University of Salerno

Department of Computer Science

Dottorato di Ricerca in Informatica
Curriculum Computer Science and Information Technology
XXXV Ciclo

Tesi di Dottorato / Ph.D. Thesis

Metodi matematici e tecnologie per la trasformazione digitale e le tokenizzazioni

Antonio RAPUANO

Supervisor:	**Gerardo IOVANE**
PhD Program Director:	**Prof. Andrea DE LUCIA**

A.A 2021/2022

Dedicato a mio nonno.

ABSTRACT

Questo lavoro ha un obiettivo duplice, che viene raggiunto tramite due filoni di ricerca. Il primo filone, di carattere prettamente industriale, si concentra sulle difficoltà tecnico-scientifiche legate alla raccolta, tokenizzazione e generazione di valore dei dati generati durante l'esecuzione di attività lavorative. In particolare, la soluzione proposta utilizza una combinazione di tecnologie IoT e Smart Contract per la raccolta dei dati e la creazione di asset blockchain basati sulle attività lavorative, fornendo allo stesso tempo un pilot study per la gestione di pagamenti automatici aziendali. Tale soluzione è all'avanguardia rispetto alla letteratura in termini di scalabilità, creazione di valore, sicurezza, attenzione alla regolamentazione e protezione dei diritti delle parti.

Il secondo filone, prettamente scientifico, si concentra sull'applicazione della Financial Computing ai token generati dagli Smart Contract durante la raccolta dei dati, attraverso l'utilizzo di metodi matematici che strizzano l'occhio alla filosofia del *Value Investing* per la valutazione degli asset. Il fine è quello di generare business model e favorire la misura del rischio finanziario di portafogli strutturati per supportare il processo decisionale in contesti di info-incertezza e info-incompletezza nella quale prevale la soggettività e l'esperienza.

Tale valutazione nasce dalla teoria della probabilità estesa. Essa permette di ricostruire l'accadibilità di un evento in contesti nella quale le informazioni sono mancanti, ed è applicabile ai mercati finanziari, dove sono necessarie decisioni basate su informazioni soggettive e incomplete. La teoria della probabilità estesa e la sua applicazione a strumenti finanziari viene validata attraverso una sperimentazione su asset immateriali facenti parte del mercato sportivo. Successivamente viene utilizzata per la costruzione di un modello chiamato Token Evaluation System. Esso permette di effettuare una valutazione di asset digitali e di favorire la gestione del rischio in un portafoglio strutturato. In sintesi, questo lavoro presenta una solida base teorica e metodologica, unita

a una dimostrazione pratica attraverso la realizzazione di una Proof of Concept, che dimostra l'efficacia della soluzione proposta e la sua capacità di generare valore per l'industria. Questo lavoro rappresenta un passo avanti importante nell'utilizzo della tecnologia blockchain e della Financial Computing per la valutazione e la gestione del rischio nei mercati finanziari innovativi, favorendo allo stesso tempo una maggiore trasparenza, sicurezza e automazione nella gestione del lavoro e degli asset associati, aprendo la strada a nuove opportunità per l'efficienza e la crescita economica.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

PoW Proof of Work
PoS Proof of Stake
PoC Proof of Concept
PoA Proof of Authority
DeFi Finanza Decentralizzata
NFT Non Fungible Token
UTXO Unspent Transaction Output
ICO Initial Coin Offering
DEX Exchange Decentralizzati
EVM Ethereum Virtual Machine
IPFS InterPlanetary File System

TES	Token Evaluation System
DeFi	Decentralized Finance
CSF	Critical Success Factor
IoT	Internet of Things
EVM	Ethereum Virtual Machine
ANN	Artificial Neural Network
DLT	Distributed Ledger Technologies
DAO	Organizzazioni Autonome Decentralizzate
TPS	Transazioni Per Secondo
MEV	Miner Extractable Value
UTXO	Unspent Transactions Output
dApp	Applicazioni decentralizzate

Part I

INTRODUZIONE

Questo lavoro ha due obiettivi principali che vengono raggiunti attraverso due filoni di ricerca: uno di natura industriale e uno scientifico. Il primo riguarda le difficoltà tecnico-scientifiche nella raccolta, tokenizzazione e generazione di valore dei dati durante l'esecuzione di un lavoro. Il secondo filone utilizza la Financial Computing per valutare gli asset attraverso metodi matematici, generare modelli di business e gestire il rischio. Il lavoro è suddiviso in quattro parti: la prima parte introduce i concetti di base delle tecnologie blockchain, alcuni cenni alle prospettive future sull'economia e agli strumenti finanziari, analizza infine la letteratura dei due filoni di ricerca e le tecnologie utilizzate per lo sviluppo. La seconda parte descrive il contributo apportato alla letteratura: la soluzione proposta per il tracciamento e la generazione di valore e il Token Evaluation System, un modello che utilizza la teoria della probabilità estesa per valutare asset digitali. La terza parte presenta una discussione sulla valutazione di strumenti finanziari immateriali, sulla scalabilità e sulla sicurezza della soluzione proposta. La quarta e ultima parte include riflessioni sull'elaborato e nuove opportunità di ricerca nell'ambito della trasformazione digitale.

1

INTRODUZIONE ALLA FINANCIAL COMPUTING E ALLA BLOCKCHAIN

Per mercato finanziario si intende quel sistema in cui qualunque partecipante può comprare o vendere strumenti finanziari tra i quali azioni, obbligazioni, fondi comuni di investimento (ETF), valute, derivati (futures, opzioni) ed un nuovo tipo di strumento finanziario basato sulla tecnologia blockchain: le monete digitali o piú comunemente, ma in maniera errata, criptovalute.

Di particolare interesse per questo elaborato sono queste ultime che, dal 2009, hanno reso possibile scambiare del valore su internet in maniera decentralizzata e peer-to-peer [54].

A tal proposito, il web3, il quale dopo diversi anni di supposizioni su cosa sarebbe stato, dall'anno 2021 é stato battezzato come l'internet basato sulla decentralizzazione e sullo scambio di valore e viene quindi chiamato internet del valore.

Possiamo suddividere gli asset basati su tecnologia blockchain nelle seguenti categorie:

Nota: Bitcoin è la rete, bitcoin è l'asset.

- **Coin**: la prima moneta digitale mai creata é bitcoin, ad opera dello pseudonimo Satoshi Nakamoto. Bitcoin consiste in un network basato su blockchain, dove puó essere scambiato valore attraverso transazioni Peer-to-Peer. Le coin sono quegli asset che si basano su una blockchain, sulla quale la moneta digitale viene utilizzata come incentivo per la messa in sicurezza del network e per il pagamento delle transazioni;

- **Token fungibili**: i token sono quegli asset che trovano vita attraverso uno Smart Contract su una blockchain abilitata agli stessi, un esempio possono essere tutti quei token presenti su rete Ethereum;

- **Non Fungible Token (NFT)**: anche in questo caso si tratta di asset implementati tramite Smart Contract, ma a differenza dei token fungibili presentati in precedenza, dove ogni token è uguale a un altro, gli NFT rappresentano una

collezione nella quale ogni token ha un ID diverso dagli altri; di conseguenza ogni token puó avere valore diverso rispetto a un altro della stessa collezione.

L'obiettivo di questo lavoro è proprio quello di focalizzarsi sulle monete digitali e sulle tokenizzazioni di beni del mondo reale in maniera da realizzare asset digitali basati sull'economia reale applicando poi le tecniche della Financial Computing al fine di effettuarne la valutazione, in un contesto dove questi saranno molto probabilmente protagonisti in futuro.

1.1 IL WEB3 E LA TECNOLOGIA BLOCKCHAIN

Dalla nascita della differenziazione fra web 1.0 e web 2.0, ci si é sempre chiesto cosa potesse essere il web 3.0. Partendo dall'inizio il web statico, o web 1.0, é uno spazio nella quale vengono indicizzati siti web i quali hanno interazione e contenuti limitati a immagini statiche, testo e hyperlink. Nel 1999 DiNucci inizió a parlare di web 2.0:

"The Web we know now, which loads into a browser window in essentially static screenfuls, is only an embryo of the Web to come. The first glimmerings of Web 2.0 are beginning to appear, and we are just starting to see how that embryo might develop. The Web will be understood not as screenfuls of text and graphics but as a transport mechanism, the ether through which interactivity happens. It will [...] appear on your computer screen, [...] on your TV set [...] your car dashboard [...] your cell phone [...] hand-held game machines [...] maybe even your microwave oven."

Il termine web 3.0 invece apparve per la prima volta in un articolo di Jeffrey Zelman nel 2006, il quale affermava:

"People keep asking what Web 3.0 is. I think maybe when you've got an overlay of Scalable Vector Graphics - everything rippling and folding and looking misty - on Web 2.0 and access to a semantic Web integrated across a huge space of data, you'll have access to an unbelievable data resource."

A seguito del fallimento di Lehman Brothers nel'ottobre del 2008 a cui ha fatto seguito la grande depressione del 2009, un anonimo inventore, il 1° Novembre 2008, pubblica un whitepaper sul sito metzdowd.com, su The Cryptography Mailing List [54].

Il whitepaper illustra e introduce una moneta elettronica decentralizzata e peer-to-peer, che l'autore immagina come contante elettronico (Electronic Cash [54]), basata su firme digitali che riutilizza alcuni concetti forniti in precedenza in diversi lavori, andando a risolvere il problema del double spending attraverso la Proof of Work (PoW) (Hashcash) [4], decentralizzando allo stesso tempo il concetto di documento a timestamping proposto da Haber e Stornetta nel 1991 [29].

Dopo diversi tentativi di creare una valuta completamente decentralizzata, fra cui b-money [16] di Wei Dai e Bit Gold di Nick Szabo, mai implementati, ma dai quali Satoshi ha estratto diversi concetti per la realizzazione di Bitcoin, (con molta probabilitá Szabo, Dai e Back, insieme ad Hal Finney e Gavin Andersen facevano parte del gruppo iniziale dietro allo pseudonimo Satoshi Nakamoto) la prima moneta elettronica peer-to-peer completamente decentralizzata vedeva la luce nel 2009 con la pubblicazione del client Bitcoin.

1.1.1 *Bitcoin*

Bitcoin é quindi la prima moneta digitale decentralizzata mai creata, nata quando Satoshi ha estratto il primo blocco (blocco genesi) il 3 Gennaio 2009 ricevendo in cambio i primi 50 Bitcoin (BTC) emessi. Nessun Bitcoin é stato preallocato a un indirizzo blockchain alla generazione del Genesis Block e, inizialmente, l'entitá che risolveva il problema computazionale relativo al blocco corrente, riceveva 50 BTC.

I Bitcoin estratti servono come incentivo alle entitá chiamate miner che dedicano potenza computazionale al fine di rendere sicuro il network ed evitare il double spending; questo nuovo tipo di computazioni è stato chiamato *Volunteer Computing*.

Gli incentivi vengono dimezzati ogni 210000 blocchi, i quali hanno una frequenza di circa 6 all'ora, di conseguenza questo evento avviene ogni 4 anni. La fornitura totale é di 21 milioni di pezzi, oltre il quale non verranno piú rilasciati come ricompensa

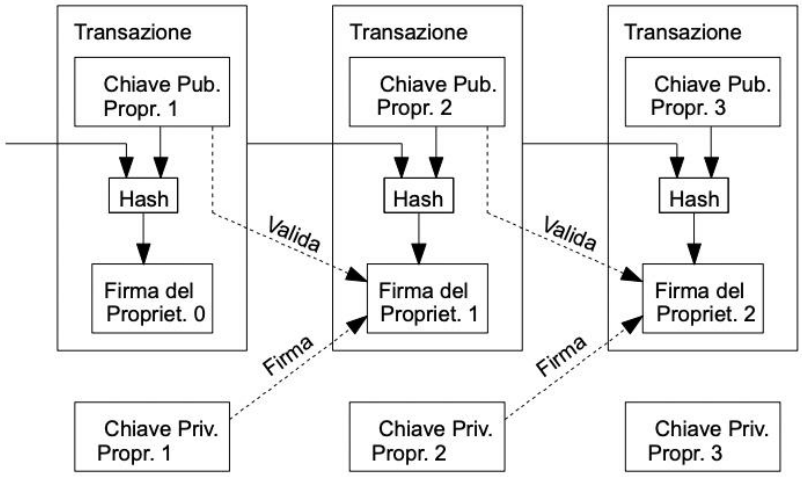

Figure 1.1: Dal whitepaper Bitcoin[54]: struttura delle catene di transazioni.

del blocco. Al raggiungimento di questo evento, l'unico incentivo per i miner saranno i costi di transazione ottenuti dalle operazioni dei partecipanti al network.

Questa tecnologia é basata sulla crittografia asimmetrica e sulle firme digitali. Alla creazione di un indirizzo Bitcoin, vengono generate una chiave pubblica e una relativa chiave privata; la chiave privata viene utilizzata per firmare le transazioni, mentre la pubblica per verificarle (Fig. 1.1).

A ogni estrazione del blocco, la prima transazione é sempre generata da un indirizzo speciale chiamato COINBASE (Fig. 1.2). Le transazioni Bitcoin sono gestite attraverso un modello contabile chiamato Unspent Transactions Output (UTXO). In questo modello, i saldi degli indirizzi vengono gestiti utilizzando gli input ed output delle transazioni, questo significa che nel momento in cui un miner estrae un blocco, riceve una UTXO pari all'incentivo attuale.

Utilizzando l'esempio in figura 1.2, Satoshi Nakamoto riceve una UTXO da 50 BTC dalla "COINBASE". Ipotizzando che Satoshi effettui delle operazioni e abbia a disposizione una UTXO di 10 BTC ed invii 4 BTC ad Hal Finney, dovrá essere generata una transazione con due output: 4 verso l'indirizzo di Hal e 6 verso il

Figure 1.2: La prima transazione di Bitcoin: 50 BTC dall'indirizzo "COINBASE" a Satoshi Nakamoto.

suo stesso indirizzo (TX 0 in 1.3), il quale rappresenta il "resto", e che sará quindi trattata come una nuova UTXO.

