna perciò optare per un doppio sistema di storage salvando il contenuto su IPFS e salvando on-chain soltanto il relativo cid (*Content Identifier*)[3].

Una funzionalità aggiuntiva data dalla gestione del contenuto su IPFS riguarda la non ripudiabilità di quest'ultimo. Il *Content Identifier* di un dato caricato su IPFS è generato combinando l'hash del dato e informazioni codec di esso, quindi al cambiare del dato cambierà anche il cid. Di conseguenza una volta creata una transazione, quel ipfsCid farà sempre riferimento allo stesso contenuto e il condivisore sarà colui che ha effettua la transazione.

Alla creazione di un nuovo contenuto sarà necessario specificare i seguenti dati: *title, ipfsCid, chatName* e *parentId*. ParentId risulta uguale a 0 se si tratta di un nuovo contenuto, altrimenti sarà uguale all'id del contenuto padre che si sta condividendo. Dato aggiuntivo che verrà salvato on-chain è il condivisore del contenuto, ovvero l'address di colui che sta effetuando la transazione. Nella figura 3.1 è possibile vedere il flusso di storage e salvataggio dei dati.

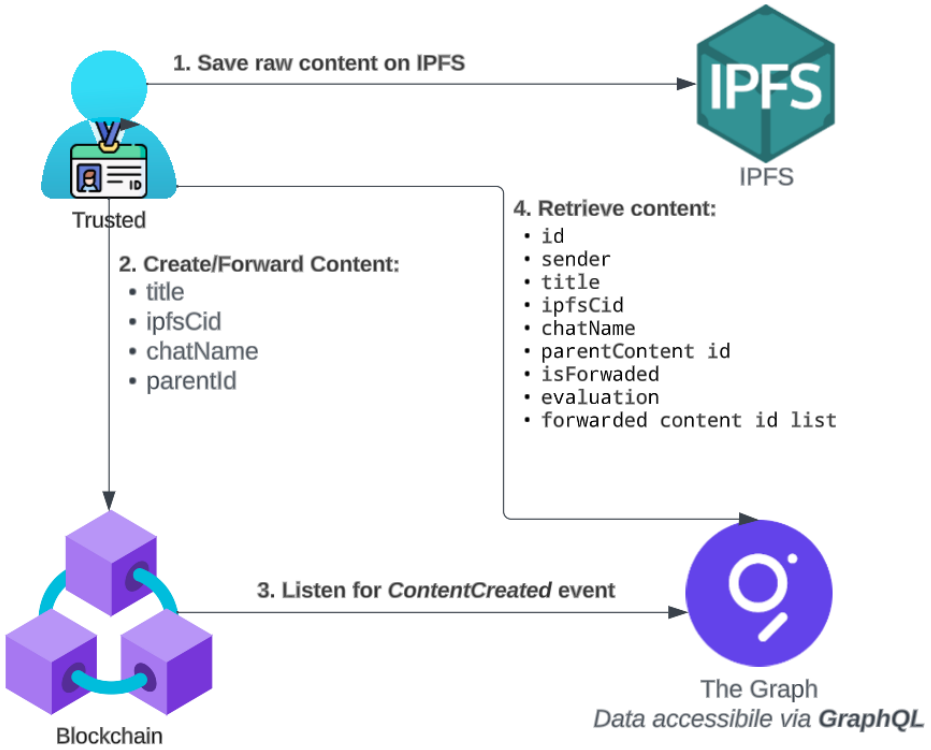

Figura 3.1: Flusso storage e ottenimento dati

3.2 Sistema di Voting

Il sistema di voting ha come obiettivo principale quello di ottenere un consenso distribuito sulla validità di un contenuto decentralizzando il potere decisionale. Permette inoltre di limitare le manipolazioni del sistema tramite un meccanismo di consenso che incentiva a votare onestamente e a condividere contenuti validi.

Il consenso distribuito è il fondamento su cui si basa l'integrità e la validità delle transazioni all'interno di una rete decentralizzata[4]. Nel caso specifico di SocialTrustr, ci troviamo di fronte a una sfida unica: garantire la validità dei contenuti all'interno di ambienti social, con un'enfasi particolare sulla lotta alla disinformazione.

Le sfide che emergono sono intrinseche alla natura stessa dei conte-

sti sociali online. La manipolazione delle informazioni, la diffusione di notizie false e la creazione di identità fittizie sono minacce costanti. Creare un meccanismo di consenso in grado di affrontare questi problemi richiede un approccio attento e innovativo.

3.2.1 Meccanismo di Consenso

La creazione di un meccanismo di consenso per la validazione dei contenuti all'interno di SocialTrustr rappresenta una sfida significativa nel contesto della blockchain. Il termine *consenso* denota l'accordo distribuito tra partecipanti sulla validità di un determinato contenuto. Nell'ambito della blockchain, ciò implica la necessità di decentralizzare il potere decisionale per evitare influenze centralizzate o manipolazioni.

Le sfide principali emergono dalla necessità di garantire che gli attori del sistema partecipino in modo onesto e che la validazione sia resistente a attacchi malevoli. In particolare, la manipolazione del consenso, ad esempio attraverso la creazione di identità false, rappresenta una minaccia significativa. Per affrontare questo problema, è necessario implementare un meccanismo di sybil-resistance.

La sybil-resistance rappresenta un concetto cruciale per garantire l'integrità del meccanismo di consenso. In un contesto di validazione dei contenuti, la sybil-resistance implica la resistenza alla creazione di identità fittizie per influenzare il processo decisionale. Partendo dal lavoro svolto di Chen et al.[6]m per affrontare questa sfida, SocialTrustr introduce il concetto di *Badge* e utenti *Trusted*.

Token come deterrente e fattore economico

Basandoci sul concetto di proof of stake[15], chiunque voglia diventare un utente Trusted deve prima acquistare una certa quantità di token definita dal sistema. Questa quantità definisce il Badge di un utente, una sorta di attestazione di fiducia che permette agli utenti di partecipare al processo di condivisione e validazione dei contenuti. Questo

requisito limita gli attacchi di tipo sybil, poiché la creazione di identità multiple richiede l'acquisizione di Badge multipli, comportando un costo significativo e agendo così da deterrente.

All'interno del sistema di votazione di SocialTrustr, il concetto di staking dei token gioca un ruolo chiave. Gli utenti devono impegnare una quantità di token per partecipare al processo di condivisione e valutazione dei contenuti. Questo meccanismo agisce come deterrente, poiché chiunque voglia partecipare deve sacrificare una risorsa economica.

Il processo di staking dei token svolge un ruolo cruciale nel garantire che gli utenti che partecipano al sistema abbiano un interesse economico nel comportarsi onestamente. La perdita di token a causa di comportamenti disonesti agisce come un disincentivo economico, al contrario la possibilità di guadagnare token attraverso partecipazione onesta costituisce un incentivo.

Incentivi Economici nel Meccanismo di Consenso

L'introduzione di incentivi economici nel meccanismo di consenso di SocialTrustr è fondamentale per garantire la partecipazione onesta e la validazione accurata dei contenuti. Gli utenti sono motivati economicamente a condividere e valutare contenuti in modo onesto per evitare perdite di token e, al contrario, ricevere ricompense in token per comportamenti onesti e contributi validi al sistema.

La combinazione di sybil-resistance attraverso l'acquisizione del Badge, il meccanismo di staking dei token come deterrente e gli incentivi economici per la partecipazione onesta costituisce la struttura robusta del meccanismo di consenso di SocialTrustr. Questa combinazione mira a creare un ecosistema in cui la validazione dei contenuti è affidabile, resistente alle manipolazioni e incentiva comportamenti onesti da parte degli utenti.

Flusso di acquisto e vendita di token TRS

Il sistema è stato progettato per l'integrazione su blockchain ethereum-based. Data la variabilità del prezzo di ETH si ha la necessità di realizzare una stablecoin che mantenga il suo prezzo nel tempo per evitare una inflazione del valore. È nata così la moneta del sistema: Trust Token (TRS).

I TRS in SocialTrustr sono quindi una valuta con un valore economico costante definito in dollari o euro. Questo conferisce loro una concretezza che va al di là del mondo digitale, collegando l'esperienza degli utenti alla dimensione tangibile della moneta.

Come parte del protocollo ERC20, i TRS in SocialTrustr possono essere scambiati e venduti tra gli utenti. Gli utenti hanno la libertà di gestire i propri token in modo dinamico, secondo le loro esigenze e preferenze.

Immagina un nuovo utente che decide di entrare attivamente nel mondo di SocialTrustr. Il primo passo è l'acquisto del badge, il cui costo è definito dalla quantità predefinita di TRS per diventare un utente Trusted. Questa quantità viene prelevata dal saldo del sistema e, nel caso in cui il saldo non sia sufficiente, il sistema interviene con il minting, generando i token necessari. Il flusso di acquisto è visibile nell'immagine 3.2. Il minting di nuovi token è riservato esclusivamente agli utenti che non hanno ancora raggiunto lo status di Trusted.

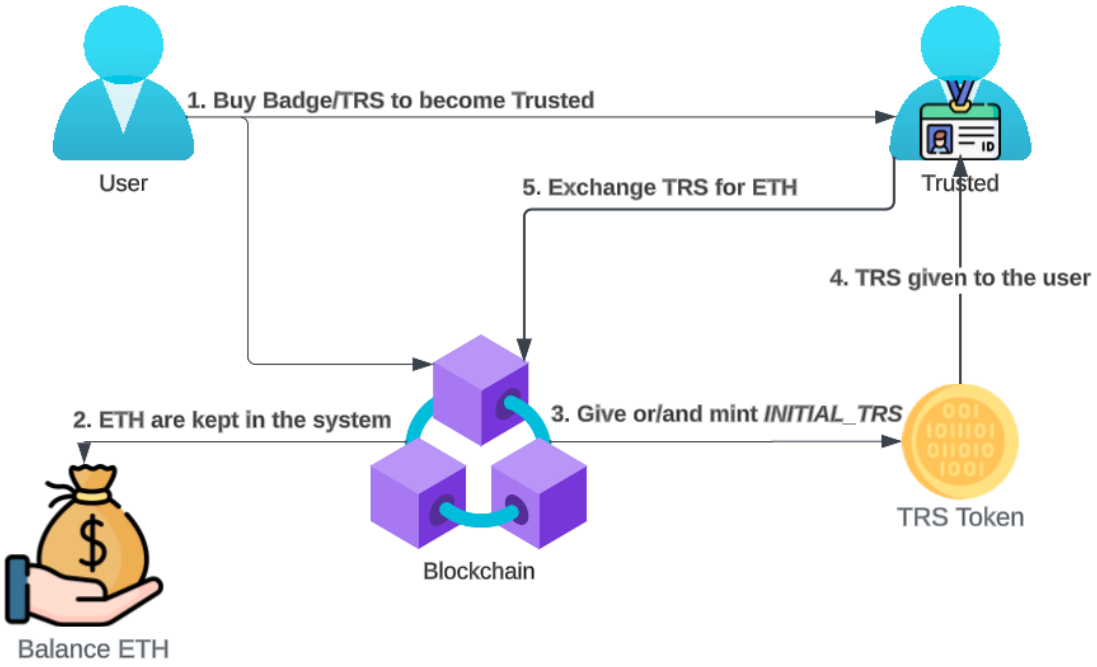

Figura 3.2: Acquisto Badge/TRS

Supponiamo che un utente, nel corso del tempo, desideri disinvestire dai suoi token. In SocialTrustr, questa azione è possibile attraverso la vendita dei token al sistema. Il valore economico dei token determina il ritorno in dollari o euro che l'utente riceverà in cambio. Questo processo fornisce una via di uscita flessibile per gli utenti che vogliono gestire il proprio portafoglio di token.