Le firme digitali e la crittografia peró non possono bastare da sole per evitare la doppia spesa e i beneficiari, avendo a disposizione soltanto un documento a timestamp, non possono in alcun modo definire se i BTC che hanno ricevuto sono stati giá spesi oppure no.

Una soluzione comunemente utilizzata é affidarsi a un' entitá fidata centrale che certifichi la bontá del documento, ma ovviamente ció non era nei piani dell'inventore di Bitcoin. L'obiettivo di Satoshi Nakamoto era di ottenere un sistema di transazioni Peer-to-Peer che non avesse alcun bisogno di entitá centrali su cui far affidamento per funzionare; a tal fine, Satoshi si serve di alcuni concetti come l'annuncio delle transazioni in un canale pubblico, sistema ipotizzato da Dai in [16], e la PoW messa in pratica da Alan Back in [4] per far sí che i nodi concordino sul fatto che la transazione sia la prima ricevuta e che non ci sia quindi doppia spesa.

Satoshi Nakamoto, in particolare, nella sezione 4 del suo whitepaper, illustra la PoW, la quale comporta una ricerca di un valore detto *nonce* per il quale l'hash del blocco ricercato abbia all'inizio un numero target di zero bit; questo numero target é chiamato *difficulty* e viene continuamente ricalibrata dal client in maniera tale da mantenere costante nel tempo il numero di blocchi prodotti.

Questo sistema di consenso funziona e risulta estremamente sicuro in quanto l'unico metodo per trovare il nonce adatto é effettuare un bruteforce iterando sul nonce e, una volta trovato e pubblicato il blocco, l'unico modo per modificarlo é rieseguire il lavoro. Al contrario la verifica é molto semplice e puó essere effettuata calcolando l'hash un'unica volta.

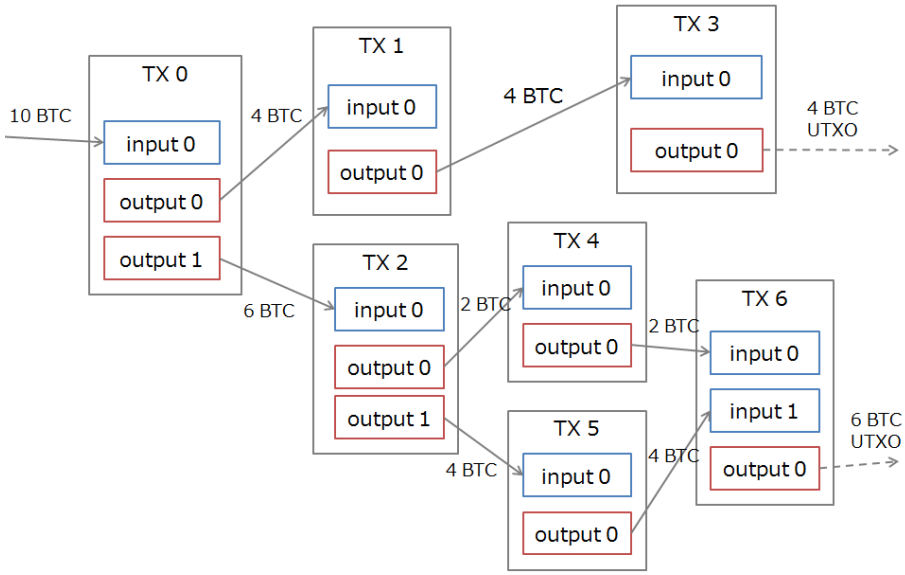

Figure 1.3: Gestione degli input ed output tramite UTXO: si nota come gli output siano al massimo 2 in quanto rappresentano l'ammontare trasferito ed il "resto", mentre gli input possono essere molteplici in quanto provenienti da diverse transazioni precedenti.

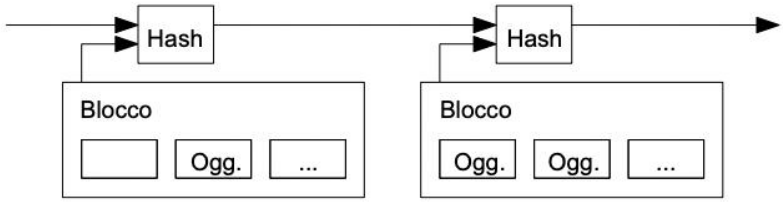

Figure 1.4: Catena di blocchi in Bitcoin.

1.1.2 *Smart Contract e Web3*

Prima del 2014, si parlava già di Smart Contract e di leggi basate sul paradigma *"the code is law"* e si immaginava un contesto dove il rispetto delle leggi fosse totalmente gestito da codice informatico.

Bitcoin nella sua implementazione ha un debole concetto di Smart Contract, ad esempio, si può costruire uno script che implementi la *multisignature* richiedendo firme da più chiavi private impiegate per validare una transazione.

Gli script possono anche essere utilizzati per pagare premi per la soluzione di problemi computazionali oppure è possibile costruire uno script che indichi che si può scambiare un bitcoin se si è in grado di provare che un'altra moneta digitale è stata inviata a un determinato indirizzo; consentendo essenzialmente un cambio tra monete digitali decentralizzato.

Tuttavia, il linguaggio di scripting, così come implementato nella rete Bitcoin, presenta alcune importanti limitazioni:

- mancanza di Turing-completezza del linguaggio di scripting: anche se esiste un grande sottoinsieme di istruzioni che il linguaggio di scripting di Bitcoin supporta, esso non offre molte possibilità. La categoria principale che viene meno è il ciclo. Questa limitazione è imposta al fine di evitare cicli infiniti durante la verifica delle transazioni i quali potrebbero generare non pochi problemi se utilizzati male volontariamente o involontariamente; teoricamente si tratta di un ostacolo sormontabile per i programmatori di script, in quanto qualsiasi ciclo può essere simulato semplicemente ripetendo il codice sottostante tante volte con un'istruzione condizionale, implicando però script inefficienti dal punto di vista dello storage;

- cecità del valore: non esiste alcun modo per le transazioni UTXO di fornire un controllo sul valore di bitcoin in un'altra moneta. Ad esempio, è impossibile implementare un contratto nella quale A e B possono depositare 1000$ in BTC che dopo 30 giorni smisti automaticamente 1000$ in BTC ad A e i profitti a B se il valore di BTC è cresciuto;

- mancanza di stato: UTXO può avere solo due stati: speso e non speso; non c'è possibilità per i contratti o gli script di mantenere altri stati al loro interno. Ciò rende difficile creare contratti con diversi stati ed Exchange Decentralizzati (DEX). Questo inoltre significa che UTXO può essere soltanto usato per costruire contratti semplici, ma è limitato nella creazione di contratti "stateful" più complessi come le Organizzazioni Autonome Decentralizzate (DAO) e meta-protocolli complessi. Lo stato binario, combinato con la cecità del valore, limita anche un'altra importante applicazione, l'interazione con le valute FIAT;

- cecità alla blockchain: UTXO è cieco alla blockchain. È quindi cieco agli hash dei blocchi. Questo pregiudica applicazioni nel campo del gioco d'azzardo e moltre altre categorie che fanno uso delle variabili casuali in quanto in questo ambito vengono utilizzati gli hash dei blocchi come *seed* per le variabili casuali, privando il linguaggio di scripting di pseudocasualità.

Al fine di superare questi problemi dello scripting di Bitcoin per lo sviluppo di Smart Contract, nel 2014, viene pubblicato il whitepaper di Ethereum da Vitalik Buterin [9]. Con Ethereum, viene costruita una piattaforma decentralizzata specializzata nello sviluppo di Smart Contract, che fornisce benefici ancora maggiori in termini di semplicità di sviluppo con client estremamente più leggeri, permettendo allo stesso tempo alle applicazioni decentralizzate di condividere l'ambiente economico e la sicurezza della blockchain.

Nota:Ethereum è la rete, Ether è l'asset.

Nel 2015 viene quindi effettivamente aggiunta una nuova rivoluzione alla già importante innovazione della tecnologia blockchain: il client di Ethereum, sviluppato da Vitalik Buterin in team con altre personalità conosciute oggi nel mondo dell'high-tech.

Ethereum [9] è una piattaforma web 3.0 decentralizzata per la creazione, la pubblicazione Peer-to-Peer e l'esecuzione automatica di contratti digitali, da cui la definizione di Smart Contract. Rappresentando così la prima implementazione della blockchain 2.0 laddove prima di Ethereum venivano lanciati per lo più progetti che pretendevano di sostituire Bitcoin cambiandone o

migliorandone qualche caratteristica. La vera innovazione di Ethereum consiste proprio negli Smart Contract.

Si tratta di algoritmi che hanno al loro interno elementi contrattuali tra persone, che venivano automaticamente eseguiti dai miners Ethereum[1] in modo distribuito, senza avvocati, notai o altri intermediari, ma solo per la volontà espressa dagli attori, con una chiara riduzione dei costi accessori e possibili controversie.

Grazie a Ethereum, si sono generati diversi trends di investimento e bolle speculative. Nei suoi primi anni di vita, è stato possibile inizializzare campagne di crowdfunding chiamate Initial Coin Offering (ICO), in cui gli sviluppatori mostravano la propria idea con un whitepaper agli investitori, i quali potevano contribuire economicamente al progetto in cambio di token creati ad hoc sulla blockchain Ethereum. Proprio quella delle ICO fu la prima bolla speculativa generata sull'ecosistema Ethereum in quanto i token non erano altro che codice Solidity standard ERC20 e gli investimenti venivano eseguiti senza regolamentazione e solo sulla base di idee. Col passare del tempo, questo tipo di operazione è andata scomparendo e con essa la fiducia degli investitori, a causa di molte persone che approfittavano del trend per raccogliere fondi e fuggire con i guadagni, lasciando gli investitori con token che non avevano alcun valore.

Nel 2020, sempre grazie agli Smart Contract, è esplosa una nuova tendenza chiamata Decentralized Finance (DeFi). Anche la DeFi agli inizi è risultata in una bolla, ma rispetto alle ICO dove tutto si basava solo su un'idea, nella DeFi si è sviluppato molto di più e le idee più solide sono sopravvissute.

In questo trend gli investitori possono investire in prodotti finanziari complessi simili a quelli della finanza tradizionale, ma con i vantaggi della tecnologia blockchain; alcuni casi d'uso sono: ricevere degli interessi sulle proprie monete digitali bloccate in un contratto, collateralizzare i propri averi al fine di prenderne in prestito altri o di generare *stablecoin*, scambiare le proprie monete su DEX, utilizzare prodotti derivati, ecc.

1 al tempo della scrittura si è passato alla Proof of Stake (PoS) e non sono più i miner ad eseguire gli Smart Contract e la validazione delle transazioni, ma i nodi validatori.

Nel 2021 invece si è formata la bolla degli NFT, i quali sono equiparabili a collezionabili digitali che mantengono al loro interno dei metadati.

Ethereum e gli Smart Contract quindi sono spesso protagonisti nelle innovazioni le quali, grazie alla elevata complessità della tecnologia, alla facilità di investimento e, conseguentemente, alla scarsa consapevolezza degli investitori (spesso improvvisati), sono causa della formazione di bolle finanziarie.

A ogni bolla finanziaria scoppiata, però, le innovazioni robuste sopravvivono e ricostruiscono il settore.

Tutto questo sembra essere solo l'inizio di una tecnologia che è ancora immatura e non ha ancora strumenti perfetti anche dal punto di vista dello sviluppo software, della sicurezza e degli strumenti di valutazione; tutto ciò è quindi oggetto del presente elaborato.

1.1.3 *Approfondimento sul funzionamento di Ethereum*

Al fine di comprendere alcune definizioni presenti in questo lavoro, è necessario prima di tutto comprendere il funzionamento della piattaforma Ethereum.

Ethereum si definisce una piattaforma web 3.0 decentralizzata per la creazione, la pubblicazione Peer-to-Peer e l'esecuzione automatica di contratti, da cui la definizione di Smart Contract.

Gli Smart Contract contengono elementi contrattuali tra persone, che vengono automaticamente eseguiti da codice, senza intermediari e possibili controversie.

Il concetto base dietro gli Smart Contract è semplice, un contratto è un frammento di codice scritto nel linguaggio di programmazione Solidity, il quale fino a settembre 2022 veniva eseguito dai miner in maniera distribuita (Figura 1.5) tramite PoW. Dal 2022, Ethereum ha cambiato il proprio metodo di consenso passando dalla PoW alla PoS [2] e i miner sono stati sostituiti dai validatori.

È evidente la semplicità di una logica sottostante di questo tipo e allo stesso tempo le ampie potenzialità in termini di complessità, generalizzazione, uso e articolazione. Ciò ha permesso alle

2 Metodo di consenso in cui i creatori del blocco, chiamati validatori, vengono scelti in maniera casuale mettendo in gioco le proprie dotazioni di Ether (stake).

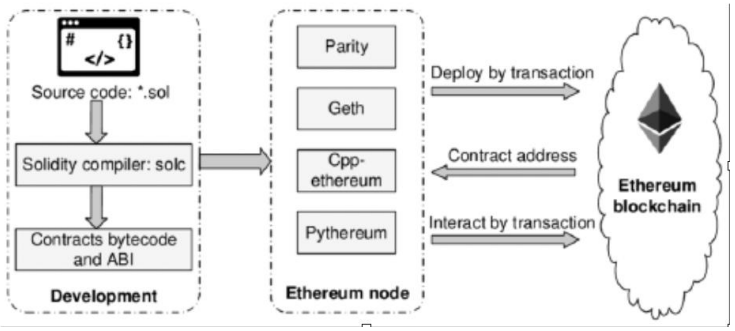

Figure 1.5: Struttura di uno Smart Contract Ethereum.

DAO di emergere; queste ultime sono Smart Contract a lungo termine che gestiscono le risorse e codificano lo statuto di un'intera organizzazione.

Uno Smart Contract corrisponde a un insieme di stati, una sorta di semaforo con più di tre elementi preordinati, nel quale, prima dell'inizio del contratto, tutti gli stati sono inizializzati a "false"; durante l'esecuzione alcuni stati diventano "true", fornendo così l'attuale livello di esecuzione in cui si trova lo Smart Contract.

Al completamento, tutti gli stati saranno "true". Da questo punto di vista, è chiaro che uno Smart Contract è un sistema dinamico e la sua esecuzione è una funzione di transizione tra stati predefiniti (gli elementi fondamentali del contratto) all'interno del sistema di transizioni di stato.

Le applicazioni e gli usi che è possibile costruire sulla base concettuale appena descritta sono praticamente infiniti, ma come descritto nel whitepaper di Ethereum [9], possono essere classificati in tre categorie principali:

- applicazioni finanziarie (altre monete digitali, derivati finanziari, contratti di hedging, gestione patrimoniale, gestione degli investimenti, ecc.);
- applicazioni con ripercussioni economiche indirette (gestione di beni di qualsiasi tipo che, attraverso attività reali, creano valore che può essere poi monetizzato);
- applicazioni non finanziarie (voto, governo, gestione, sociale, identità, reputazione, ecc.).

Questo tipo di applicazioni decentralizzate, rispetto alla controparte centralizzata, eredita tutti i vantaggi relativi alla blockchain permettendo di automatizzare, certificare, rendere più trasparente, aumentare la sicurezza e generare valore.

Il motore dell'esecuzione delle transazioni Ethereum è il GAS. Fino al 2021 era la commissione che ci si impegnava a pagare ai miner (o, in questo momento, ai validatori) per l'esecuzione di ogni operazione in una transazione.

Il GAS viene pagato in Ether (ETH), la moneta digitale di Ethereum. Dal 2021 l'aggiornamento London "brucia"[3] una parte di Ether utilizzati per il pagamento della transazione, rendendo Ether deflattivo durante gli utilizzi intensivi del network.

Una unità di GAS equivale all'esecuzione di un'operazione bytecode. Un operatore è in grado di aumentare la priorità con cui la propria transazione può essere validata aumentando il prezzo del GAS e di conseguenza la velocità con cui si esegue l'operazione. Il prezzo del GAS viene quindi definito GASPRICE. Questo meccanismo a oggi in Ethereum si è rivelato controproducente durante le fasi di congestione in quanto favorisce i possessori di grandi quantità di ETH che non hanno problemi ad aumentare il costo del GAS per transazioni più veloci, a discapito dei piccoli operatori che si trovano a pagare l'equivalente di decine di dollari solo per trasferire le proprie monete. In questo modo, si va a generare un'asta al rialzo sul prezzo del GAS.

Ethereum sta cercando di superare il problema della scalabilità e dei costi di transazione molto alti attraverso i layer 2, sidechain che lavorano in parallelo a Ethereum e che attraverso alcune tecniche ne ereditano la sicurezza[4]. L'architettura ad asta porta a un altro problema: quello del frontrunning, descritto in dettaglio nel capitolo 8.

Al fine di evitare esecuzioni infinite, che siano esse malevole o accidentali, si definisce il limite di GAS che può essere utilizzato in una transazione. Esso viene definito "GASLIMIT" o "STARTGAS".

3 eliminare dal circolante

4 Le attuali tecniche di scaling sono basate su Optimistic rollups e Zero-Knowledge rollups

Dal whitepaper [9] è possibile analizzare in dettaglio la funzione di transizione di stato di Ethereum. Essa può essere definita in passi come segue:

1. si controlla se la transazione è ben impostata (cioè che abbia il giusto numero di valori), che la firma sia valida e il nonce corrisponda a quello dell'account del mittente. In caso contrario, si emette un errore;

2. si calcola la commissione di transazione come STARTGAS * GASPRICE e si determina l'indirizzo del mittente dalla firma. Viene sottratta la commissione dal bilancio dell'account del mittente e se ne incrementa il nonce. Se il bilancio non è sufficiente, viene restituito un errore;

3. si inizializza GAS = STARTGAS e si sottrae una certa quantità di GAS per i byte della transazione;

4. si trasferisce il valore della transazione dall'account del mittente all'account del destinatario. Se l'account del destinatario non esiste, viene creato. Se l'account del destinatario è un contratto, si esegue il codice del contratto fino al completamento o fino a quando l'esecuzione non termina il GAS;

5. se il trasferimento fallisce perché il mittente non possiede abbastanza ETH, il codice di esecuzione termina il GAS, tutti i cambiamenti di stato vengono ripristinati, ad eccezione del pagamento della commissione la quale viene pagata ugualmente al validatore;

6. altrimenti, se la transazione va a buon fine, vengono rimborsate al mittente le commissioni per il GAS rimanente e le commissioni utilizzate per il GAS vengono pagate al validatore.

Il codice nei contratti Ethereum è scritto in un linguaggio di basso livello, cioè il linguaggio bytecode a cascata, ed è denominato "codice EVM". Il codice consiste in una serie di byte, ognuno rappresentante un'operazione.

In generale, l'esecuzione del codice è un loop infinito che consiste nell'eseguire ripetutamente l'operazione indicata dal

Program Counter attuale (PC, il quale inizia da zero), incrementandolo di uno fino a che non si raggiunge la fine del codice, un errore, uno "STOP", o se è rilevata l'istruzione "RETURN".

Le operazioni hanno accesso a tre tipi di spazio nel quale registrare i dati:

- lo stack, una struttura dati di tipo last-in-first-out, che è un contenitore nel quale i valori possono essere inseriti o rimossi;
- la memoria, un array di byte espandibile all'infinito;
- lo storage a lungo termine del contratto, si tratta di una memoria a modello chiave/valore.

A differenza dello stack e della memoria dove i valori si resettano dopo la fine del calcolo, lo storage è persistente.

Il codice può anche accedere al valore, all'indirizzo del mittente e ai dati del messaggio in arrivo, così come ai dati dell'header del blocco, superando i limiti dello scripting di Bitcoin. Esso può anche restituire un array di dati come output.

Mentre la EVM lavora, tutto il suo stato computazionale può essere definito dall'insieme di dati (stato del blocco, transazione, messaggio, codice, memoria, stack, PC, gas), dove lo stato del blocco è lo stato globale che contiene tutti gli account e include i bilanci e lo storage.

All'inizio di ogni turno di esecuzione, l'istruzione corrente viene recuperata attraverso l'i-esimo (con i=PC) byte del codice (o 0 se PC è maggiore o uguale alla lunghezza del codice); ogni istruzione ha la propria definizione di come interagisce con l'insieme di dati. Ad esempio, "ADD" recupera due oggetti dallo stack e ne inserisce la somma risultante, riduce GAS di 1 e incrementa PC di 1. "SSTORE" recupera due oggetti dalla cima dello stack e inserisce il secondo oggetto nello storage del contratto all'indice specificato dal primo oggetto.

La blockchain di Ethereum è in molti versi simile a quella di Bitcoin, seppur con qualche differenza. La differenza principale tra la blockchain Ethereum e quella di Bitcoin è nell'architettura. Contrariamente a quella di Bitcoin, i blocchi di Ethereum contengono una copia sia dell'elenco delle transazioni sia dello stato

più recente. Inoltre, il numero di blocco e la difficoltà vengono memorizzati nel blocco stesso.

Descriviamo in dettaglio la validazione di un blocco Ethereum. Sia $APPLY(S, Tx) \rightarrow S'$ una funzione con input uno stato S ed una transazione Tx e output un nuovo stato S', l'algoritmo di validazione del blocco alla base in Ethereum funziona nel seguente modo:

1. controlla che il precedente blocco di riferimento esista e che sia valido;

2. controlla che la marcatura temporale del blocco sia più grande di quella del blocco di riferimento precedente e che la differenza sia inferiore a 15 minuti;

3. controlla che numero del blocco, difficoltà, origine della transazione, transazione derivata e GASLIMIT siano validi;

4. prima del 2022 si controllava che la PoW del blocco fosse valida, dal passaggio a PoS viene selezionato casualmente un numero di validatori che controlla la validità del blocco proposto dal validatore scelto;

5. sia $S[0]$ lo stato alla fine del blocco precedente; sia Tx la lista delle transazioni del blocco, con n transazioni; per tutte le transazioni Tx[i] con $0 \leq i \leq n-1, S[i+1] = APPLY(S[i], Tx[i])$.

6. Se qualsiasi $APPLY(S, Tx) \rightarrow S'$ restituisce un errore o se il GAS totale è consumato nel blocco fino a che eccede il GASLIMIT, viene restituito un errore.

7. Sia S FINAL = S[n] lo stato finale, il quale comprende la ricompensa per il blocco pagata al validatore, l'algoritmo controlla che la radice del Merkle tree dello stato S FINAL sia uguale alla radice dello stato finale S fornito nell'intestazione del blocco. In tal caso, il blocco è valido; in caso contrario, non lo è.

Un quesito comune che ci si pone frequentemente è "dove" il codice del contratto venga eseguito, in termini di hardware fisico: il processo di esecuzione del codice del contratto è una parte

della definizione della funzione di transizione di stato che, a sua volta, è una parte dell'algoritmo di validazione del blocco, così che, se una transazione viene aggiunta al blocco B, l'esecuzione del codice generata da quest'ultima sarà eseguita da tutti i nodi che scaricheranno e valideranno il blocco B.

1.1.4 *Token Fungibili e Non Fungibili*

Un'ultima definizione tecnica è necessaria al fine della comprensione del presente elaborato ed è quella del concetto di token. Un token è una moneta digitale che viene programmata su una blockchain pre-esistente; è possibile quindi emettere dei token attraverso gli Smart Contract.

Esistono due tipi di standard per i token, i quali possono venire estesi[5].

Standard ERC20, token fungibili

Nel 2015 venne proposto uno standard per l'implementazione di token su rete Ethereum. Questo standard è stato battezzato ERC20 (Ethereum Request for Comment). Lo standard ERC20 è stato il catalizzatore delle ICO siccome gli "sviluppatori" potevano utilizzare lo standard per creare un token e affiancargli un whitepaper. Per appartenere a questo standard, un token deve implementare alcune funzioni che permettono alle altre applicazioni di riconoscerlo. Le funzioni sono le seguenti:

- *totalSupply()*: restituisce il numero di token coniati fino al momento dell'invocazione della funzione;
- *balanceOf(address owner)*: restituisce il numero di token che l'account possiede;
- *transfer(address to, uint256 amount)*: trasferisce il numero di token passato come parametro (variabile amount), dall'account che richiama la funzione all'indirizzo passato come parametro (variabile to);

5 I token standard possono essere modificati per implementare funzioni di governance, di tokenomics interne come il burn nelle transazioni o la ridistribuzione, ecc.

- *approve(address spender, uint256 amount)*: permette a un account terzo, passato come parametro (variabile spender) di utilizzare un numero di token (variabile amount) per conto dell'utilizzatore della funzione. Questa funzione è utile nel caso in cui un contratto abbia bisogno di effettuare operazioni con i token dell'indirizzo che lo utilizza. Lo spender quindi, dovrebbe essere una risorsa o un agente di cui ci si fida in quanto può utilizzare le monete digitali del wallet che effettua l'approve;
- *transferFrom(address from, address to, uint256 amount)*: permette di inviare i token da un indirizzo a un altro, specificati nei parametri. Questa funzione può essere utilizzata solo se l'indirizzo "from" ha precedentemente effettuato un approve in favore dell'account che sta richiamando questa funzione.

Oltre alle citate funzioni, ne possono essere implementate altre personalizzate a seconda dei bisogni dello sviluppatore e dei requisiti del progetto. Le funzioni più utilizzate e non presenti nello standard sono quelle di *mint* e *burn*, le quali permettono di coniare nuove monete o bruciare una parte di quelle in circolazione (solitamente solo dal proprio account).

Standard ERC721, token non fungibili (NFT)

Al fine di ottenere beni digitali unici, nasce un altro tipo di token, l'ERC721. Proposto nel 2018, in questo standard viene coniato un token alla volta, fornendogli un numero identificativo (ID). In questo modo, ogni token diventa unico e ha caratteristiche differenti dagli altri. Un generico caso d'uso è l'implementazione di token che rappresentano oggetti collezionabili. Ogni oggetto ha una rarità propria e determinate caratteristiche.

Siccome conservare dati molto pesanti in termini di byte in blockchain è molto costoso, è di uso comune conservare i dati delle caratteristiche del token su una struttura dati diversa dalla blockchain e mantenere in essa solo un riferimento a questi dati. Una tecnologia che si sposa bene con questo approccio è l'InterPlanetary File System (IPFS), la quale consente di effettuare

lo storage di dati multimediali su nodi decentralizzati; quando si effettua lo storage dei dati di un file, IPFS ne calcola l'hash, il quale viene utilizzato come riferimento. Nel caso degli ERC721, è di comune usanza inserire il riferimento (l'hash) al file in blockchain.

1.2 FINANCIAL COMPUTING NELLA FINANZA DECENTRALIZZATA

Nel momento in cui Satoshi Nakamoto concepisce il whitepaper di Bitcoin cambia per sempre la finanza introducendo in questo mondo un nuovo tipo di asset, ancora oggi impossibile da catalogare in maniera chiara a causa delle sue peculiari caratteristiche.

Bitcoin é un asset decentralizzato ed estratto da entitá chiamate miner, ma creato da una o piú persone, per questo motivo si fa fatica a catalogarlo nella finanza tradizionale; a oggi l'asset sembra essere stato classificato dalla Securities and Exchange Commission (SEC) e dalla Commodity Futures Trading Commission come una commodity, in quanto fungibile ed estraibile in maniera similare al processo di estrazione dell'oro.

Satoshi Nakamoto, nel 2009, potrebbe inoltre aver cambiato per sempre anche l'economia; proprio questo ultimo cambiamento é oggetto del presente lavoro, il quale si pone l'obiettivo di fornire alla letteratura uno strumento utilizzabile nel presente a partire dalla generazione di royalties utilizzando poi la Financial Computing per la valutazione delle stesse e dando allo stesso tempo uno sguardo al futuro immaginando una gestione dei pagamenti e delle leggi in maniera totalmente automatica.