Per mantenere un certo grado di controllo e fiducia nel sistema, l'acquisto di token direttamente dal saldo del sistema è riservato agli utenti che hanno raggiunto lo status di Trusted. Questa restrizione garantisce che solo coloro in possesso di un badge e quindi anche di una quantità di TRS possano contribuire direttamente allo scambio di token e denaro.

3.2.2 Valutazione e Affidabilità

Dopo aver esaminato il meccanismo di consenso di SocialTrustr, è essenziale comprendere come gli utenti valutano e contribuiscono al sistema, evidenziando la centralità della fiducia e dell'affidabilità.

Solo gli utenti Trusted hanno il privilegio di condividere contenuti nel sistema. La valutazione non è istantanea, ma avviene nel corso di un periodo di tempo prestabilito. Durante questo intervallo, gli altri utenti Trusted possono esaminare il contenuto, basare la propria valutazione sull'affidabilità dell'utente che l'ha condivisa e attribuire un *Confidence Score*, espresso in percentuale, che determina l'impatto della loro decisione sulla valutazione totale.

Introdurre il concetto di affidabilità come valore separato dai token TRS è cruciale. L'affidabilità, rappresentata da un valore numerico in un range da 0 a 100 (indicato come AF), riflette la credibilità di un utente nel sistema. Un nuovo utente entra con un AF neutrale di 50, con il grado di affidabilità che aumenta man mano che dimostra onestà e partecipazione costruttiva.

La separazione di affidabilità e token è una scelta deliberata. Mentre i token TRS agiscono come deterrente e fattore economico nel sistema, l'affidabilità è un indicatore della coerenza e dell'onestà di un utente nel tempo. Questo approccio mira a evitare una centralizzazione del potere decisionale basata esclusivamente sulla quantità di token detenuti, che potrebbe minare la democratizzazione del processo di consenso.

SocialTrustr è progettato per essere un ecosistema in cui la fiducia, l'affidabilità e l'equità svolgono un ruolo centrale. Gli utenti Trusted, attraverso la combinazione di token e affidabilità, partecipano attivamente alla validazione dei contenuti, garantendo un meccanismo di consenso resistente e decentralizzato.

3.3 Modello predittivo probabilistico

Il componente relativo allo sviluppo di un modello predittivo probabilistico all'interno di SocialTrustr si propone di raggiungere due obiettivi principali:

1. **Determinare la validità di un contenuto**: L'obiettivo primario è sviluppare un algoritmo capace di determinare la validità di un contenuto, basandosi su un pool di valutazioni fornite dagli utenti Trusted. Questo processo è fondamentale per garantire che solo contenuti accurati e affidabili siano riconosciuti come tali all'interno del sistema.
2. **Definire un algoritmo per la redistribuzione dei token e il tuning dell'affidabilità degli utenti**: Il secondo obiettivo è la creazione di un algoritmo concreto per gestire la redistribuzione dei token tra gli utenti e il tuning dell'affidabilità. Questo meccanismo è essenziale per mantenere un ecosistema dinamico, dove la partecipazione onesta e le valutazioni accurate sono premiate, mentre comportamenti disonesti sono sanzionati.

Determinare la validità di un contenuto all'interno di un contesto social presenta diverse sfide. La varietà di opinioni e la possibilità di valutazioni di parte richiedono un approccio bilanciato. Il modello predittivo deve essere in grado di gestire questa variabilità e cercare di giungere a una conclusione oggettiva sulla validità di un contenuto.

Inoltre, la necessità di ridurre il rischio di manipolazioni e valutazioni distorte richiede un meccanismo di disincentivi robusto. Gli utenti non dovrebbero essere in grado di influenzare il risultato attraverso identità multiple o comportamenti fraudolenti.

Adesso andremo a introdurre nel dettaglio i vari passaggi dell'algoritmo relativo al meccanismo di *Proof of Stake* a incentivi basato sull'entropia, come mostrato in figura 3.3.

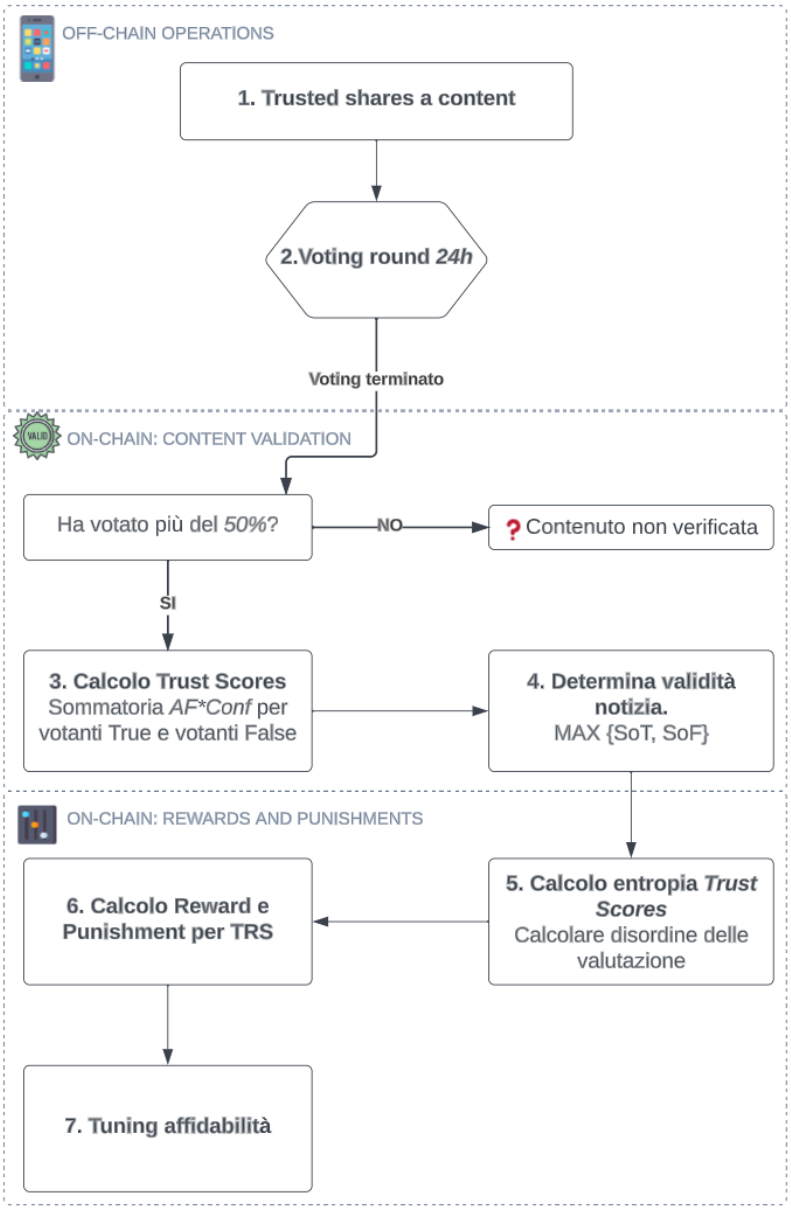

Figura 3.3: Flusso validazione di un contenuto

3.3.1 Validità di un contenuto

La validità di un contenuto è determinata da un processo rigoroso durante il periodo di validazione. Una valutazione è considerata valida solo se almeno il 50% dei votanti partecipa attivamente durante questo

periodo.

1) Condivisione del contenuto:

Un utente Trusted effettua lo stake dei suoi TRS per condividere un contenuto, avviando il periodo di validazione. La quantità di TRS da mettere in gioco è decisa dal sistema.

2) Voting round:

Gli utenti Trusted effettuano lo stake dei loro TRS per valutare il contenuto. Come per la condivisione, la quantità di TRS per la valutazione è decisa dal sistema. Ogni utente assegna un *Confidence Score* al proprio voto, indicando se ritengono il contenuto valido o non valido (*vero o falso*).

3) Calcolo Trust Scores:

Al termine del periodo di validazione si procede con il calcolo dei Trust Scores in base alle affidabilità dei valutatori e relative confidence assegnate. Calcoliamo innanzitutto lo **Score of True (SoT)**, descritto nell'equazione 3.1, che rappresenta la possibilità che un contenuto sia valido. T_T è l'insieme dei valutatori che ha approvato il contenuto.

$$SoT = \sum_{i=1}^{n} AF_i \times Conf_i, \forall i \in T_T \tag{3.1}$$

Successivamente calcoliamo lo **Score of False (SoF)**, descritto nell'equazione 3.2, che rappresenta invece la possibilità che un contenuto non sia valido. T_F è l'insieme dei valutatori che non ha approvato il contenuto.

$$SoF = \sum_{i=1}^{n} AF_i \times Conf_i, \forall i \in T_F \tag{3.2}$$

4) Determina validità contenuto:

Dopo aver calcolato i Trust Scores, si procede con il determinare la validità del contenuto. Ciò avviene confrontano i due Trust Scores: il maggiore dei due determina la validità o non validità di un contenuto, come espresso nell'equazione 3.3.

$$Max\{SoT, SoF\} \tag{3.3}$$

3.3.2 Meccanismo di ricompense e penalizzazioni

Il meccanismo di ricompense e penalizzazioni all'interno del modello predittivo probabilistico di SocialTrustr è fondamentale per garantire l'integrità del sistema e incoraggiare comportamenti onesti. Questo processo intricato coinvolge il calcolo dell'entropia dei Trust Scores, nonché la distribuzione ponderata dei token TRS e la regolazione dell'affidabilità degli utenti.

Per quanto riguarda il condivisore del contenuto, questi verrà considerato nel calcolo di ricompense e penalizzazioni come un utente che ha valutato il contenuto come `True` con 100% confidence.