La Financial Computing é quella branca della scienza che si colloca fra l'informatica e la finanza e che permette la valutazione degli investimenti attraverso algoritmi, in maniera tale da calcolare il rischio e il potenziale rendimento di strumenti finanziari.

Questo lavoro è quindi suddiviso nel seguente modo: dopo aver mostrato in questo capitolo alcune definizioni utili alla comprensione del presente elaborato, nel capitolo 2 verranno analizzati i lavori presenti allo stato dell'arte che si collocano negli stessi settori trattati da questo lavoro.

Nel capitolo 3 verranno illustrate le tecnologie abilitanti allo sviluppo dei modelli definiti e alla validazione degli stessi.

Nel capitolo 4 è presente una Proof of Concept (PoC) per la rendicontazione e la tokenizzazione del lavoro attraverso la generazione di token di royalties.

Nel capitolo 5 verrà introdotta la teoria della Plausibilità la quale è alla base del modello di valutazione dei token presentato in questo lavoro.

Nel capitolo 6 viene introdotto il Token Evaluation System (TES), un sistema di supporto alle decisioni che permette di stimare il prezzo degli asset digitali attraverso la filosofia del value investing [27], il modello del TES viene poi validato nel capitolo 7 utilizzando il machine learning su dati reali.

Nel capitolo 8 viene discussa la sicurezza dell'approccio mostrato nel capitolo 4 e viene eseguita una proiezione dei costi di transazione delle diverse piattaforme Smart Contract presenti sul mercato, dimostrando che la PoC è applicabile a un contesto reale.

Infine nel capitolo 9 vengono discusse le implicazioni e gli scenari aperti da questo elaborato, in termini di ricerca e di sviluppi futuri.

2

STATO DELL'ARTE

Questo lavoro ha l'obiettivo di documentare completamente il ciclo di vita del processo chiamato tokenizzazione degli asset il quale, con presunzione, verrà utilizzata massicciamente in ambito industriale. Per questo motivo, viene analizzato lo stato dell'arte a partire dalla produzione di un asset tokenizzato, fino alla sua valutazione eseguita attraverso un modello basato sulla filosofia del value investing [27].

Trattandosi di una tecnologia prototipata solo nel 2009 e che ha visto lo sviluppo di Smart Contract nel 2015, la letteratura riguardante la produzione e la valutazione di asset digitali è molto scarna.

L'utilizzo della tecnologia blockchain, insieme all' Internet of Things (IoT), è il fondamento dell'industria 4.0, quindi esistono diversi lavori interessanti allo stato dell'arte. In [7] sono analizzate tutte le barriere e i problemi dell'utilizzo della blockchain in ambito industriale. La review è dell'anno 2019, nella quale la piattaforma Ethereum aveva il monopolio per l'esecuzione di Smart Contract. Le barriere presentate in questo lavoro sono:

- questioni organizzative e regolamentative: la finalità della transazione, la legalità delle parti interessate e le strategie di prezzo;
- questioni relative all'ambiente competitivo ed economico: il comportamento di nuovi attori, operatori storici e consumatori;
- problemi di progettazione tecnologica: supporto alla piattaforma software, progettazione in ambiente distribuito;
- problemi di scalabilità.

Se i problemi regolamentativi sono ancora una questione aperta e vanno al di la degli obiettivi di questo elaborato, la soluzione proposta più che tracciare la catena produttiva, elabora

il comportamento del lavoratore e ne gestisce le royalties ottenendo un sistema altamente generalizzato e scalabile. I problemi di scalabilità verranno risolti adottando delle nuove piattaforme blockchain EVM compatibili che saranno analizzate nel capitolo 8.

Le tokenizzazioni vanno di pari passo con la tracciabilità e le problematiche risultano pressocchè simili. In [63] è descritto lo stato dell'arte delle tokenizzazioni mostrando qualche esempio presente in letteratura. Nella stessa review sono inoltre descritte le challanges che risultano:

- regolamentative: in particolare il lavoro cita la mancanza di regolamentazione degli asset basati su blockchain;
- mancanza di business partners;
- compliance: il paper cita la mancanza e l'alto costo di framework per la compliance adattati a questo tipo di asset;
- sicurezza: fino al 2020, dagli Smart Contract presenti a mercato sono stati prelevati illecitamente asset per più di 2 miliardi di dollari in controvalore (al momento della scrittura la cifra ammonta a 5.93 miliardi di dollari)

2.1 SICUREZZA

Quando si applica la tecnologia blockchain al campo dell'industria 4.0, la sicurezza è molto importante, soprattutto quando si utilizzano sensori IoT [50] i quali sono poco affidabili; Stifter et al. in [68] forniscono una panoramica della ricerca sulla sicurezza blockchain, delineando alcune progettazioni di sistemi che potrebbero presentare vulnerabilità e fornendo esempi di proposte potenzialmente insicure nel campo dei sistemi cyberfisici (CPS). In particolare, in questo lavoro vengono mostrate le vulnerabilità di un approccio di monitoraggio basato sulla PoW; nel caso della nostra PoC, i problemi a livello di consenso sono scalati dall'utilizzo di una blockchain pubblica già esistente.

In [77], gli autori propongono un sitema basato su escrow per lo sviluppo collaborativo di dispositivi elettronici. In questo lavoro viene delineato come un produttore può utilizzare una

blockchain per presentare offerte per la progettazione e l'implementazione di dispositivi elettronici. Gli ingegneri che accettano il compito e risolvono il problema in modo collaborativo vengono pagati automaticamente sbloccando le monete digitali se il progetto viene implementato e verificato da tutti gli ingegneri.

Sulla base delle informazioni disponibili, la collaborazione di ingegneri malintenzionati potrebbe portare alla verifica di design errati, rendendo un ingegnere fraudolento in grado di ricevere il successivo pagamento.

Un approcco per la quale questo tipo di problema potrebbe essere risolto, è l'utilizzo di accessi con identità biometriche; disincentivando gli utenti a compiere azioni fraudolente (capitolo 7).

Zivic in [83] analizza l'impatto delle Distributed Ledger Technologies (DLT) nell'industria automobilistica 4.0. Yizhi et al. in [48] usano la blockchain per migliorare la sicurezza dei dispositivi IoT. Wang et al. in [73], propongono uno schema sicuro basato su blockchain per la condivisione di Private Charging Pile (PCP).

Un'altra applicazione delle tecnologie blockchain nel campo dell'IoT e dell'industria 4.0 è proposta da Zaidi et al. in [80]. In questo lavoro, gli autori propongono un modello di controllo degli accessi per dispositivi IoT basato su attributi, registrati in una blockchain, evitando così il tempering dei dati ed eliminando il single point of failure nei dispositivi di edge computing.

2.2 PRIVACY

Oltre alla sicurezza, quando si implementa una soluzione blockchain in ambiente industriale, è importante la privacy [82], in quanto una delle principali caratteristiche dei DLT è la trasparenza e, per questo motivo, il segreto industriale potrebbe essere a rischio.

Uno dei più grandi problemi tecnico-scientifici dell'applicazione della tecnologia blockchain alla tracciabilità nell'industria, è la dualità fra trasparenza e mantenimento del segreto industriale. Una blockchain privata potrebbe favorire quest'ultimo, a discapito della trasparenza. Mentre una blockchain pubblica permette di rendere il processo completamente trasparente.

In [82] vengono valutati alcuni approcci interessanti per la computazione di dati con Smart Contract in un ambiente privacy-oriented. In particolare vengono illustrati alcuni strumenti crittografici che permettono di manipolare dati senza esporli:

- crittografia omomorfica;
- crittografia attribute-based;
- secure multi-party computation basata su zero-knowledge proofs.

Nella PoC mostrata in questo elaborato per semplicità non sono state utilizzate tecniche per l'offuscamento dei dati elaborati, ma è di sicuro interesse negli sviluppi futuri applicarle in quanto è necessario mantenere la privacy dei lavoratori e delle aziende.

2.3 PAGAMENTI AUTOMATICI

Un'altra applicazione nel campo dell'industria 4.0 viene proposta da Wijaya et al. in [76], dove gli autori sfruttano la tecnologia blockchain per sviluppare un sistema che consente ai contribuenti di pagare l'imposta di bollo sui propri documenti elettronici. Altri lavori interessanti nel campo dei pagamenti automatici e della tracciabilità del lavoro sono [5, 11, 40, 56, 78]. Fra questi, il lavoro che più si avvicina a quello proposto in questo elaborato è [56], nella quale Pinna e Ibba propongono un sistema basato su Smart Contract che permette di regolare il rapporto fra lavoratori occasionali e aziende, gestendo automaticamente i pagamenti e proteggendo allo stesso tempo il lavoratore in caso di insolvenza.

Questo lavoro però presenta diverse limitazioni fra le quali le più importanti sono: la gestione della sicurezza degli utenti, il problema regolamentativo dei pagamenti automatici tramite monete digitali e la mancanza di indennizzi in caso di eventi avversi. Infine, gli autori non indicano su quale piattaforma hanno effettuato la distribuzione degli Smart Contract creati.

2.4 VALUTAZIONE DI ASSET DIGITALI

La seconda macro-area nella quale si colloca il presente elaborato è quella della valutazione di asset digitali.

In letteratura troviamo diversi lavori che analizzano il processo di valutazione che l'uomo effettua su oggetti materiali e immateriali.

Li e Camerer in [47] mostrano che il senso più utilizzato dalle persone nella valutazione di un oggetto materiale (nel caso di questo studio gli oggetti sono frutti) è la vista. In questo articolo, gli autori hanno eseguito un esperimento che mostra come si commetta errori quando si valuta oggetti utilizzando solo le caratteristiche visive.

Seppure esistano molte opere che trattano l'applicazione della tecnologia blockchain all'industria 4.0, allo stato dell'arte nessuno ha mai valutato il valore economico delle attività umane e di conseguenza degli asset digitali emessi sulla base di una valuechain. Anche se non esistono al momento questo tipo di valutazioni, in letteratura troviamo sperimentazioni riguardanti la valutazione delle risorse immateriali nell'economia tradizionale come la conoscenza, l'informazione, la proprietà intellettuale e l'esperienza. Inoltre le attività umane sono strettamente legate ai risultati microeconomici.

In particolare, Kendrick in [42] stima il valore del lavoro umano attraverso un indice, la produttività, indicando come input la misura in termini di occupazione, ore di lavoro oppure ore di lavoro ponderate per la retribuzione oraria media e come output il valore del prodotto. Sia gli input che gli output sono a sua volta correlati alla classificazione della relativa industria. Clemhout in [12] scrive una nota per correggere alcune imperfezioni nell'opera di Kendrick che rivede le sue teorie in [43]. L'indice di Kendrick è utilizzato in microeconomia al fine di valutare l'andamento del proprio business e di conoscere la collocazione all'interno del mercato di un'impresa.

Entrando più a fondo, troviamo diversi lavori che utilizzano il machine learning per la valutazione di asset intangibili:

Li et al. [46] hanno modellato una Artificial Neural Network (ANN) basata su otto caratteristiche che valuta asset bancari.

Il Tobin's Q ratio è un indice che permette di valutare un'azienda attraverso gli asset che ha a disposizione e si definisce come:

$$Q = M/A$$

dove M rappresenta il valore a mercato dell'impresa e A il quello degli asset. Un Q ratio > 1 indica che un'azienda è sopravvalutata, mentre $Q < 1$ che è sottovalutata.

A causa della mancanza di regolamentazione sui beni intangibili, questi possono presentare pesanti asimmetrie sulle valutazioni.

I beni immateriali possono essere organizzati in gerarchie con una differente valutazione, in particolare possono essere suddivisi in capitale intellettuale, capitale strutturale, capitale di innovazione e capitale di relazione [53].

Diversi studi [3, 30] indicano che, quando $Q > 1$, il valore in eccesso è dato dai beni immateriali.

Sulla base di queste presunzioni, gli autori in [51, 70] analizzano le prestazioni di diversi classificatori su un set di dati di molteplici industrie utilizzando come input il Q value. In particolare il classificatore costruito indica se le aziende prese in esame hanno alti livelli di asset intangibili o meno. I loro risultati mostrano che la migliore classificazione delle industrie basata sui beni intangibili è ottenuta attraverso un classificatore ibrido basato su k-means e classificatori basati su boosting/bagging.

In letteratura, al fine di valutare asset intangibili, oltre al machine learning vengono utilizzati i modelli competitivi. Ali et al. in [1], mostrano come in una negoziazione di informazioni, l'intermediario può massimizzare le sue entrate attraverso la teoria dei giochi, supponendo che conosca il prezzo esatto del bene. Un altro uso dei modelli competitivi in economia è mostrato in [55] di Olea et al. dove vengono utilizzati per prevedere la proliferazione di "fattori" che spiegano la variazione trasversale dei rendimenti azionari attesi. In altre parole, gli autori di [55] immaginano la negoziazione di asset produttivi come un modello competitivo dove ogni offerente ha una percezione, emotività ed esperienza diversa e la funzione di payoff del nuovo proprietario dell'asset sarà

$$P = M - (a - y)^2$$

dove M è una quantità positiva conosciuta, a è l'azione che il nuovo proprietario sceglie di compiere e y è una variabile casuale. In questo caso, il valore dell'asset è dato dalla capacità degli attori di prevedere y.

Chiaramente, y dipende da diversi fattori che ricadono nel contesto delle decisioni in mancanza di informazioni.

In [28], gli autori riassumono come nella finanza tradizionale i fattori più influenti per la valutazione di un bene siano: la richiesta da parte di chi compra, la valutazione di chi vende, la valutazione di un supervisore, la politica nazionale, la politica economica e l'ambiente informativo.

Proprio l'ambiente informativo e le decisioni in contesti di info-incertezza e info-incompletezza è oggetto del presente elaborato e della teoria presentata in [35], la quale è ripresa nel capitolo 5.

Infine, nel campo degli asset decentralizzati, Lo e Medda [49] hanno condotto uno studio empirico analizzando 86 tokenomics blockchain nel 2018. Tale studio risale al periodo delle ICO. Gli autori hanno analizzato le funzioni, le caratteristiche e la distribuzione dei token, utilizzando queste informazioni per dare evidenza a quattro ipotesi. Nello studio risulta che il valore dei token nel periodo delle ICO era correlato a:

- prezzo di bitcoin ed Ether;
- capitale degli insiders;
- business model;
- tipo di token.

Le quattro ipotesi nulle sulla quale si basa il lavoro sono:

- H1: Le variazioni di prezzo in Bitcoin ed Ethereum causano cambiamenti di prezzo nei token ICO;
- H2: Il tipo di token influisce sulla relazione fra il modello di business sottostante e il prezzo di mercato;
- H3: Gli Utility Token che sono l'unico mezzo di scambio nella relativa piattaforma, scambiano a un prezzo più alto;
- H4: Le ICO con una percentuale maggiore di token in circolazione riservati all'emittente e una percentuale inferiore di token riservati al mining, vengono scambiati a un prezzo più alto.

Attraverso l'analisi effettuata sugli 88 token presi in esame, gli autori riescono a dare evidenza ad H1 e H2.

L'interesse di questo elaborato (oltre a quello di fornire un metodo per la creazione di token basati sul lavoro umano) non è tanto quello di prevedere il prezzo di un asset, quanto di mostrare un metodo per la valutazione del prezzo equo di asset digitali basati su attività[1], che fornisca un sistema di supporto alle decisioni il quale permetta di gestire al meglio il portafoglio.

La filosofia è la stessa del value investing o, semplificando, quella di avere uno strumento come l'indice P/E (Price/Earnings) nell'equity[2] o il già citato Tobin's Q ratio. Viene successivamente dimostrato come tale approccio, a differenza del Value Investing, permette di ottenere livelli di prezzo target in mercati direzionali particolarmente ottimisti o depressi generando un sistema di supporto alle decisioni in grado di offire all'investitore un'opinione più equilibrata rispetto al sentiment di mercato.

2.5 TRACCIABILITÀ

La tracciabilità del processo produttivo è ampiamente analizzata in letteratura [69]. Nella review, per prima cosa, gli autori descrivono la dualità fra tracciabilità e trasparenza, i quali termini vengono spesso erroneamente intercambiati. In particolare, mentre la trasparenza permette di ottenere tutte le informazioni dettagliatamente, nella tracciabilità di un prodotto o di un processo produttivo è estremamente delicata la convivenza della dualità fra trasparenza e mantenimanto del segreto industriale e per questo motivo essa è oggetto di approfondita ricerca.

Prima della nascita degli Smart Contract, sono stati proposti diversi sistemi per l'implementazione della tracciabilità. In [64] viene proposta una strategia per il monitoraggio della qualità della carne. Mentre una soluzione più generalizzata viene proposta in [41]. Questo tipo di soluzioni sono centralizzate e soffrono di alta manipolabilità. Per questo motivo, alla nascita di Ethereum e degli Smart Contract la ricerca si è spostata verso

1 ma anche non basati su attività lavorative

2 L'indice P/E non prevede quanto sarà il prezzo di un asset fra n anni, ma indica quanto gli investitori sono disposti a pagare in più rispetto al valore degli utili dell'azienda sottostante

soluzioni più decentralizzate che, grazie alle caratteristiche della blockchain, permettono di ottenere dati certificati e non falsificabili.

In letteratura, ogni soluzione deve essere verticale verso il campo di applicazione in quanto una supply chain può essere significativamente diversa da un'altra [69].

L'applicazione di contratti di lavoro digitali può essere utile a superare il problema della generalizzazione in quanto permette di tracciare il processo produttivo senza dover scendere nei particolari dello stesso, permettendo alle aziende di avere più controllo sulla supply chain senza doversi munire di costosissime attrezzature.

2.6 TOKENIZZAZIONE

Da un punto di vista economico, Madsen et al. in [52] hanno studiato il rapporto ricchezza-reddito e quindi la disuguaglianza della ricchezza mondiale tra il 1850 e il 2015.

Gli autori dimostrano che essa è correlata negativamente al tasso di crescita economica e positivamente ai tassi di investimento in beni immateriali e tangibili. In particolare dal 1970, gli investimenti in attività immateriali sono stati un fattore determinante nella disuguaglianza della ricchezza.

La tokenizzazione delle attività e la valutazione di token basati su attività umane potrebbero portare più consapevolezza e accessibilità agli investimenti, evitando di continuare a favorire la disuguaglianza della distribuzione della ricchezza.

Le monete digitali sono strumenti finanziari nuovi, innovativi e di grande interesse. Proprio perchè nuovi, innovativi e complessi, gli stessi governi sono attenti a definire un quadro normativo. Allo stato attuale, la questione aperta dal punto di vista normativo è classificarli in una delle seguenti categorie: share, utility o security token.

Grazie agli Smart Contract, dal 2014, è diventato possibile creare token blockchain che traspongono gli asset fisici nel mondo digitale. In particolare esistono tre metodi per tokenizzare asset reali su una blockchain:

- token garantiti da asset;

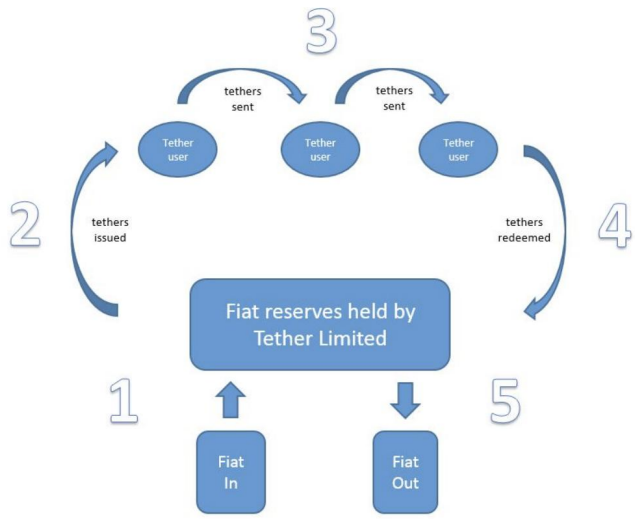

Figure 2.1: Utilizzare le riserve per portare una risorsa reale in una risorsa digitale.

- token algoritmici;
- token supportati da attività.

Un esempio di token garantiti da asset sono USDT [25], USDC e PAXOS [10]. Essi funzionano tutti allo stesso modo (Fig. 2.1): un'istituzione centralizzata custodisce gli asset reali (es. USD o Oro) nelle proprie riserve ed emette un token per ogni dollaro USA o per ogni oncia/grammo di oro.

Il secondo tipo di tokenizzazione è realizzato attraverso algoritmi: Dai [24], la stablecoin di MakerDAO, è il token primogenito di questo tipo, ed è sovracollateralizzato da altre monete digitali.

Un altro esempio di tokenizzazione attraverso gli algoritmi sono le stablecoin che utilizzano il signoraggio (l'ecosistema Terra [44] faceva parte di queste ed era garantito dal signoraggio con la moneta digitale LUNA).

Nel 2022 è stato dimostrato che le monete stabili completamente algoritmiche senza una garanzia o sovracollateralizzazione non sono affidabili.

UST ha perso la stabilità legata al dollaro USA e LUNA è crollata a zero soffrendo di quella che viene chiamata *Death Spiral*[3].

L'ultimo tipo di token è quello in esame all'interno di questo elaborato e comprende quegli asset coperti da attività nell'economia reale.

In questo caso, viene coniato un token da uno Smart Contract a patto che vi siano attività produttive a copertura.

Questo concetto è ciò che si avvicina di più a quello di equity, dove l'andamento delle azioni delle aziende quotate in borsa rispecchia gli utili delle società che le emettono.

Nei capitoli 4 e 6 mostriamo rispettivamente una PoC della creazione di token basati su attività reali e l'applicazione del modello matematico TES a diversi asset crittografici a partire da bitcoin, per poi mostrare un'applicazione ad un token basato su una materia prima: il litio.

2.7 RISPETTO AL SISTEMA MONETARIO ATTUALE

Ammous, nel suo lavoro "The Bitcoin Standard: The Decentralized Alternative to Central Banking" [2], fa una interessante panoramica sui sistemi monetari in essere durante la storia dell'uomo. In particolare viene mostrato come in antichità, dopo il baratto, si sia iniziato ad utilizzare come sistema di pagamento oggetti come conchiglie o pietre, passando alle monete in oro e poi alle monete emesse attraverso il *Gold Standard*.

Nel 1971 il *Gold Standard* viene eliminato lasciando le valute fiat prive di collaterale e basate totalmente sulla fiducia nelle banche centrali.

Nel libro [2], l'autore descrive la sua visione su un'economia basata su bitcoin allo stesso modo di come era organizzata durante il *Gold Standard*. Il testo, purtroppo, suona più come una critica al sistema monetario attuale, piuttosto che la presentazione di una valida e più vantaggiosa altenativa.

Difatti Warren E. Weber, in [74], analizzando a fondo le differenze fra il *Gold Standard* e un ipotetico *Bitcoin Standard*, giunge

3 Dai sopravvive grazie alla sovracollateralizzazione, oltre ad avere altre *stablecoin* come collaterale

alla conclusione che uno standard Bitcoin avrebbe due vantaggi principali rispetto agli attuali standard di denaro fiat.

Il primo sarebbe la maggiore prevedibilità dei prezzi. Nonostante bitcoin attualmente sia un asset molto volatile, grazie al determinismo dell'emissione delle monete, questa volatilità verrà annullata completamente in futuro.

Il secondo vantaggio è che le risorse che attualmente vengono dedicate alla copertura contro le fluttuazioni dei tassi di cambio verrebbero utilizzate in maniera più produttiva.

Tuttavia, secondo l'autore, è improbabile che uno standard simile al *Gold Standard* basato su Bitcoin possa esistere, perché i governi e le banche centrali prenderanno provvedimenti per prevenirlo in quanto interessate a proteggere le rendite generate dal signoraggio frutto del denaro creato a costo zero. Inoltre, per loro è necessario mantenere la capacità di attuazione di politiche di interesse al fine di influenzare le proprie economie nazionali.

Tecnologicamente parlando, anche se si dovesse ipoteticamente arrivare ad un *Bitcoin Standard*, è presumibile che possa nascere una innovazione tecnologica che fornisca benefici uguali o maggiori a quelli di Bitcoin e con costi inferiori, che lo renda velocemente obsoleto.

Per tali motivi, immaginiamo che le monete generate dal lavoro umano tracciato in blockchain possano funzionare maggiormente come delle *royalties* o asset finaziari piuttosto che come un nuovo standard monetario. Tali convinzioni hanno portato la generazione di una ricerca che permettesse di emettere e valutare tali asset. La stessa garantisce una gestione più ragionata di un portafoglio che comprenda al proprio interno asset innovativi oltre ad asset tradizionali. Inoltre, lato sicurezza, questo elaborato risponde efficacemente a tutti i rischi evidenziati in [7] nel capitolo 8.

3 TECNOLOGIE ABILITANTI

In questo capitolo presentiamo gli strumenti che hanno permesso di sviluppare le simulazioni, gli Smart Contract e di effettuare la validazione dei modelli.

3.1 RSTUDIO

Al fine di eseguire le simulazioni descritte nei capitoli 5 e 6 sono stati utilizzati il linguaggio R ed RStudio.

RStudio è un ambiente di sviluppo integrato per R, un linguaggio di programmazione per il calcolo statistico ed il plotting di risultati statistici.

Il linguaggio R, attraverso le variabili aleatorie, ha permesso la creazione dei dataset per le simulazioni nell'ambito della presentazione della teoria dell'infoincertezza e infoincompletezza [35] e per l'esempio mostrato nel capitolo 6, nel modello del TES [39], ha inoltre permesso di generare i relativi plot.

3.2 LINGUAGGI DI PROGRAMMAZIONE PER SMART CONTRACT EVM

Nonostante la EVM utilizzi il linguaggio a cascata bytecode, esistono diversi linguaggi di programmazione ad alto livello per sviluppare Smart Contract EVM: i più diffusi sono Vyper e Solidity.

Vyper

Questo linguaggio è basato su Python ed ha come focus la leggibilità e la sicurezza. Al fine di garantire la trasparenza per chi esegue i contratti, Vyper semplifica enormemente la scrittura di Smart Contract, eliminando ereditarietà, overload delle funzioni o degli operatori, modificatori e ricorsività. Nessuno dei citati è necessario al fine di ottenere un linguaggio Turing-completo

e questo garantisce una maggiore leggibilità del codice per chi non è un programmatore. All'aumentare della complessità dei concetti citati, aumentano i problemi di sicurezza. Inoltre, i modificatori possono favorire la scrittura di codice ingannevole. Gli sviluppatori di Vyper hanno dichiarato che Vyper "vieterà deliberatamente alcune funzionalità o renderà le cose più difficili qualora lo si ritenga opportuno per l'obiettivo di aumentare la sicurezza". Quindi Vyper non mira ad essere un sostituto di Solidity, ma si propone come linguaggio di programmazione da utilizzare quando la sicurezza, la semplicità e la trasparenza sono prioritarie.

Solidity

Solidity è un linguaggio di programmazione orientato agli oggetti per la scrittura di Smart Contract. Esso fornisce ereditarietà degli oggetti, librerie e dati tipati. È famoso per essere utilizzato per l'implementazione di Smart Contract Ethereum, ma viene utilizzato anche per altre blockchain come Tron e Binance Smart Chain (BSC), le quali sono compatibili con la EVM. Solidity viene quindi compilato in bytecode ed è eseguito dalla EVM. A oggi Solidity ha raggiunto un buon livello di maturità ed è quindi la prima scelta per la programmazione di Smart Contract EVM. Proprio per questo motivo è stato scelto Solidity per la programmazione degli Smart Contract presentati in questo elaborato.

3.2.1 *Compilatori*

I compilatori Solidity si occupano di "tradurre" il linguaggio di programmazione ad alto livello in bytecode. Negli anni sono pochi gli strumenti che sono stati sviluppati per la compilazione di Smart Contract Solidity, trattandosi una tecnologia ancora molto giovane. Molti di quelli che sono stati proposti, sono deprecati. Gli unici compilatori che hanno raggiunto una buona maturità sono Remix e la suite Truffle.