1) Calcolo entropia dei Trust Scores:

Il calcolo dell'entropia dei Trust Scores svolge un ruolo cruciale nel determinare la quantità di disordine presente nelle valutazioni degli utenti Trusted. L'entropia di Shannon[23], che è definita in 3.4, è una delle metodologie più utilizzate per misurare il disordine[17]. Un alto livello di entropia implica una distribuzione più omogenea dei voti, mentre un'entropia bassa indica un consenso maggiore tra gli utenti. Questa misurazione è preziosa poiché determina l'ampiezza delle ricompense e penalizzazioni. Maggiore è l'entropia, maggiore è l'incertezza e, di conseguenza, minori sono le penalizzazioni e le ricom-

pense per mantenere un equilibrio tra disincentivare manipolazioni e incoraggiare la partecipazione accurata.

$$Entropy(S) = -\sum_{i=1}^{k} p_i \log_k p_i \tag{3.4}$$

In 3.4, definiamo S come un insieme che contiene i risultati delle valutazioni di tutti i valutatori. Abbiamo definito $k = 3$ perché ci saranno tre classi di risultati di valutazione: ***True***, ***False*** e ***Neutro***. Inoltre, p_i rappresenta la proporzione dei risultati di valutazione nella classe i. In particolare, definiamo p_1 come proporzione dei risultati di valutazione pari a ***True*** e p_2 per quelli pari a ***False***. Inoltre, definiamo p_3 come proporzione dei valutatori che esprimono la propria opinione come ***Neutro***. Riassumendo calcoliamo quindi p_1, p_2 e p_3 come descritto in 3.5.

$$\begin{aligned} p_1 &= \frac{\sum_{i \in T_T} Conf_i}{n} \\ p_2 &= \frac{\sum_{i \in T_F} Conf_i}{n} \\ p_3 &= 1 - p_1 - p_2 \end{aligned} \tag{3.5}$$

2) Calcolo Ricompense e Penalizzazioni:

Per i calcoli che seguiranno, analizzeremo il caso in cui $SoT > SoF$. Nel caso contrario il calcolo è analogo.

Per quanto riguarda i token TRS, il meccanismo prevede una redistribuzione ponderata degli staked token. In caso di valutazioni corrette, la perdita di stake degli utenti che hanno valutato in modo errato viene trasferita a coloro che hanno espresso valutazioni accurate, pro-

porzionalmente ai confidence score espressi. Questo processo mira a mantenere un equilibrio dinamico tra gli utenti e promuovere la partecipazione onesta.

In 3.6 calcoliamo la somma di token TRS tolti dagli stake dei T_F.

$$Punishment = \sum_{i \in T_F} -(Stake * Conf_i) \tag{3.6}$$

Successivamente in 3.7 calcoliamo la ricompensa per ogni T_T redistribuendo i token TRS perquisiti in base al rapporto del proprio confidence score sul totale.

$$Reward_i = \frac{Conf_i}{TotSoT_{Conf}} * Punishment \ \forall T_T \tag{3.7}$$

Per regolare l'affidabilità degli utenti, il sistema fa uso dell'entropia che influenza direttamente la severità delle penalizzazioni applicate. Maggiore è l'entropia, minore sarà la penalizzazione, poiché un elevato grado di incertezza suggerisce una distribuzione più casuale delle valutazioni, rendendo inopportuni giudizi punitivi troppo drastici. Questo approccio è fondamentale per adattare il sistema alle dinamiche mutevoli delle valutazioni degli utenti.

Per contribuire a enfatizzare l'importanza della sicurezza delle valutazioni, penalizzazione e ricompense sono entrambe moltiplicate per la confidence assegnata. Questo approccio mira a promuovere la riflessione e differenziare ricompense e penalizzazioni in base alla certezza nell'esprimere una valutazione.

Come mostrato in 3.8, la penalizzazione è strutturata in modo molto più drastico rispetto alle ricompense, al fine di scoraggiare comportamenti non affidabili.

$$Punishment_i = AF_i * Conf_i * (1.0 - Entropy) \ \forall T_F \tag{3.8}$$

Come mostrato in 3.9, la ricompensa è calcolata in modo che sia inversamente proporzionale all'affidabilità, riducendo così la centralizzazione del potere e prevenendo accumuli eccessivi di affidabilità. In altri termini, gli utenti più affidabili vedranno un incremento meno significativo nelle loro ricompense rispetto a quelli meno affidabili, promuovendo così una distribuzione più equa e decentralizzata delle valutazioni. Inoltre, la ricompensa è divisa per una costante **M** (*e.g. 2, 2.5 o 3*), sottolineando l'importanza di mantenere un sistema in cui le penalizzazioni risultino decisamente più drastiche delle ricompense,

$$Reward_i = \frac{(100 - AF_i) * Conf_i * (1.0 - Entropy)}{M} \forall T_T \tag{3.9}$$

In conclusione, questo complesso meccanismo di ricompense e penalizzazioni è progettato per mantenere l'integrità e la robustezza di SocialTrustr, fornendo incentivi per la partecipazione accurata e disincentivando comportamenti fraudolenti attraverso una distribuzione equa e bilanciata delle ricompense e delle penalizzazioni.

Capitolo 4

Implementazione

In questo capitolo andremo ad analizzare l'implementazione pratica di SocialTrustr, esplorando gli smart contract e l'integrazione di The Graph. Mostreremo quali sono state le principali sfide incontrate e come siamo riusciti ad ottenere un sistema funzionante, pronto per l'integrazione in qualunque applicazione social. È possibile testare l'intero sistema anche localmente utilizzando una suite di strumenti dedicati allo sviluppo e al testing degli smart contract

Il repository contenente l'intero codice sviluppato è disponibile al seguente link: `https://github.com/ManuelArto/SocialTrustr`.

Come specificato suddivideremo il lavoro svolto in due componenti:

- **Blockchain Application**: insieme di Smart Contract che permettono di interagire con le funzioni di condivisione e valutazione dei contenuti e di gestione dei token TRS;
- **Subgraph Application**: API personalizzata per l'interrogazione dei contenuti condivisi salvati su blockchain, insieme allora loro tracciabilità e rispettive validazioni finali.

4.1 Blockchain Application

Gli smart contract costituiscono la parte principale del sistema SocialTrustr, in quanto gestiscono la logica di condivisione, valutazione e gestione dei token TRS.

Il processo di sviluppo degli smart contract è stato semplificato e accelerato grazie all'utilizzo di `Foundry`[8], suite di strumenti per lo sviluppo blockchain, che ha semplificato operazioni come la gestione di una blockchain locale, attraverso `Anvil`, lo sviluppo di test, sempre scritti in Solidity, e la creazione di script per interagire con gli smart contract deployati.

Abbiamo deciso di suddividere il sistema in tre smart contract, ognuno rappresentane una specifica funzionalità nel sistema: *ContentSharing*, *ContentEvaluation* e *TrustToken*. Oltre a questi sono stati sviluppate sei librerie relative a: strutture dati, errori, eventi, funzioni di conversione da ETH a USD, funzioni per il calcolo della validità di un contenuto e funzioni per il tuning di token e affidabilità.

4.1.1 ContentSharing

Lo smart contract `ContentSharing` offre le funzionalità di storage dei contenuti condivisi tramite una lista di *Content Struct*, contenente i dati: *title*, *ipfsCid*, *chatName*, *sharere*, *isForwarded* e *timestamp*. Lo smart contract include diverse funzioni per aggiungere e ottenere i contenuti condivisi.

Creazione di un contenuto

Tramite la funzione `createContent`, visibile in figura 4.1, è possibile aggiungere un nuovo contenuto specificando il titolo, l'indirizzo IPFS, la chatName in cui è stato condiviso il contenuto e un valore intero che fa riferimento all'indice nella lista del contenuto padre o `0` se rappresenta un nuovo contenuto. La funzione verifica che l'indice del

contenuto padre sia valido e successivamente effettua lo stake dei TRS necessari per condividere e registra il contenuto su blockchain. Infine emette l'evento `ContentCreated` che verrà indicizzato dal nostro Subgraph come vedremo più avanti.

```
function createContent(
  string calldata title,
  string calldata ipfsCid,
  string calldata chatName,
  uint parentId
) external returns (uint id) {
  if (parentId >= s_content.length) {
    revert Errors.ContentSharing_NoParentContentWithThatId();
  }

  i_trustToken.stakeTRS(msg.sender, i_trustToken.TRS_FOR_SHARING());

  DataTypes.Content memory content = DataTypes.Content(
    title,
    ipfsCid,
    chatName,
    msg.sender,
    parentId != 0,
    block.timestamp
  );
  s_content.push(content);

  id = s_content.length - 1;
  emit Events.ContentCreated(id, msg.sender, title, ipfsCid, chatName, parentId);
}
```

Listing 4.1: ContentSharing.sol::createContent

4.1.2 ContentEvaluation

Lo smart contract `ContentEvaluation` si occupa di gestire le valutazioni degli utenti e di fornire un metodo per la validazione finale di un contenuto.

All'interno del contratto, sono presenti diverse variabili di stato, tra cui un mapping per memorizzare le valutazioni dei contenuti e un mapping per tenere traccia dei voti degli utenti.

La valutazione di un contenuto è definita da una struct contenente: lo stato della valutazione, una lista delle varie valutazioni degli utenti e infine la valutazione finale del contenuto.

Il contratto dispone di una serie di funzioni che consentono agli utenti di valutare i contenuti specificandone l'id, la valutazione (vero o falso) e il confidence score. Ci sono anche funzioni per controllare lo stato di validazione del contenuto e per chiudere la validazione e ottenere una valutazione finale basata sui voti ricevuti. Infine, il contratto fornisce alcune funzioni getter per recuperare lo stato di validazione del contenuto, la valutazione finale e il numero di valutazioni effettuate.

Valutazione di un contenuto

In figura 4.2 è mostrata la funzione `evaluateContent` che permette agli utenti di valutare un contenuto dopo aver effettuato lo stake dei loro token TRS e dopo aver verificato le seguenti condizioni:

- Non è possibile valutare contenuti con id non validi;
- Non è possibile valutare un contenuto condiviso da se stessi;
- Non è possibile valutare nuovamente un contenuto;
- È possibile valutare solamente contenuti creati (*con parentId 0*) e non contenuti inoltrati;
- Non deve essere scaduto il tempo di validazione per il contenuto che si vuole valutare;

```
function evaluateContent(
  uint contentId,
  bool evaluation,
  uint confidence
) external validContent(contentId) {
  if (msg.sender == s_contentSharing.getSharerOf(contentId)) {
    revert Errors.ContentEvaluation_AuthorCannotVote();
  }
  if (s_usersHasVoted[contentId][msg.sender]) {
    revert Errors.ContentEvaluation_AlreadyVoted();
  }
  if (s_contentSharing.isForwarded(contentId)) {
    revert Errors.ContentEvaluation_CannotEvaluateForwardedContent();
  }

  DataTypes.ContentValidation storage contentValidation = s_contentValidations[
      contentId];
  if (contentValidation.status != DataTypes.EvaluationStatus.Evaluating ) {
    revert Errors.ContentEvaluation_ContentValidationPeriodEnded();
  }

  i_trustToken.stakeTRS(msg.sender, i_trustToken.TRS_FOR_EVALUATION());

  s_usersHasVoted[contentId][msg.sender] = true;
  contentValidation.evaluations.push(
    DataTypes.Evaluation(msg.sender, evaluation, confidence)
  );
}
```

Listing 4.2: ContentEvaluation.sol::evaluateContent

Calcolo validità contenuto

La funzione `closeContentValidation`, mostrata in figura 4.3, si occupa di eseguire la validazione di un contenuto e, in caso sia valida, effettuare il tuning di affidabilità e trust token, altrimenti redistribuire lo stake.

Prima di tutto, viene controllato se è il periodo di validazione è terminato. In caso negativo, la funzione interrompe l'esecuzione e genera un errore. Se non sono presenti abbastanza valutazioni o voti, viene aggiornato lo stato di validazione e viene restituito un messaggio di errore.

Se invece ci sono abbastanza valutazioni, viene calcolata la valutazione finale utilizzando una funzione della libreria `ContentEvaluationCalculator`, che esegue l'algoritmo di validazione visto in 3.3.1. Se la

valutazione non è valida, viene aggiornato lo stato di validazione e viene restituito un messaggio di errore.

In caso contrario, la funzione aggiorna lo stato di validazione del contenuto a `Evaluated` e effettua il tuning dell'affidabilità e dei trust token del condivisore e dei valutatori attraverso la funzione della libreria `TokenAndTrustLevelTuning`, che esegue l'algoritmo visto in 3.3.2.

Infine, viene emesso l'evento `ContentEvaluated` che verrà indicizzato dal nostro Subgraph, riportando lo stato di validazione, la valutazione, la confidece generale e il numero di valutazioni.

```
function closeContentValidation(uint contentId) external validContent(contentId)
    returns (string memory response, bool evaluation, uint confidence, bool valid)
    {
  // Check if content is still in the validation period
  bool isUpkeepNeeded = checkContentValidation(contentId);
  if (!isUpkeepNeeded) {
    revert Errors.ContentEvaluation_UpkeepNotNeeded();
  }

  DataTypes.ContentValidation storage validation = s_contentValidations[contentId];
  address sharer = s_contentSharing.getSharerOf(contentId);
  uint evaluations = validation.evaluations.length;
  uint trustedUsers = i_trustToken.s_trustedUsers();

  // Check if there are enough evaluations to make a decision
  if (trustedUsers <= 1 || (evaluations <= trustedUsers / 2)) {
    TokenAndTrustLevelTuning.returnStake(sharer, validation, i_trustToken);
    validation.status = DataTypes.EvaluationStatus.NotVerified_NotEnoughVotes;

    emit Events.ContentEvaluated(contentId, validation.status, false, 0,
        evaluations);
    return ("Evaluation failed: Not enough votes", false, 0, false);
  }

  (evaluation, confidence, valid) = ContentEvaluationCalculator.getFinalEvaluation(
      validation, i_trustToken);
  if (!valid) {
    TokenAndTrustLevelTuning.returnStake(sharer, validation, i_trustToken);
    validation.status = DataTypes.EvaluationStatus.
        NotVerified_EvaluationEndedInATie;

    emit Events.ContentEvaluated(contentId, validation.status, false, 0,
        evaluations);
    return ("Evaluation failed: Tie", false, 0, valid);
  }

  validation.finalEvaluation = DataTypes.FinalEvaluation(evaluation, confidence);
  validation.status = DataTypes.EvaluationStatus.Evaluated;
  TokenAndTrustLevelTuning.tuneTrustLevelAndTrustToken(sharer, validation,
      i_trustToken);

  emit Events.ContentEvaluated(contentId, validation.status, evaluation, confidence
      , evaluations);
  return ("Evaluated", evaluation, confidence, valid);
}
```

Listing 4.3: ContentEvaluation.sol::closeContentValidation

4.1.3 TrustToken

In questa sezione andremo ad esplorare lo smart contract `TrustToken.sol`, che rappresenta l'implementazione del token TrustToken (*TRS*) basato sul protocollo ERC20. Analizzeremo le sue caratteristiche principali, compreso la modalità con cui avviene l'acquisto del

badge, come viene gestito lo staking, come viene calcolato il prezzo statico in USD e come avviene la restituzione di TRS.

Il contratto `TrustToken.sol` segue lo standard ERC20, che è un protocollo comune per i token basati su Ethereum. Questo significa che TrustToken è compatibile con le applicazioni decentralizzate (dApp) e gli scambi di criptovalute che supportano il protocollo ERC20. Essendo basato su questo standard, TrustToken eredita funzionalità come il trasferimento di token, l'approvazione di trasferimenti e altre funzioni standard associate ai token ERC20.

La funzione `buyBadge` permette agli utenti di acquistare un badge attraverso l'invio di una somma predefinita di ETH e ricevendo in cambio una quantità specifica di token TRS. All'interno della funzione, vengono gestiti diversi aspetti, come la verifica che l'utente non abbia già dei TRS, la conversione dell'ETH in token TRS utilizzando un rapporto di scambio predefinito e il trasferimento e eventuale minting dei token TRS all'utente. Questa funzione è progettata per garantire un processo di acquisto sicuro e trasparente per i badge all'interno del sistema TrustToken.

Il prezzo statico per l'acquisto del badge è definito in USD. Per effettuare la conversione da ETH in entrata a USD è stato utilizzato il sistema di oracle decentralizzato di `Chainlink Price Feeds`[5], che permette di ottenere il rapporto ETH/USD in tempo reale.

Il sistema di staking di TrustToken è gestito tramite l'ausilio di una struttura dati che memorizza la quantità di TRS attualmente in stake per i vari utenti e un insieme di funzioni, richiamabili solo dai due smart contract precedenti, per l'aggiunta e rimozione di TRS in stake.

All'interno del contratto TrustToken.sol, viene effettuato l'override del metodo `_update`. Questo metodo fa parte dell'interfaccia ERC20 e viene sovrascritto per personalizzare il comportamento del contratto TrustToken. Il metodo `_update` viene chiamato in determinati momenti, ad esempio quando avvengono transazioni o quando si effettua

il minting di token. La sua implementazione personalizzata permette di verificare se l'utente dispone dei token da inviare considerando anche quelli in stake.

4.2 Subgraph Application

Come introdotto precedentemente nella sezione 3.1, TheGraph è una soluzione che permette di indicizzare i dati su blockchain creando entità basate sugli eventi emessi dagli smart contract. Fornisce inoltre un'interfaccia di accesso ai dati tramite GraphQL, semplificando l'interazione con i dati on-chain.

La creazione di un subgraph avviene attraverso la definizione dell'entità del sistema, seguita dallo sviluppo di alcune funzioni di handler degli eventi trasmessi. Questi handler sono responsabili di interpretare i dati dell'evento e convertirli in informazioni comprensibili per le entità precedentemente definite.

Le entità del sistema sono definite all'interno del file `schema.graphql`, visibile in figura 4.1. Nello specifico l'entità definite sono `Content` e l'entità `Evaluation`, rappresentano rispettivamente i dati relativi ai contenuti presenti sulla piattaforma e i dati relativi alle valutazioni dei contenuti.

```
enum EvaluationStatus {
  Evaluating,
  Evaluated,
  NotVerified_NotEnoughVotes,
  NotVerified_EvaluationEndedInATie
}

type Evaluation @entity {
  id: ID!
  evaluation: Boolean
  confidence: BigInt
  status: EvaluationStatus!
  evaluationsCount: BigInt
}

type Content @entity {
  id: ID!
  sender: Bytes!
  title: String!
  ipfsCid: String!
  chatName: String!
  parentContent: Content
  isForwaded: Boolean!
  evaluation: Evaluation!
  forwarded: [Content!]
  # block transaction infos
  blockNumber: BigInt!
  blockTimestamp: BigInt!
  transactionHash: Bytes!
}
```

Figura 4.1: schema.graphql

Per ottenere i dati on-chain utilizzando TheGraph, è necessario ascoltare gli eventi emessi dagli smart contract. All'interno dei file `content-sharing.ts` e `content-evaluation.ts`, sono implementati i meccanismi di listening sugli eventi trasmessi dagli smart contract pertinenti al contenuto e alla valutazione. Questi meccanismi consentono di catturare gli eventi emessi e di creare nuove entità o aggiornare entità esistenti.

In figura 4.4 è possibile vedere la funzione `handleContentCreated` che gestisce l'evento *ContentCreated*. Questa funzione viene chiamata quando viene creato un nuovo contenuto sulla piattaforma. All'interno di questa funzione, viene creata un'istanza dell'entità `Content` corrispondente all'id del contenuto creato. Vengono quindi impostate

le proprietà dell'entità con i valori provenienti dall'evento, come l'indirizzo del mittente, il titolo, l'identificatore IPFS, il nome del canale di chat e altre informazioni correlate al blocco e alla transazione. Se il contenuto è un inoltro, viene aggiornato il contenuto padre e il riferimento alla valutazione del contenuto viene impostato con il valore dell'entità `Evaluation` del contenuto padre. Se il contenuto non è un inoltro, viene creata una nuova entità `Evaluation` iniziale e viene assegnato il valore *Evaluating* allo stato della valutazione. Infine, l'entità `Content` viene salvata.

```
export function handleContentCreated(event: ContentCreatedEvent): void {
  let content = new Content(event.params.id.toString())
  content.sender = event.params.sender
  content.title = event.params.title
  content.ipfsCid = event.params.ipfsCid
  content.chatName = event.params.chatName
  content.forwarded = []
  // Parent Content Infos
  content.isForwaded = event.params.parentContent.notEqual(BigInt.zero())
  if (content.isForwaded) {
    updateParentForwardedContent(content, event.params.parentContent)
    content.parentContent = event.params.parentContent.toString()
    // Link to parent content evaluation
    content.evaluation = Content.load(content.parentContent!)!.evaluation
  } else {
    content.parentContent = "0"
    content.evaluation = createInitialEvaluation(event).id
  }

  content.blockNumber = event.block.number
  content.blockTimestamp = event.block.timestamp
  content.transactionHash = event.transaction.hash

  content.save()
}
```

Listing 4.4: content-sharing.ts:handleContentCreated

L'implementazione di TheGraph nel progetto ci ha permesso di ottenere i dati on-chain in modo efficiente e semplificato. Utilizzando TheGraph, abbiamo potuto indicizzare i dati creando entità basate sugli eventi emessi dagli smart contract, e fornire un'interfaccia di accesso ai dati tramite GraphQL. L'analisi dell'entità presente in `schema.graphql` e dei meccanismi di listening sugli eventi nei file `content-sharing.ts` e `content-evaluation.ts` ci ha fornito una panoramica

completa sull'implementazione di TheGraph nel progetto.