Suite Truffle

La suite Truffle è un framework completo per lo sviluppo di Smart Contract Solidity. Essa si divide in quattro moduli:

- Truffle: si tratta di un framework per sviluppare, testare e rilasciare Smart Contract Solidity su diverse blockchain. In particolare, per Ethereum e le altre reti EVM compatibili, è possibile distribuire contratti sia su mainnet che su testnet. È altresì possibile testare i contratti su una blockchain locale grazie a Ganache;
- Ganache: è una blockchain locale per lo sviluppo rapido di applicazioni distribuite EVM e Corda. Consente di sviluppare, distribuire e testare le applicazioni decentralizzate in un ambiente sicuro e deterministico. Ganache UI supporta sia la tecnologia Ethereum che Corda. Una versione Ethereum di Ganache è disponibile come strumento da riga di comando: ganache-cli (precedentemente noto come TestRPC);
- Drizzle: è una raccolta di librerie front-end che rendono più semplice la scrittura del front-end delle applicazioni decentralizzate;
- Truffle teams: permette di gestire e monitorare lo stato di salute delle applicazioni blockchain. Esso include: tracciamento e dati dApp; test automatici continui degli Smart Contract; distribuzioni automatizzate; monitoraggio di transazioni, stati ed eventi di uno Smart Contract distribuito; visualizzazione cronologica della build e dello stato corrente nel flusso di lavoro.

La suite Truffle, quindi, è un ambiente a 360° che costituisce la migliore scelta per la progettazione e lo sviluppo di Smart Contract e applicazioni decentralizzate di discreta grandezza, le quali comprendono lo sviluppo di diversi Smart Contract e l'interazione fra loro.

Se il suo vantaggio è la facilità con cui si gestiscono progetti molto grandi, lo svantaggio è la grande mole di settaggi che

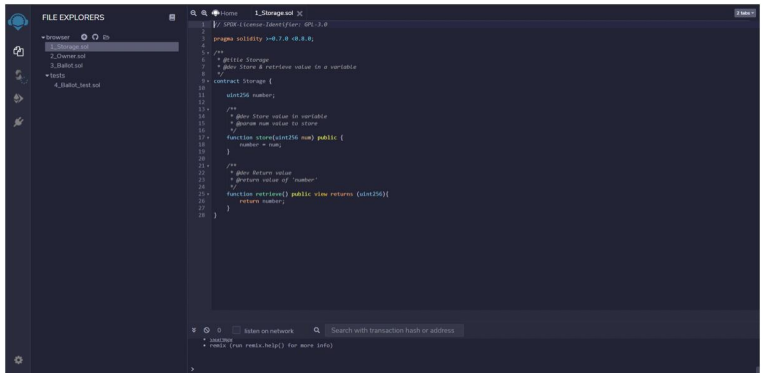

Figure 3.1: Remix IDE.

questo tool necessita per funzionare. Siccome per questo lavoro si è deciso di distribuire i contratti su una testnet pubblica al fine di testare i contratti in un ambiente più dinamico, e poichè gli Smart Contract da pubblicare sono relativamente pochi, si è deciso di utilizzare un compilatore più immediato.

Remix

Remix è un compilatore Solidity che nasce online, ma che è stato recentemente rilasciato anche come IDE desktop. È ottimo per la compilazione e il deploy di Smart Contract semplici e per chi vuole iniziare ad imparare a programmare Smart Contract. L'interfaccia è estremamente semplice e il rilascio di un contratto si divide in tre passi:

- creazione del contratto;
- compilazione;
- distribuzione.

Creazione del contratto: una volta entrati nell'IDE, sulla sinistra (Figura 3.1) troviamo i file Solidity a disposizione, è possibile importare file da github o organizzare i file in cartelle e crearne di nuovi. Dalla figura si può comprendere come Remix sia un ottimo tool per lo sviluppo di progetti semplici, con pochi contratti, ma che diventa confusionario quando i contratti diventano numerosi.

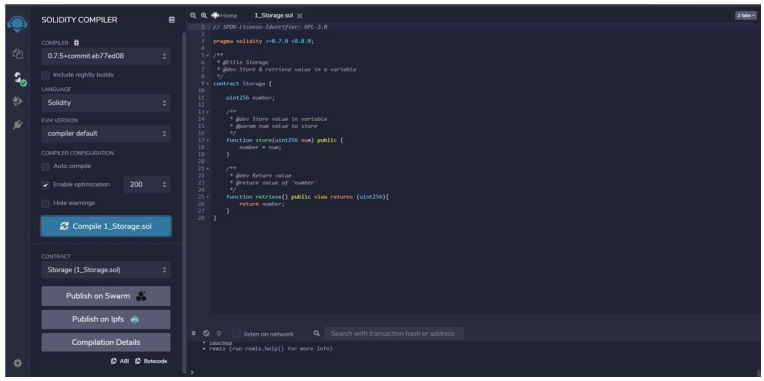

Figure 3.2: Compilazione di un contratto.

Compilazione: terminata la scrittura del contratto è possibile compilarlo nella sezione "compilazione" (Figura 3.2). Qui è possibile selezionare la versione del compilatore (dichiarata nel contratto attraverso l'istruzione *Pragma*). Molto importante è l'opzione di ottimizzazione della compilazione. Grazie a questa opzione, il bytecode generato viene ottimizzato e si ottiene una discreta riduzione dei costi in GAS. Dopo aver modificato le opzioni, si deve cliccare sul bottone "Compile" per dare inizio alla compilazione del contratto.

Deploy: dopo aver compilato il contratto, la fase finale è la sua pubblicazione su rete EVM compatibile. Utilizzando Ethereum, da Remix è possibile pubblicare un contratto sulla Mainnet dalla sezione "Deploy" (Figura 3.3), selezionando come ambiente "Injected Web3" e utilizzando Metamask. Al fine di utilizzare una delle testnet a disposizione per contratti EVM, basta selezionare la relativa rete su Metamask prima di effettuare il deploy del contratto; allo stesso modo è possibile selezionare una blockchain alternativa compatibile con EVM come Tron o Binance Smart Chain (BSC). È altresì possibile utilizzare un full node Ethereum Geth selezionando "Web 3 provider". Selezionato l'ambiente, bisogna indicare l'indirizzo da cui si vuole fare il deploy. Una volta distribuito il contratto, è possibile interagire con esso direttamente dall'ambiente Remix, dalla sezione "Deployed Contracts". Selezionando come ambiente "JavaScript VM" è possibile utilizzare l'ambiente di test Remix per testare i contratti in maniera

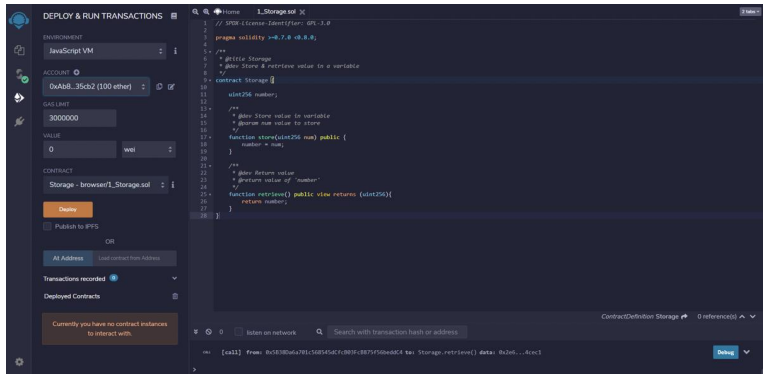

Figure 3.3: Distribuzione di un contratto su una rete blockchain.

rapida e più deterministica rispetto a una rete di test pubblica.

Verifica Etherscan

Completato il deploy degli Smart Contract è importante, al fine di garantire la trasparenza ai potenziali utenti del sistema proposto, effettuare la verifica e la conseguente pubblicazione del codice degli Smart Contract su Etherscan [1], l'explorer di Ethereum o il relativo explorer della piattaforma scelta. Al fine di effettuare la verifica del contratto è sufficiente recarsi all'indirizzo dello Smart Contract su Etherscan, selezionare il tab "Contract" e seguire il procedimento di verifica del codice.

Testnet

Non essendo gli Smart Contract modificabili dopo il rilascio, è di vitale importanza utilizzare una testnet al fine di verificare il corretto funzionamento del codice. Le reti di test sono di due tipi: private e pubbliche. Le testnet private, come Ganache, sono già state presentate in precedenza: consentono di rilasciare contratti in un ambiente deterministico. Per testare i contratti in un ambiente più dinamico, sono disponibili diverse reti di test Ethereum, ognuna con delle caratteristiche differenti.

Esse sono:

1 https://etherscan.io/

- **Ropsten**: era la prima rete di test e si trattava di quella più simile alla mainnet in quanto utilizzava come sistema di consenso la PoW. Era possibile fare mining di ETH di test oppure riceverne tramite faucet. Era supportata anche in Geth. Ropsten è stata dismessa dopo il merge e il conseguente passaggio a PoS;
- **Kovan**: utilizzava la Proof of Authority (PoA) [19]. Si tratta di un algoritmo di consenso basato sulla reputazione e sull'identità. I nodi validatori vengono selezionati arbitrariamente come entità affidabili e sono quindi limitati. Questo garantisce prestazioni e scalabilità superiori. Gli Ether di test Kovan erano ottenibili solo tramite faucet. Non era supportata in Geth. Kovan è stata dismessa dopo il merge e il conseguente passaggio a PoS;
- **Rinkeby**: come Kovan utilizzava la PoA. Gli ETH di test Rinkeby erano ottenibili solo tramite faucet. Non era supportata in Geth. Rinkeby è stata dismessa dopo il merge e il conseguente passaggio a PoS;
- **Goerli**: anche essa utilizza la PoA. Gli ETH di test Goerli sono ottenibili solo tramite faucet. È la testnet più stabile ed è supportata in Geth. Goerli è l'unica testnet a non essere stata dismessa dopo il merge e il conseguente passaggio a PoS.

Strumenti di analisi dei costi

Essendo quello dello sviluppo di Smart Contract un settore relativamente giovane, non esistono strumenti di analisi dei costi delle transazioni. Questo è un punto focale per un'azienda che vuole proporre una soluzione blockchain e la mancanza di tool di analisi dei costi è un'assenza molto pesante dal punto di vista del risk management. L'unico modo per analizzare i costi, attualmente, è quello di utilizzare una testnet attivando l'opzione "mostra conversione nelle reti di test" e simulando i diversi prezzi del GAS in previsione di una possibile congestione di rete. Proprio per questo motivo nel capitolo 8 sarà mostrata una prospettiva dei costi sulle diverse piattaforme EVM.

3.3 STRUMENTI PER LO SCRAPING

Per la validazione della teoria descritta in questo elaborato, è stato costruito un dataset raccogliendo i dati dalle seguenti fonti web:

- Transfermarkt;
- Wikipedia.

In entrambi i casi è stato necessario utilizzare la tecnica del web scraping in quanto non è stato possibile ottenere le informazioni utili al lavoro presentato in altra maniera. Questa tecnica è stata implementata sviluppando uno script in linguaggio Python e utilizzando la libreria Beautiful Soup. Questa permette di estrarre dati da file HTML e XML. Può essere utilizzata in combinazione con un parser il quale permette di facilitare la raccolta dei dati fornendo modi idiomatici di navigazione, ricerca e modifica dell'albero di analisi.

Part II

TOKENIZZAZIONE E VALUTAZIONE DEL LAVORO

In questa parte viene descritta la soluzione proposta ai fini della raccolta dei dati riguardanti il lavoro umano. In particolare, la soluzione presente nel capitolo 4 utilizza la combinazione fra tecnologia IoT e Smart Contract e produce asset blockchain basati sulle attività lavorative. La teoria del capitolo 5 permette la ricostruzione delle probabilitá in contesti dove mancano le informazioni. Tale teoria si applica perfettamente ai mercati finanziari, dove si ha bisogno di prendere decisioni e dove le informazioni a disposizione dell'operatore non sono complete in quanto soggettive. Il risultato é l'applicazione della Financial Computing ai token generati dagli Smart Contract presenti nel capitolo 4 attraverso un modello matematico chiamato TES, descritto nel capitolo 6, il quale permette di effettuare una valutazione di questi asset al fine di favorire la gestione del rischio in un portafoglio strutturato.

4

TOKENIZZAZIONE DEL LAVORO

4.1 SMART CONTRACT APPLICATI AL MONDO DEL LAVORO: UNA PROOF OF CONCEPT

In questo capitolo mostriamo una PoC per la gestione di prestazioni lavorative con gli Smart Contract. Questa PoC è pensata per essere orientata ai lavori a prestazione, ma può essere specializzata per essere utlizzata in diversi settori. Nel presente elaborato è stata concepita per lavori che prevedono uno spostamento (come i corrieri o i rider), al fine di mostrare alcune potenzialità in termini assicurativi, di vincolo di paga minima, rispetto del lavoro in condizioni avverse, privacy e anti discriminazione. Essa, inoltre, favorisce l'eliminazione dei tempi delle operazioni burocratiche risultando parallelamente flessibile. In particolare può essere utilizzata per svariate operazioni, per esempio:

- generare asset che tokenizzano il tempo/uomo e i dati raccolti durante un lavoro;
- depositare stablecoin (o Central Bank Digital Currencies, CBDC) gestendo i pagamenti e le relative regole tramite il paradigma della tecnoregolazione;
- utilizzare monete digitali (comprese materie prime tokenizzate) come pagamento ai dipendente;
- raccogliere i dati sulla prestazione lavorativa, favorendo il controllo delle supply chain da parte delle aziende e l'utilizzo di curriculum certificati per i lavoratori;
- gestire la tassazione in maniera rigorosa;
- gestire eventuali controversie.

I vantaggi quindi sono molteplici, ma al fine di garantire l'adeguata regolazione di tali strumenti è necessaria la presenza di studi in ambito legislativo al fine di favorirne il corretto utilizzo. Tali

studi vanno al di la di questo lavoro, che quindi si propone come pilot study e base per eventuali sviluppi futuri in questo campo.