Nelle figure 4.2 e 4.3 è possibile visualizzare un esempio di query per l'ottenimento dei contenuti condivisi e relativo risultato.

```
contents(orderBy: blockTimestamp) {
  id
  sender
  title
  ipfsCid
  chatName
  isForwaded
  parentContent {
    id
    title
  }
  evaluation {
    id
    status
    evaluationsCount
  }
  forwarded {
    id
    title
  }
  blockTimestamp
  blockNumber
  transactionHash
}
```

Figura 4.2: GraphQL query example

```
{
  "data": {
    "contents": [
      {
        "id": "1",
        "sender": "0×f39fd6e51aad88f6f4ce6ab8827279cfffb92266",
        "title": "Title",
        "ipfsCid": "https://ipfs.filebase.io/ipfs/QmSnuWmxptJZdLJpKRarxBMS2Ju2oANVrgbr2xWbie9b2D/",
        "chatName": "Twitter",
        "isForwaded": false,
        "parentContent": null,
        "evaluation": {
          "id": "1",
          "status": "NotVerified_NotEnoughVotes",
          "evaluationsCount": "1"
        },
        "forwarded": [
          {
            "id": "3",
            "title": "Title"
          }
        ],
        "blockTimestamp": "1701099317",
        "blockNumber": "81",
        "transactionHash": "0×1cb2e520317168da60dd6f8b17572e42f017e7f742cbefc024471b6265603390"
      },
      {
        "id": "2",
        "sender": "0×f39fd6e51aad88f6f4ce6ab8827279cfffb92266",
        "title": "Title2",
        "ipfsCid": "https://ipfs.filebase.io/ipfs/bafyreicnokmhmrnlp2wjhyk2haep4tqxiptwfrp2rrs7rzq7uk766chqvq",
        "chatName": "Livechat:Scuola",
        "isForwaded": false,
        "parentContent": null,
        "evaluation": {
          "id": "2",
          "status": "Evaluating",
          "evaluationsCount": null
        },
        "forwarded": [],
        "blockTimestamp": "1701099334",
        "blockNumber": "83",
        "transactionHash": "0×8c5d3608a6bb3f28ef2d06790cadea964acdf9b51418149b8bcc31c286f9db01"
      },
      {
        "id": "3",
        "sender": "0×70997970c51812dc3a010c7d01b50e0d17dc79c8",
        "title": "Title",
        "ipfsCid": "https://ipfs.filebase.io/ipfs/QmSnuWmxptJZdLJpKRarxBMS2Ju2oANVrgbr2xWbie9b2D/",
        "chatName": "Whatsapp:Amici",
        "isForwaded": true,
        "parentContent": {
          "id": "1",
          "title": "Title"
        },
        "evaluation": {
          "id": "1",
          "status": "NotVerified_NotEnoughVotes",
          "evaluationsCount": "1"
        },
        "forwarded": [],
        "blockTimestamp": "1701099348",
        "blockNumber": "87",
        "transactionHash": "0×5b296b083fb658863bbfe1848f0a01b283db950423c221b8ef8d5973e5766548"
      }
    ]
  }
}
```


Figura 4.3: GraphQL response example

Capitolo 5

Validazione

5.1 Integrazione su dApp

Le decentralized applications (dApp) stanno rivoluzionando il modo in cui interagiamo con le applicazioni online, garantendo una maggiore sicurezza e controllo agli utenti. Una dApp[33] è un'applicazione software che funziona su una rete decentralizzata come una blockchain. Utilizza smart contract per interfacciarsi con i dati presenti sulla blockchain.

5.1.1 Livechat Integration

Livechat è un'applicazione mobile di messaggistica istantanea con funzionalità social avanzate come la condivisione della posizione e una classifica basata sul numero di passi effettuati giornalmente.[1]

La sezione interessata per l'integrazione di SocialTrustr è la sezione di chat, visibile in figura 5.1. Questa integrazione permette di implementare un sistema di valutazione e fiducia tra gli utenti, migliorando la qualità e l'affidabilità degli scambi di messaggi all'interno dell'app.

[1] https://github.com/ManuelArto/LiveChat

Ogni utente potrà così visualizzare a fianco di ogni messaggio l'affidabilità dell'utente che lo ha condiviso e l'esito della validazione di quel messaggio.

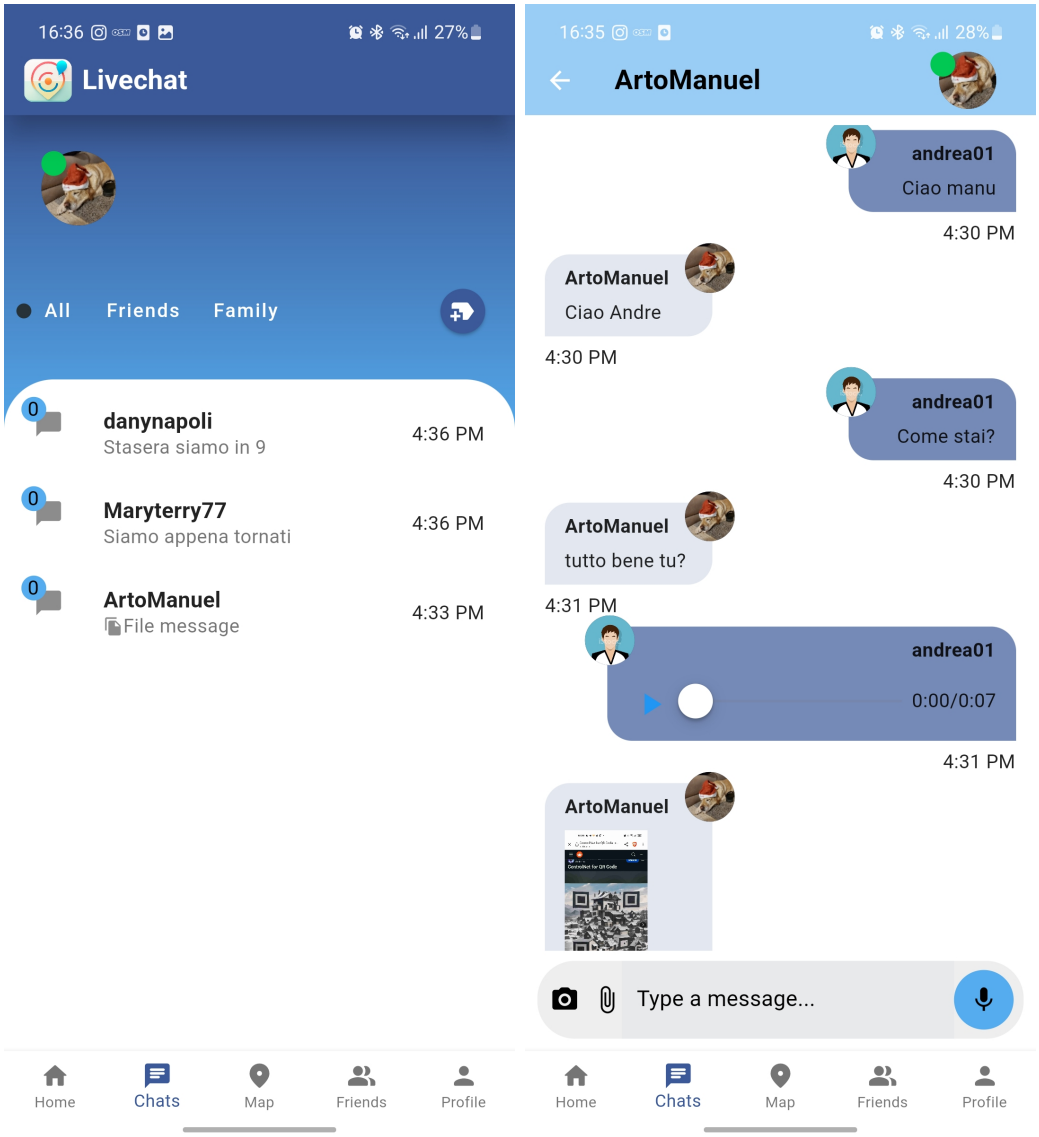

Figura 5.1: Livechat - Chat Screen

L'approccio web3 è stato adottato per rendere Livechat una App decentralizzata. Questo approccio consente di fornire agli utenti un sistema per valutare e tracciare i messaggi scambiati. Librerie come

`web3dart`[21] sono state utilizzate per integrare il supporto di Web3 e blockchain all'interno dell'app, consentendo transazioni sicure e interazioni affidabili con gli smart contract di SocialTrustr.

Il protocollo `WalletConnect`[30] è stato implementato per consentire agli utenti di connettere il proprio portafoglio crittografico alle funzionalità di SocialTrustr direttamente dall'app Livechat. Questo processo semplifica l'esperienza utente, consentendo loro di utilizzare il proprio portafoglio crittografico per partecipare alle attività di valutazione e condivisione di contenuti senza dover abbandonare l'ambiente di Livechat.

L'ottenimento dei dati da blockchain è stato ottimizzato attraverso l'utilizzo di GraphQL interrogando direttamente l'applicazione realizzata su TheGraph. Questo approccio consente di interrogare in modo efficiente i dati on-chain necessari per le funzionalità di SocialTrustr, garantendo tempi di risposta rapidi e un'esperienza utente fluida.

La chiarezza e la semplicità dell'integrazione di SocialTrustr su Livechat illustrano l'efficacia di adottare soluzioni decentralizzate anche in contesti complessi come le dApp di messaggistica istantanea. La trasparenza e la sicurezza offerte dalla blockchain contribuiscono a migliorare le dinamiche sociali all'interno di Livechat.

Flusso valutazione contenuto

Ogni volta che un utente invia un messaggio condivisibile, questo e gli eventuali inoltri verranno salvati su blockchain attraverso una transazione. L'upload su ipfs del contenuto del messaggio avverrà lato client e verrà poi salvato on-chain solo il `Content Identifier`.

All'interno della sezione di chatting di Livechat, è stata implementata una pagina relativa all'interazione con SocialTrustr. Quello che segue sono i vari passaggi che un utente deve effettuare all'interno di tale sezione per valutare un contenuto:

1. **Connettere il proprio wallet**: grazie al protocollo walletconnect l'utente può, attraverso un bottone, connettere il proprio wallet (e.g. metamask), che verrà poi utilizzato per effettuare transazioni ed interagire con SocialTrustr.
2. **Diventare un utente Trusted**: come specificato in 3.2.1, per diventare un utente Trusted bisogna acquistare un badge, ossia una quantità di Token TRS definita dal sistema. Sarà presente un bottone che richiama il metodo `buyBadge` dello smart contract presentato in 4.1.3.
3. **Visualizzazione contenuti**: tramite una query graphQL sarà possibile ottenere tutti i contenuti salvati on-chain. Sarà inoltre possibile suddividerli in già validati o in validazione.
4. **Valutare un contenuto**: è possibile selezionare un contenuto dalla lista dei contenuti in validazione ed effettuare una propria valutazione associando un confidence score. Il sistema in automatico si occuperà di effettuare lo stake dei token richiesti.

5.2 Analisi

L'abilità di valutare la resistenza di un sistema alle manipolazioni e la sua capacità di mantenere un comportamento coerente con le aspettative in presenza di comportamenti malevoli è un aspetto critico nello sviluppo di sistemi sicuri e affidabili. Nella seguente sezione, esamineremo la robustezza di SocialTrustr, un sistema che si propone di garantire integrità e veridicità dei contenuti in ambito digitale. Analizzeremo come il sistema è protetto contro i tentativi di manipolazione, come l'attacco del 51% e gli attacchi Sybil, e valuteremo l'efficacia del meccanismo di tuning basato sull'entropia per incentivare l'onestà e scoraggiare le manipolazioni. Infine, ci occuperemo di esplorare vari casi d'uso, dimostrando come SocialTrustr possa essere integrato

in diverse applicazioni come un servizio affidabile per la verifica della veridicità dei contenuti.

5.2.1 Resistenza agli attacchi di manipolazione del sistema

Nel capitolo 3, abbiamo introdotto il `Sistema di voting` e il `Modello predittivo probabilistico` di SocialTrustr delineando come ogni utente Trusted può impiegare i propri token TRS per condividere e valutare i contenuti. L'approccio adottato mira a rafforzare la veridicità delle informazioni e a sviluppare una comunità basata sulla fiducia reciproca.

Per quanto riguarda la resistenza agli attacchi, il sistema deve essere capace di contrastare le manipolazioni tipiche delle piattaforme basate su voto, come il **51% attack** e il **Sybil attack**. Nel 51% attack, un attore malevolo potrebbe acquisire la maggioranza dell'influenza nel sistema per manipolare i processi di valutazione dei contenuti. Tuttavia, SocialTrustr cerca di mitigare questo rischio tramite tre meccanismi:

- L'incentivo tramite entropia rende difficile aumentare la propria affidabilità in autonomia;
- Le ricompense sono inversamente proporzionali alla attuale affidabilità;
- Le ricompense sono sempre distribuite, perciò risulta difficile nel tempo acquisire una grossa affidabilità in autonomia;

Il Sybil attack, in cui un utente crea molteplici identità false per ottenere un'influenza sproporzionata, è contrastato dal requisito che ogni utente deve possedere dei token TRS da stakeare per partecipare, rendendo onerosa la creazione di false identità in numero significativo.

5.2.2 Analisi del meccanismo di tuning tramite entropia

Il meccanismo di tuning tramite entropia in SocialTrustr è progettato per incentivare l'onestà e penalizzare le manipolazioni attraverso

un processo di redistribuzione dei token TRS e l'aggiornamento dell'affidabilità degli utenti. Analizziamo i principali elementi di questo meccanismo:

1. **Redistribuzione dei token**: Quando un utente condivide un contenuto mette in stake una parte dei propri TRS, così come ogni utente che effettua una valutazione. In base all'esito del contenuto, il sistema restituisce i token a chi ha valutato correttamente, incentivando l'onestà. Chi ha votato in modo contrario, subisce una penalità e vengono confiscati i TRS in base al confidence score e vengono redistribuiti agli utente corretti. L'aggiunta del confidence score permette non solo di moderare l'influenza che si ha nel sistema di valutazione, ma anche di effettuare una vera e propria scommessa; infatti maggiore sarà il confidence score maggiori saranno ricompense/penalità;
2. **Aggiornamento dell'affidabilità degli utenti**: L'affidabilità degli utenti viene aggiornata in base alle valutazioni corrette e incorrette. Gli utenti che valutano correttamente vedono aumentare la loro affidabilità, mentre quelli che valutano incorrettamente subiscono una diminuzione. L'entropia è incorporata in questo processo, favorendo una maggiore varietà di valutazioni e disincentivando il comportamento manipolativo.
3. **Entropia dei Trust Scores**: L'entropia viene calcolata per misurare la varietà delle valutazioni ricevute da un utente. Una bassa entropia indica che un utente ha ricevuto valutazioni uniformi, mentre una maggiore entropia indica una distribuzione più equa. Questo contribuisce a rendere più difficile per gli utenti manipolare il sistema concentrando le proprie valutazioni in modo selettivo.

Simulazione Montecarlo

In questa sezione si discutono i risultati di una simulazione Monte Carlo[12] che dimostrano come il meccanismo di SocialTrustr incentivi il comportamento onesto e penalizzi le manipolazioni all'interno di un ecosistema digitale decentralizzato. Una nota importante da considerare è che lo script della simulazione è fondato su una generazione di valutazioni pseudo randomiche che di conseguenza non simulano correttamente un'applicazione reale del sistema, in cui la valutazione di un utente è fondata su una serie di fattori complessi come bias o conoscenze personali sul contenuto che si sta valutando.

La simulazione è stata eseguita su 10 utenti e 200 iterazioni. Lo stato iniziale del sistema è rappresentato da una lista dei livelli di fiducia (impostata a 50 inizialmente) e una lista relativa alla quantità di token TRS (impostata a 500 inizialmente) per ogni utente. In ogni iterazione della simulazione, si sceglie casualmente un utente che condivide un contenuto e mette in stake un certo numero di token (impostato a 20 TRS). Solo gli utenti con abbastanza token possono condividere e valutare contenuti.

Vengono simulati gli altri utenti che mettono in stake una quantità di TRS (impostata a 10 TRS) e valutano l'affidabilità del contenuto condiviso con una proporzione del 70% e 30% rispettivamente per `Vero` e `Falso` e un confidence score in un range del 40%-100%. Questa decisione è stata presa per simulare un ambiente più realistico in cui gli utenti si impegnano a condividere contenuti onesti. Al termina di ogni iterazione vengono raccolti i dati della simulazione, inclusi affidabilità e token TRS. L'esame dei grafici forniti da questi dati rivela delle informazioni sui meccanismi fondamentali che definiscono l'ecosistema SocialTrustr.

Per quanto riguarda il terzo grafico, rappresentante il delta relativo all'incremento/riduzione dell'affidabilità per ogni entropia calcolata, è stata eseguita la stessa simulazione solamente per una iterazione

ripetuta per `10.000` volte.

Adesso passiamo ad analizzare visivamente i risultati dei tre grafici generati:

Affidabilità

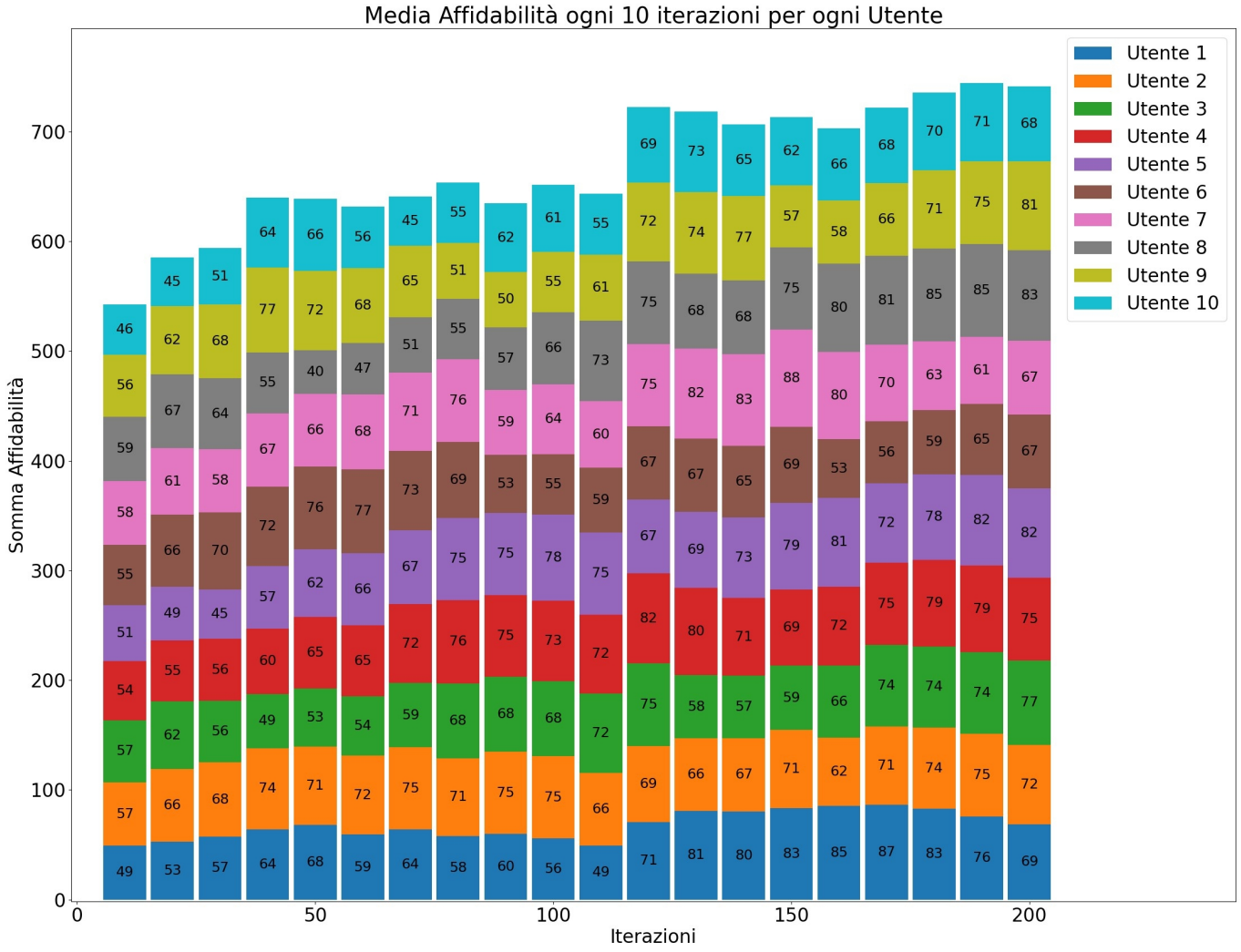

Figura 5.2: Analisi affidabilità su 10 utenti e 200 iterazioni

Il primo grafico esamina l'evoluzione dell'affidabilità all'interno della simulazione. L'esito di una validazione è determinato da una media ponderata delle valutazioni, di conseguenza il sistema premierà sem-

pre la maggioranza degli utenti. Questo è visibile nel grafico grazie all'andamento crescente della somma delle affidabilità. Inoltre, relativamente all'ultimo insieme di iterazioni, è possibile notare come non ci siano utenti che detengono una affidabilità di gran lunga maggiore rispetto agli altri. In altre parole risulta impossibile aumentare la propria affidabilità in autonomia e questo previene una centralizzazione del potere decisionale. Risulta invece evidente l'impatto quasi raddoppiato delle penalizzazioni (M=2.5 nell'equazione 3.9), ma che comunque non limita l'utente a incrementare la sua affidabilità in caso di successivi comportamenti corretti, due esempi visibili nel grafico sono il caso degli utenti numero `8` e `9` che si trovano ad avere una affidabilità inferiore rispetto agli altri all'iterazione 100 e concludono con un affidabilità sopra a 80 alla fine della simulazione.