4.1.1 *Implementazione*

Gli Smart Contract sono stati sviluppati in Solidity siccome è il linguaggio utilizzato dalle piattaforme blockchain EVM compatibili, permettendone così la distribuzione sulle alternative a Ethereum senza dover modificare il codice. A tal proposito si propone, nel capitolo 8, uno studio e una proiezione sui costi degli Smart Contract sviluppati e rilasciati su diverse piattaforme EVM compatibili, permettendo di dare uno sguardo all'utilizzo in contesti reali[1].

Al fine di mostrare le potenzialità della combinazione fra tecnologie blockchain e IoT è previsto l'utilizzo di un dispositivo IoT wearable contenente una chiave privata blockchain che raccoglie le informazioni dal mondo reale e le inserisce in quello digitale attraverso lo Smart Contract.

Lo pseudocodice dell'implementazione degli Smart Contract si trova nella sezione 4.4 di questo capitolo. Il codice completo degli Smart Contract è presente nell'appendice A di questo elaborato. In questo capitolo, invece, ne descriviamo il funzionamento.

Quando si sviluppa uno Smart Contract, la prima cosa da decidere è se esso sia completamente decentralizzato oppure abbia bisogno di un admin. Siccome Lo Smart Contract proposto nella PoC è pensato per essere utilizzato dalle aziende come "metodo di pagamento" e strumento tecnoregolativo, è stato sviluppato by design come un contratto *Ownable*, ciò significa che può essere di proprietà di un indirizzo che otterrà alcuni poteri decisionali su di esso.

In questo caso si è scelto di lasciar distribuire lo Smart Contract sulla piattaforma blockchain direttamente dall'azienda. Di default, quando un *employer* (in questo caso, l'azienda) distribuisce un contratto, questo viene automaticamente impostato come *Owner* rendendola effettivamente proprietaria del contratto e

1 Quasi tutti i lavori di ricerca in questo ambito propongono Ethereum in quanto è la piattaforma più utilizzata, ma come vedremo, alle condizioni tecnologiche attuali essa non è sostenibile nel lungo periodo in un contesto reale

garantendole alcuni poteri di admin. La proprietà conferisce all' *employer* la possibilità di eseguire quattro azioni:

- effettua un deposito sul contratto tramite la funzione *Payable*[2] deposit(). Questa funzione tiene semplicemente traccia dei depositi del datore di lavoro attraverso un evento depositMade(uint256 amount).
  Attraverso questo approccio, a seconda della piattaforma utilizzata, il deposito può essere effettuato nella valuta nativa della piattaforma. Abbiamo scelto questo metodo per semplicità, ma i pagamenti potrebbero essere effettuati anche in token ERC20 (EURS, USDC, ecc.) o, in futuro, CBDC modificando questa funzione. Allo stesso tempo, alla fine dell'esecuzione, verranno coniati e distribuiti al lavoratore token ERC20 che in questo caso sono trattati come royalties;

- aggiungere un indirizzo di un dispositivo IoT verificato dall'azienda tramite la funzione addIoT(address IoT). Questa funzione aggiunge l'indirizzo del dispositivo IoT alla mappa di indirizzi *registeredIoT*. Questi indirizzi sono gli unici in grado di richiamare la funzione addDelivery(uint256 latitude, uint256 longitude, uint256 fatigue, bool accident, bool isNight, uint256 start, uint256 end, bool isUnfavorableWeather, bool isHoliday, address worker), questa funzione e i suoi parametri verranno approfonditi in seguito;

- rimuovere un indirizzo di un dispositivo IoT verificato tramite la funzione removeIoT(address IoT). Allo stesso modo, il datore di lavoro può rimuovere gli indirizzi dei dispositivi IoT in caso di compromissione degli stessi;

- accettare un lavoro completato da un dipendente attraverso la funzione acceptJob (address payable employee). Tale funzione analizzerà il lavoro effettuato dal dipendente, calcolerà e trasferirà automaticamente il pagamento[3]. In questa

2 Una funzione si dice Payable quando nella transazione accetta degli Eth(nel caso dell'utilizzo della piattaforma Ethereum) in input da poter utilizzare nella funzione

3 il vincolo di soglia della paga minima potrebbe essere prelevato in maniera decentralizzata tramite oracolo Chainlink. In questa PoC, la paga minima è fissa

PoC è stato scelto questo approccio per lasciare la decisione di accettazione del lavoro al datore, ma il calcolo e il pagamento potrebbero essere automatici al completamento del lavoro, per far si che questo sia possibile, basta spostare il codice di questa funzione in addDelivery o gestire il tutto con variabili booleane.

Nel momento in cui l'azienda distribuisce lo Smart Contract tramite un apposito client che ne semplifica il processo, ne verrà automaticamente richiamato il costruttore. Esso inizializza il set di indirizzi dei dispositivi IoT e imposta l'indirizzo dei token ERC20 che fungono da royalties. Allo stesso tempo verrà ceduta la proprietà di questi token al nuovo Smart Contract. Questa operazione permette di dare a quest'ultimo pieni poteri sulla coniazione.

Dopo l'inizializzazione i lavoratori possono registrarsi tramite la funzione employeeRegistration(), essa si occuperà di inserire l'indirizzo del lavoratore nella matrice employees[] utilizzando la variabile msg.sender[4]. Su questo inidirizzo verrà inviata la retribuzione alla fine del lavoro.

Le consegne dei dipendenti vengono tracciate attraverso una *struct*. All'interno della struttura sono presenti alcune informazioni come le ore lavorative, le condizioni metereologiche, le posizioni e conseguentemente permettono di calcolare la paga finale. In particolare la struttura delivery è costituita dalle seguenti variabili:

- address employee: è l'indirizzo del dipendente che ha eseguito la consegna;
- bool isUnfavorableWeather: viene settato a true se c'è tempo sfavorevole, false altrimenti. Nella PoC, il meteo viene analizzato attraverso un'applicazione Smartphone che interagisce con il dispositivo IoT[5] è predisposto il codice per l'utilizzo dell'oracolo;
- uint256 startBlock: contiene il numero del blocco nel quale la consegna è iniziata;

4 msg.sender indica l'indirizzo che sta interagendo con lo Smart Contract, esso viene inserito all'interno dell'array employees tramite push()

5 per una maggior decentralizzazione, potrebbe essere utilizzato un oracolo blockchain come Chainlink, in A

- uint256 endBlock: contiene il numero del blocco corrispondente nel quale si è conclusa la consegna;
- uint256 latitude: è la latitudine della consegna. Questo dato viene raccolto tramite applicazione dallo Smartphone;
- uint256 longitude: la longitudine della consegna. Questo dato viene raccolto tramite applicazione dallo Smartphone;
- uint256 start: il timestamp dell'inizio della consegna. Viene raccolto dallo Smartphone;
- uint256 end: il timestamp della fine della consegna. Raccolti dallo Smartphone;[6];
- uint256 fatigue: la fatica del dipendente, ricavata dal battito cardiaco fornito dal dispositivo IoT[7];
- bool accident: impostato su true se il dipendente subisce un incidente, false altrimenti. Questo dato viene raccolto tramite dispositivo IoT, viene utilizzato per eventuali indennizzi assicurativi, rendendo anche l'assicurazione gestita automaticamente dagli Smart Contract;
- bool isNight: impostato su true se la consegna avviene durante la notte, false altrimenti;
- bool isHoliday: impostato su true se la consegna avviene durante una vacanza, false altrimenti. Nella PoC, sia isHoliday che is Night vengono assegnati utilizzando i dati raccolti dallo Smartphone[8];

Ogni consegna viene inserita dal dipendente utilizzando in combinazione i dispositivi IoT e lo Smartphone con la tecnologia NFC, questa interazione permette la compilazione automatica dei parametri di cui ha bisogno lo Smart Contract per la paga

6 sia per l'inizio che per la fine della consegna è possibile utilizzare il timestamp o il numero di blocco

7 Nella PoC questa variabile è inutilizzata. Potrebbe essere utilizzata per il calcolo del pagamento, se si vuole pagare il dipendente in base alla sua fatica. Questo potrebbe essere utile per premiare di più le persone più deboli o per avere soccorsi tempestivi in caso di problemi

8 anche in questo caso per una maggiore decentralizzazione si potrebbe utilizzare Chainlink

della prestazione, permettendo al dipendente di completare la consegna con un solo semplice gesto.

Il wallet contenuto nel dispositivo IoT richiamerà la funzione addDelivery(uint256 latitude, uint256 longitude, uint256 fatigue, bool accident, bool isNight, uint256 start, uint256 end, bool isUnfavorableWeather, bool isHoliday, address worker) che costruirà l'oggetto delivery e lo inserirà all'array deliveries.

La presenza del dispositivo IoT permette di isolare la chiave privata e, in combinazione con le nozioni presentate nel capitolo 8 rendere più sicura la firma della transazione; allo stesso tempo permette di prelevare informazioni sullo stato fisico del lavoratore le quali possono essere preziose per eventuali royalties, indennizzi o soccorsi.

Nella PoC sviluppata nel presente elaborato, quando l'azienda accetta la conclusione di un lavoro, deve richiamare la funzione acceptJob(address payable employee). Questa scelta di design è stata presa al fine di non rendere l'azienda completamente impotente rispetto alla decisione di accettare o no un lavoro, ma potrebbe essere estesa ad una terza parte in caso di gestione di controversie[9].

Lo Smart Contract analizzerà le consegne, calcolerà i pagamenti e li trasferirà nel portafoglio del dipendente. La formula per il calcolo della paga è la seguente:

$$t = \begin{cases} ph \; se \; f = 0 \\ (p + 0.1p)h \; se \; f = 1 \\ (p + 0.15p)h \; se \; f = 2 \\ (p + 0.20p)h \; se \; f = 3 \end{cases}$$

con il vincolo $p \geq m$.

Dove t è la paga finale, p è la paga oraria del lavoro, m è la paga minima oraria, h è il numero di ore di lavoro ed f è il numero di parametri settati a true fra *meteo*, *orario* e *festivi*.

Nella PoC è inoltre presente la funzione makeReview(string memory reviewReference) che verrà richiamata dal wallet del

9 Si è deciso di non estendere questo contesto in quanto si ricadrebbe in ulteriori ricerche in ambito tecno-regolativo che vanno oltre gli obiettivi del presente elaborato

dipendente nel caso in cui voglia rilasciare una recensione sull'azienda. Questa funzione memorizzerà un riferimento IPFS della recensione sulla blockchain (al fine di non appesantire sulla stessa la quantità di dati inseriti), in questa maniera, le recensioni possono essere scritte solo da chi ha effettivamente collaborato con l'azienda e sarebbero quindi certificate. Le interazioni degli attori con il sistema sono descritte in Figura 4.1.

Nella prossima sezione descriveremo il ciclo di esecuzione della PoC.

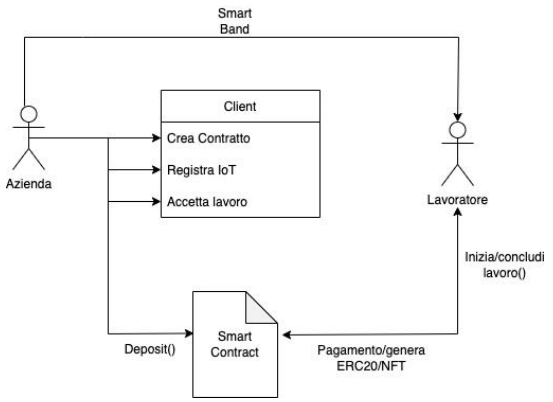

Figure 4.1: Interazione fra lavoratore, azienda e sistema.

4.2 ESEMPIO DI ESECUZIONE

In questa sezione simuliamo un ciclo di esecuzione del sistema. Tutte le operazioni possono essere eseguite in modo custodial (per utenti inesperti) e in modo non custodial (per esperti) se l'utente preferisce mantenere il possesso del proprio portafoglio tramite chiavi private. Il ciclo di esecuzione è suddiviso in cinque fasi come segue:

1. l' *employer* crea lo Smart Contract attraverso il client: questa operazione viene semplificata utilizzando un client in cui è memorizzato uno Smart Contract pronto per la distribuzione. Essa distribuirà lo Smart Contract in modo trasparente all'*employer* il quale deve solo inserire gli indirizzi dei dispositivi IoT in un modulo;

2. l' *employer* ricarica i fondi Smart Contract attraverso la funzione a pagamento deposit(), il deposito potrebbe essere effettuato una sola volta in quanto consente di pagare tutti i dipendenti che partecipano al contratto e potrebbe funzionare quindi da fondo stipendi semplificando la contabilità;

3. l' *employee* si registra sul client che richiamerà la funzione employeeRegistration() dello Smart Contract, questa transazione sarà firmata con la chiave privata del dipendente;

4. l' *employer* fornisce il dispositivo IoT NFC all' *employee*;

5. l'*employee* firma l'inizio del lavoro con il dispositivo IoT in dotazione e lo esegue: questa operazione viene eseguita tramite l'applicazione smartphone e la tecnologia NFC;

6. l' *employee* firma la fine del lavoro con il dispositivo IoT in dotazione. L'NFC attiverà la firma della funzione addDelivery(uint256 latitude, uint256 longitude, uint256 fatigue, bool accident, bool isNight, uint256 start, uint256 end, bool isUnfavorableWeather, bool isHoliday, address worker). Le informazioni necessarie da passare come parametri possono essere prelevate tramite Smartphone[10] oppure utilizzando un oracolo, rendendo il tutto decentralizzato;

7. *employer* accetta il lavoro del *employee* tramite la funzione acceptJob(address employee) che attiverà il pagamento.
   Il pagamento potrebbe anche essere automatizzato quando l'*employee* termina il lavoro, ma in questa PoC si è scelto di lasciare la decisione all'azienda in quanto si è preferito lasciare un maggior controllo sui pagamenti da parte della stessa;

8. nel caso di un lavoratore occasionale, l' *employee* potrebbe lasciare una recensione certificata facoltativa utilizzando il client.