Token TRS

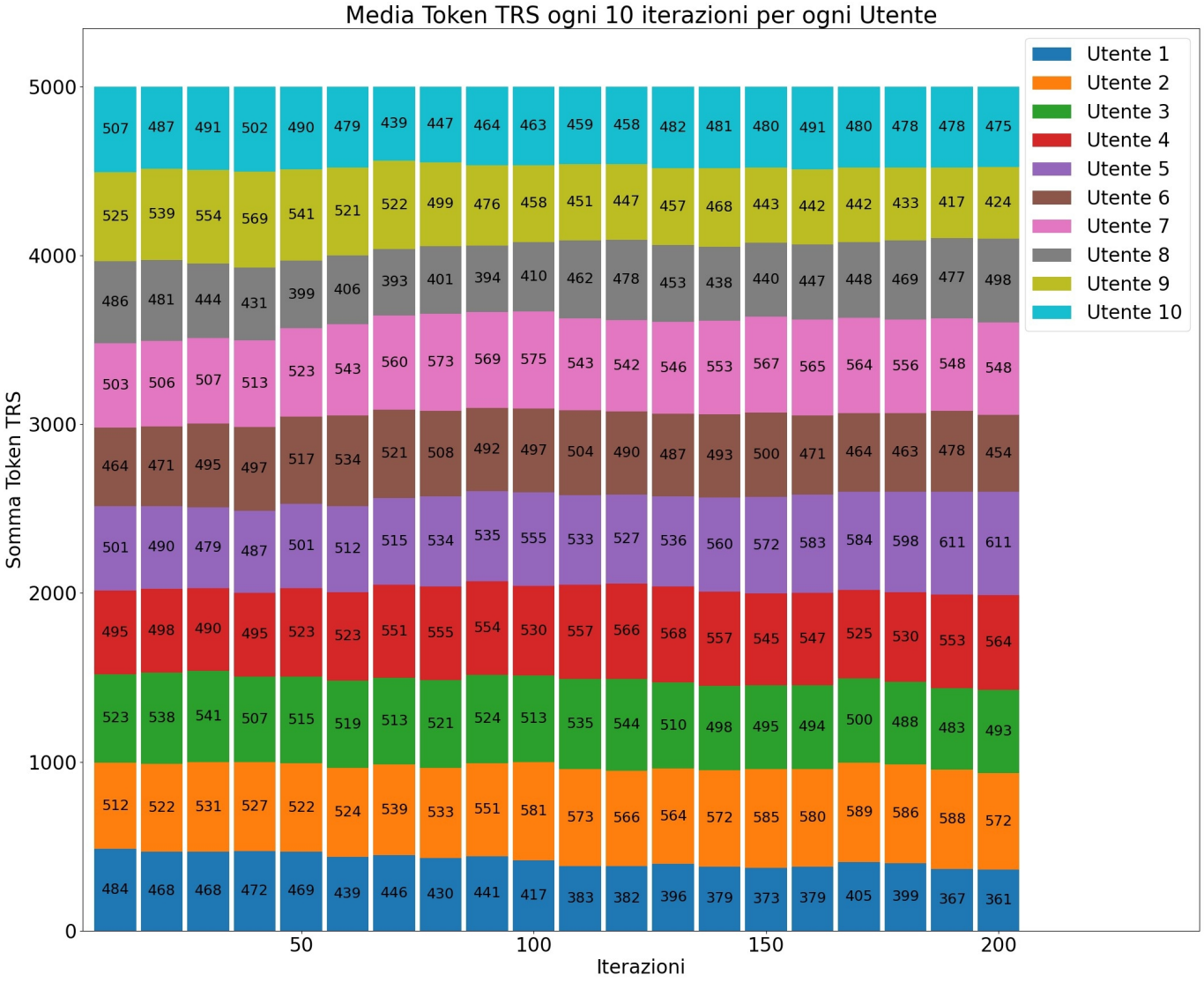

Figura 5.3: Analisi redistribuzione TRS su 10 utenti e 200 iterazioni

Il grafico dei token TRS rivela la dinamicità dei fattori economici del sistema. Nel grafico è possibile vedere come a ogni gruppo di iterazioni la quantità totale di TRS rimanga invariata. Questo accade poichè i TRS persi vengono redistribuiti come ricompensa agli utenti che hanno valutato correttamente; questo permette al sistema di resistere agli accumuli e alle manipolazioni. È importante notare che, essendo solo dieci utenti all'interno della simulazione, i guadagni non

saranno elevati, infatti dopo le duecento iterazioni l'utente con il maggiore numero di TRS ha ottenuto un guadagno totale di soli 100 TRS circa. L'introduzione di un costo in token per la condivisione e la valutazione agisce come meccanismo deterrente che disincentiva attività malintenzionate quali spam e falsificazione.

Entropia e delta incremento/riduzione affidabilità

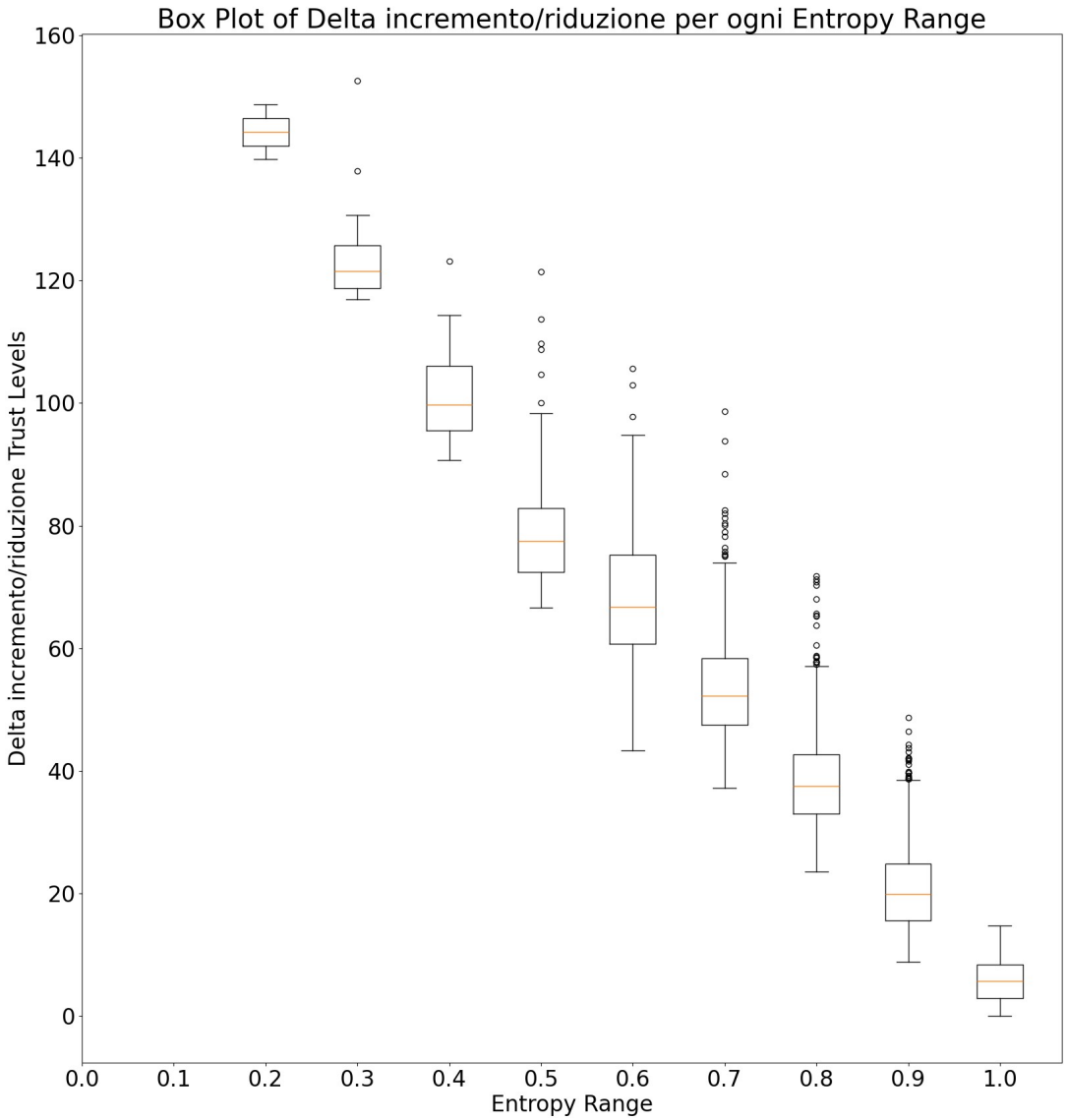

Figura 5.4: Analisi entropia e delta incremento/riduzione affidabilità su 10 utenti e 10.000 ripetizioni

L'entropia è un elemento cruciale nel meccanismo di tuning di Social-Trustr. Il grafico illustra le fluttuazioni del delta relativo a incremento e riduzione dell'affidabilità degli utenti per ogni valore dell'entropia calcolato nella prima iterazione di una simulazione ripetuta `10.000` volte. Una bassa entropia è indicativa di un sistema ordinato, con

comportamenti prevedibili e un'elevata affidabilità delle valutazioni. Al contrario, un'entropia elevata può segnalare la presenza di tentativi di manipolazione o di incertezza nelle valutazioni. La caratteristica più importante che si evince dal grafico è la diminuzione del delta incremento/riduzione all'aumentare dell'entropia. Questo consente di ridurre le ricompense e le penalizzazioni dell'affidabilità in caso di alto disordine della valutazione, come mostrato nelle equazioni 3.8 e 3.9. Grazie a questo sistema è possibile non solo ridurre le regolazioni dell'affidabilità in caso di validazioni incerte, ma permette anche di punire in modo più drastico utenti che valutano in modo anomalo confronto alla maggioranza, infatti utilizzando l'entropia come meccanismo di tuning, il sistema può aggiustare automaticamente i parametri relativi all'affidabilità, rinforzando le difese contro le manipolazioni e mantenendo alta l'onestà del comportamento collettivo.

Dalle analisi effettuate emerge che il sistema SocialTrustr presenta un meccanismo robusto e sofisticato di tuning basato sull'entropia, capace di incentivare l'onestà e contenere le manipolazioni. Con il contributo degli insight empirici forniti dalla simulazione Monte Carlo, si possono identificare potenziali vettori d'attacco e migliorare le strategie di difesa. In particolare, i risultati suggeriscono un percorso di ottimizzazione continua del sistema basato su algoritmi adattivi che regolino il flusso di token e la valutazione di affidabilità degli utenti in risposta alle variazioni di entropia.

5.2.3 Casi d'uso

SocialTrustr non si propone solo come un sistema stand-alone ma piuttosto come un framework integrabile in applicazioni esistenti, offrendo un modello scalabile di gestione della fiducia e validazione dei contenuti agli sviluppatori di app social. Grazie alla sua natura modulare, può essere adattato per vari scenari, dal moderare le fake news alla valutazione di qualunque contenuto online.

L'integrabilità di SocialTrustr in diverse piattaforme assicura che sia i piccoli che i grandi ecosistemi digitali possano beneficiare di una soluzione decentralizzata per la valutazione della veridicità dei contenuti, trasferendo così la responsabilità e l'autorità direttamente agli utenti piuttosto che a un ente centrale. Con il crescente bisogno di trasparenza e responsabilità nelle piattaforme digitali, SocialTrustr offre una proposta innovativa in grado di rispondere a questi imperativi.

Questi esempi vanno solo a illustrare una frazione delle potenziali applicazioni di SocialTrustr, dimostrando l'estensibilità e la versatilità di questo strumento nella lotta contro la disinformazione online.

Capitolo 6

Conclusioni e sviluppi futuri

La presente tesi ha esplorato il concetto, la progettazione e l'implementazione di SocialTrustr, un sistema basato sulla tecnologia blockchain per migliorare l'autenticità e la tracciabilità dei contenuti condivisi online. La motivazione alla base di questo lavoro deriva dalla crescente preoccupazione per la diffusione di disinformazione e fake news, che ha sollevato un'esigenza urgente di strumenti di verifica e validazione fidati da parte degli utenti.

SocialTrustr indirizza questa esigenza introducendo un token digitale e un sistema di incentivi basato sull'entropia per ridurre le manipolazioni e premiare i comportamenti onesti. Integrato in piattaforme sociali e di messaggistica, come Livechat, SocialTrustr permette agli utenti di assegnare e ricevere feedback sulla veridicità dei contenuti, con le loro valutazioni che influenzano direttamente la reputazione online di altri utenti e contenuti. Questo approccio, che decentralizza il controllo della veridicità e lo distribuisce tra i pari, si propone come una soluzione scalabile e resistente alla censura, ribaltando il paradigma centralizzato tipico delle odierne authority di verifica.

La tesi ha illustrato in dettaglio un'analisi approfondita del problema della tracciabilità dei contenuti e ha sviluppato un modello robusto per SocialTrustr, incluso il meccanismo di incentivazione basato sul-

l'entropia che funge da importante deterrente alle manipolazioni. Attraverso varie analisi svolte, è stato possibile osservare l'efficacia teorica del sistema nel promuovere l'onestà e nel punire comportamenti ingannevoli.

Eventuali sviluppi futuri di SocialTrustr includono principalmente la sua verifica in scenari con utenti reali per testarne l'efficacia pratica e raccogliere dati comportamentali significativi. Questi test offriranno l'opportunità di ottimizzare e affinare i meccanismi che regolano le ricompense e le penalizzazioni, migliorando così l'efficienza del sistema. Successivamente, si cercherà di espandere l'applicazione di SocialTrustr su piattaforme di più vasta portata, per osservarne l'effetto su una scala più ampia. Infine l'ultimo aspetto riguarda l'interoperabilità con diverse blockchain, al fine ampliare la portata del progetto, migliorare la sicurezza e ottimizzare i costi delle transazioni on-chain.

Riferimenti bibliografici

[1] Mahmoudreza Babaei, Abhijnan Chakraborty, Juhi Kulshrestha, Elissa M. Redmiles, M. Cha, and Krishna P. Gummadi. Analyzing biases in perception of truth in news stories and their implications for fact checking. *IEEE Transactions on Computational Social Systems*, 9:839–850, 2018.

[2] B Sobhan Babu, Poojitha Tatineni, Lakshmi Narayana I, Chitturi Prasad, Hema Chindu, and Bala Bhaskara Rao Emani. The graph - a decentralized query protocol for block chain technology. In *2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 1–7, 2021.

[3] Juan Benet. Ipfs-content addressed, versioned, p2p file system. *arXiv preprint arXiv:1407.3561*, 2014.

[4] Nakamoto S Bitcoin. Bitcoin: A peer-to-peer electronic cash system, 2008.

[5] Chainlink. Using Data Feeds — Chainlink Documentation — docs.chain.link. `https://docs.chain.link/data-feeds/price-feeds`.

[6] Chien-Chih Chen, Yuxuan Du, Richards Peter, and Wojciech Golab. An implementation of fake news prevention by blockchain and entropy-based incentive mechanism. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 2476–2486, 2021.

[7] Tim Draws, David La Barbera, Michael Soprano, Kevin Roitero, Davide Ceolin, Alessandro Checco, and Stefano Mizzaro. The effects of crowd worker biases in fact-checking tasks. *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022.

[8] foundry rs. Foundry. `https://github.com/foundry-rs/foundry`.

[9] Paula Fraga-Lamas and Tiago M. Fernández-Caramés. Fake news, disinformation, and deepfakes: Leveraging distributed ledger technologies and blockchain to combat digital deception and counterfeit reality. *IT Professional*, 22(2):53–59, 2020.

[10] Zhijiang Guo, M. Schlichtkrull, and Andreas Vlachos. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206, 2021.

[11] Jeffrey T. Hancock and Jeremy N. Bailenson. The social impact of deepfakes. *Cyberpsychology, Behavior, and Social Networking*, 24(3):149–152, 2021. PMID: 33760669.

[12] Robert L Harrison. Introduction to monte carlo simulation. In *AIP conference proceedings*, volume 1204, pages 17–21. American Institute of Physics, 2010.

[13] Tim Hayward. The problem of disinformation. *Available at SSRN 4502104*, 2023.

[14] Zakwan Jaroucheh, Mohamad Alissa, William J Buchanan, and Xiaodong Liu. Trustd: Combat fake content using blockchain and collective signature technologies. In *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 1235–1240, 2020.

[15] Sunny King and Scott Nadal. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. *self-published paper, August*, 19(1),

2012.

[16] Srijan Kumar and Neil Shah. False information on web and social media: A survey. *ArXiv*, abs/1804.08559, 2018.

[17] Annick Lesne. Shannon entropy: a rigorous notion at the crossroads between probability, information theory, dynamical systems and statistical physics. *Mathematical Structures in Computer Science*, 24(3):e240311, 2014.

[18] Ziyao Liu, Nguyen Cong Luong, Wenbo Wang, Dusit Niyato, Ping Wang, Ying-Chang Liang, and Dong In Kim. A survey on applications of game theory in blockchain. *arXiv preprint arXiv:1902.10865*, 2019.

[19] Faraz Masood and Arman Rasool Faridi. Distributed ledger technology for closed environment. In *2019 6th International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 1151–1156. IEEE, 2019.

[20] Jeroen Ooms and Facebook, Inc. *graphql: A GraphQL Query Parser*, 2023. https://docs.ropensci.org/graphql/, https://graphql.org (upstream) https://github.com/ropensci/graphql (devel).

[21] pwa.ir. web3dart. `https://pub.dev/packages/web3dart`.

[22] Eishvak Sengupta, Renuka Nagpal, Deepti Mehrotra, and Gautam Srivastava. Problock: a novel approach for fake news detection. *Cluster Computing*, 24, 12 2021.

[23] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.

[24] Jieun Shin and Kjerstin Thorson. Partisan selective sharing: The biased diffusion of fact-checking messages on social media. *Journal of Communication*, 67:233–255, 2017.

[25] Kai Shu, Amrita Bhattacharjee, Faisal Alatawi, Tahora H. Nazer, Kaize Ding, Mansooreh Karami, and Huan Liu. Combating disinformation in a social media age. *WIREs Data Mining and Knowledge Discovery*, 10(6):e1385, 2020.

[26] James Surowiecki. *The wisdom of crowds*. Anchor, 2005.

[27] Emily A. Thorson. Belief echoes: The persistent effects of corrected misinformation. *Political Communication*, 33:460 – 480, 2016.

[28] Michela Del Vicario, Alessandro Bessi, Fabiana Zollo, Fabio Petroni, Antonio Scala, Guido Caldarelli, Harry Eugene Stanley, and Walter Quattrociocchi. The spreading of misinformation online. *Proceedings of the National Academy of Sciences*, 113:554 – 559, 2016.

[29] Soroush Vosoughi, Deb K. Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359:1146 – 1151, 2018.

[30] Inc WalletConnect. Walletconnect. `https://github.com/WalletConnect`.

[31] Nathan Walter, Jonathan Cohen, R Lance Holbert, and Yasmin Morag. Fact-checking: A meta-analysis of what works and for whom. *Political Communication*, 37(3):350–375, 2020.

[32] Xiaowan Wang, Huiyin Xie, Shan Ji, Liang Liu, and Ding Huang. Blockchain-based fake news traceability and verification mechanism. *Heliyon*, 9(7):e17084, 2023.

[33] Kaidong Wu, Yun Ma, Gang Huang, and Xuanzhe Liu. A first look at blockchain-based decentralized applications. *Software: Practice and Experience*, 51(10):2033–2050, 2021.

[34] Tolga Yilmaz and Özgür Ulusoy. Modeling and mitigating online misinformation: a suggested blockchain approach. *ArXiv*, abs/2303.10765, 2023.