10 per semplicità nella PoC sono prelevate dallo Smartphone

4.3 ASSET GENERATI

Gli Smart Contract sviluppati in questa PoC sono flessibili e possono essere utilizzati per diversi scopi, garantendo tutti i vantaggi derivati dall'utilizzo degli Smart Contract e dello storage tramite blockchain. Anche il risultato scaturito al termine del lavoro risulta flessibile e può essere di tre tipi:

- pagamento semplice: l'azienda ha depositato dei fondi allocati per gli stipendi e lo Smart Contract si occupa di calcolare e inoltrare il compenso che può essere erogato in CBDC, token ERC20 o monete digitali;
- generazione di token fungibili: si tratta di token che vengono generati al completamento del lavoro in maniera variabile in base al mercato, alle ore di lavoro, alla fatica, alle condizioni avverse, etc. Questo tipo di token risulta particolarmente interessante in quanto programmabile: potrebbe essere parte di retribuzione del tipo *Work for Equity*; potrebbe essere legato alle materie prime, in caso di estrazione delle stesse (Capitolo 6). Questo rende il lavoratore più consapevole in ambito finanziario, favorendo una migliore distribuzione della ricchezza [52];
- generazione di NFT: token che vengono generati o aggiornati al completamento del lavoro. Ogni NFT mantiene la storia lavorativa del dipendente e i dati storici relativi al proprio lavoro diventando così un curriculum digitale pseudonimo, permettendo al dipendente di poter gestire la proprietà dei dati.

Si noti che i tre approcci non risultano in contrasto fra di loro e potrebbero essere implementati nello stesso Smart Contract, permettendo di gestire nello stesso momento i pagamenti, le royalties e i dati. Definiti gli asset che possono essere creati in questo contesto, al fine di valutarli, nel prossimo capitolo utilizzeremo una modello basato sulla teoria dell'infoincertezza [39].

4.4 PSEUDOCODICE

4.4.1 *Variabili*

```
[] employees
hasEmployeeWorked
incentive
incentiveMap
minimumPaymentPerHour=11
struct delivery{
  employee;
  isUnfavorableWeather;
  startBlock;
  endBlock;
  latitude;
  longitude;
  startTimestamp;
  endTimestamp;
  //counted in time there, could be also in block for better
      decentralization
  fatigue;
  hadAccident;
  isNight;
  isHoliday;
}
[] deliveries
registeredIoTMap
```

4.4.2 *Eventi*

```
depositMade(amount)
reviewMade(numberOfDeliveries,review)
paymentMade(amount,employee)
```

4.4.3 *Costruttore*

```

constructor(_registeredIoT,_incentive):
incentive=_incentive
for(registeredIoT):
registeredIoTMap[i]=true
```

4.4.4 *Funzioni*

```
employeeRegistration():
employees.push(msg.sender)


addDelivery(collected from IoT):
require(isRegistered(Iot)
build delivery object
push delivery in deliveries
hasWorked[worker]=true

acceptJob (employee) onlyOwner:
for(deliveries):
calculate pay with conditions
transfer pay to employee
set incentive for employee
emit payment event

deposit() payable onlyOwner:
deposit through payable function
emit depositMade event

addIoT(IoTAddress) onlyOwner:
registeredIoT[IoTAddress]=true

removeIoT(IoTAddress) onlyOwner:
registeredIoT[IoT]=false

makeReview(reviewReference):
require(hasWorked[msg.sender])
mint send incentive to msg.sender
emit reviewMade event
```

5

DECISIONI IN CONTESTI DI INFO-INCERTEZZA E INFO-INCOMPLETEZZA

5.1 INTRODUZIONE ALLA TEORIA DELLA PLAUSIBILITÀ

La base di questo lavoro è la teoria delle decisioni in contesti di info-incertezza e info-incompletezza [35], in questo capitolo riprendiamo i concetti che permettono di sviluppare sistemi decisionali che fanno utilizzo di questa teoria.

Per descrivere l'info-incertezza e info-incompletezza, si pensi al lancio di un dado a sei facce. In questo caso sappiamo che la probabilità che il risultato sia pari a uno è di $\frac{1}{6}$ perchè esso dovrebbe essere completamente casuale fra le sei facce.

Se la probabilità venisse descritta in questa maniera e conoscessimo tutte le informazioni relative alle facce, all'angolo di lancio, alla pressione atmosferica, alla forma del suolo, etc. saremmo in grado di calcolare il risultato esatto del lancio con una probabilità del 100%. Se conoscessimo una parte di queste informazioni, allora le probabilità inizierebbero a cambiare. Se invece non conoscessimo il numero totale delle facce del dado, ci troveremmo a decidere in contesti di info-incertezza e info-incompletezza.

Un esempio reale caratterizzante di contesti in cui la mancanza di informazione è quasi totale sono i mercati finanziari. In essi ogni operatore è un'entità a sé stante ed ha un comportamento diverso da ogni altro. Il mercato è quindi formato da miliardi di operatori che prendono decisioni basate sulla propria esperienza e conoscenza, le quali determinano l'andamento del prezzo di un asset.

Nel contesto del decision making, avendo a disposizione un sistema con delle variabili, si ha bisogno di analizzarle al fine di trovare una soluzione che ottimizzi il payoff del sistema. Quando ci sono più soluzioni o una soluzione target raggiungibile attraverso diverse strategie competitive, al fine di formalizzare un problema bisogna:

- definire le condizioni iniziali;
- fissare i vincoli;
- individuare l'obiettivo;
- definire la strategia della soluzione;
- ottimizzare la soluzione.

Una prima classificazione del problema è stabilire se esso è deterministico o stocastico. Le variabili che compongono il problema, invece, potrebbero essere descritte da leggi dinamiche fuori dal controllo dell'analista. Per questo motivo è emerso il bisogno di estendere la teoria della probabilità al fine di descrivere un evento o un sistema. Per questo motivo, in letteratura, sono stati definiti diversi concetti. Il primo ipotizzato è quello della plausibilità.

Da un punto di vista filosofico, uno dei primi a definire il ragionamento plausibile è Polya [57, 58], il suo pensiero è in opposizione al ragionamento dimostrativo.

Successivamente Rescher, nel 1976, concepisce il ragionamento plausibile come una forma di deduzione a partire da premesse incerte, finalizzato al trattamento delle "dissonanze cognitive".

Dezert, nel 2002, tratta il ragionamento plausibile come un'estensione della teoria dell'evidenza di Dempster-Shafer[20, 65], e quindi della probabilità bayesiana. La sua teoria permette di affrontare informazione non solo incerta, ma anche paradossale. Questo ragionamento è stato ampiamente studiato: Cuzzolin in [15] definisce le proprietà della plausibilità. Friedman, Halpern e Barnett hanno calcolato e misurato la plausibilità [6, 31], mentre Shved, Wang e Wen forniscono un metodo per la misurazione dell'incertezza [66, 72, 75].

Il calcolo della plausibilità [6, 31]è stato utilizzato anche in applicazioni nel mondo reale, come in [18].

La teoria della plausibilità, ancora prima di Polya, era conosciuta nell'antichità grazie a Sesto Empirico, il quale diffuse la composizione di Carneade di Cirene "Against the Logicians".

In questo contesto la plausibilità non è necessariamente qualcosa che sia vero o si creda vero, ma soddisfa alcuni requisiti:

- è plausibile se sembra vero;

- è ancora più plausibile e stabile, cioè sembra vero ed è anche compatibile con altri eventi che sono plausibili;
- è stabile e testato.

In letteratura, oltre la plausibilità, viene definita l'incertezza, la quale può essere di due tipi:

- aleatoria o stocastica, la quale riguarda la casualità dell'oggetto del sistema di conoscenza;
- epistemica o soggettiva, cioè quando si riferisce allo stato di conoscenza del soggetto sull'ambiente.

L'incertezza della plausibilità appartiene al secondo tipo. Successivamente questa teoria viene ripresa da Polya nel 1954. Nei suoi lavori [57, 58] inizia a definire la plausibilità come casi particolari degli eventi, affermando: "l'analogia e i casi particolari sono le fonti più abbondanti di argomenti plausibili". Aggiungendo poi che: "l'inferenza per analogia sembra la forma più comune ed essenziale di inferenza. Essa produce congetture più o meno plausibili che possono o non possono essere confermate dall'esperienza o da un ragionamento più rigoroso". In questo caso, Polya indica la plausibilità come un ragionamento impersonale che produce conclusioni di tipo provvisorio e soggettivo. Le regole impersonali legate al ragionamento plausibile sono quindi descritte da Polya in [57].

Per lui la probabilità è il fulcro della nozione di plausibilità in quanto il teorema di Bayes è il fondamento del suo concetto di plausibilità.

Basandosi sulla concezione bayesiana, la concezione probabilistica di Polya ne risente di tutti i limiti.

Per superare i limiti della concezione bayesiana e della relativa visione dell'inferenza plausibile, è stata definita la teoria dell'evidenza di Dempster-Shafer, di seguito indicata come DS. Anche per questa teoria la concezione bayesiana è alla base della nozione di plausibilità, ma la regola di combinazione sostituisce quella bayesiana. In particolare, "include la teoria bayesiana come un caso speciale e quindi conserva almeno alcune delle attrattive di tale teoria" [65].

La teoria DS parte da una serie di alternative che si escludono a vicenda, chiamate discernimento o universo del discorso u. La

teria DS rinuncia alla rappresentazione unidimensionale della credenza non assegnando la restante parte della credenza stessa alla negazione della proposizione.

La teoria DS è formata da tre componenti:

- probabilità di base;
- funzione di credenza;
- funzione di plausibilità dei sottoinsiemi u, chiamati proposizioni.

La teoria DS definisce la seguente assegnazione a una probabilità di base:

$$m : 2^u \to [0,1] \text{ tale che } m(\emptyset) = 0, \sum_{A \subseteq u} m(A) = 1$$

che rappresenta ed esprime numericamente la forza dell'evidenza, la credenza esatta che un agente ha in A. La funzione

$$m : 2^u \to [0,1]$$

è chiamata funzione di credenza Bel, quando soddisfa le seguenti condizioni:

- $Bel(\emptyset) = 0$;
- $Bel(u) = 1$.

È ora possibile definire la plausibilità. Shafer osserva che "una proposizione è plausibile alla luce dell'evidenza al punto in cui l'evidenza non supporta il suo contrario" [65].

Quindi introduce la funzione dubbio, Dub, ponendola uguale a $Bel(\neg A)$ dove $\neg A$ è il complemento di A in u e definiamo la funzione di plausibilità, Pl, come

$$Pl(A) = 1 - Dub(A) = 1(Bel\neg A).$$

La funzione di plausibilità esprime quanto si dovrebbe credere nella proposizione A se tutti i fatti non noti allo stato attuale supportassero A. Questa funzione esprime anche il massimo valore probabilistico che può essere attribuito a una proposizione A, in particolare "misura la massa totale di credenza che può essere spostata in A" [20].

Il limite inferiore corrisponde alla funzione di credenza, e quindi vale la relazione

$$Bel(A) \leq Pl(A).$$

Secondo la teoria DS, la differenza tra credenza e plausibilità si riflette anche a livello funzionale: la funzione credenza "è spesso zero per tutte le proposizioni atomiche in domini complessi, a meno che non sia disponibile un gran numero di prove". L'altra grande differenza tra la teoria bayesiana e la teoria DS è la combinazione di prove, cioè la regola di aggiornamento delle credenze quando c'è una nuova evidenza. Infatti, nella teoria DS la regola della combinazione sostituisce quella bayesiana e viene chiamata somma ortogonale.

La somma ortogonale soffre di alcuni noti limiti strutturali, il principale dei quali sono i risultati, che potrebbero essere inaspettati e non intuitivi quando da situazioni complesse si va a ridurre al semplice [79].

Successivamente Dezert e Smarandache, nel 2009, al fine di superare le difficoltà della teoria DS, studiano e definiscono una nuova teoria chiamata Dezert-Smarandache, di seguito denominata DSm [22].

Nella teoria DSm, la plausibilità è un limite superiore del valore di una probabilità che un evento o una proposizione può assumere. Similmente alla teoria DS, la plausibilità DSm è concepita come un modello concettuale per guidare il processo decisionale in condizioni di incertezza, estendendo quella introdotta nella teoria DS. Gli stati informativi, nella teoria DSm, vengono così definiti:

- si dice "razionale" quando l'assegnazione di base m è a somma 1 e la chiusura degli operatori $\cap$ e $\cup$ sugli elementi dell'universo u è 0;
- si dice strettamente "incerto" quando è a somma 1 e la chiusura dell'operatore $\cup$ può essere diversa da 0;
- si dice "paradossale" quando è a somma 1 e la chiusura dell'operatore $\cap$ può essere diversa da 0;
- si dice "incerto e paradossale" quando la chiusura degli operatori $\cap$ e $\cup$ può essere diversa da 0.

Anche se la modifica delle ipotesi produce effetti sul modello concettuale, essa non consente di superarne i limiti. Sebbene la teoria DSm tenda a una discussione più ampia del ragionamento plausibile, il suo approccio contribuisce a lasciarlo ancorato a una visione fortemente riduzionista.

Tra gli approcci non probabilistici alla teoria della plausibilità, spicca l'approccio deduttivista di Rescher, il quale invece di adottare la teoria della probabilità come base per modellare la plausibilità, basa il suo approccio sul tradizionale principio di Teofrato, tornando alle radici storiche della teoria della plausibilità.

Secondo Nicholas Rescher, la "teoria della plausibilità mira a fornire uno strumento razionale per trattare le dissonanze cognitive" [61]. Rescher fornisce una distinzione preliminare tra plausibilità e probabilità:

- la plausibilità "classifica le proposizioni secondo lo stato delle fonti probatorie o dei principi che le garantiscono";
- la probabilità "pesa varie alternative e le valuta".

Rescher non sviluppa mai calcoli che convergano la nuova conoscenza in nuovi risultati, ma procede solo confrontando i diversi dati.

Un ulteriore aspetto dell'approccio deduttivistico alla nozione di plausibilità è il collegamento con l'argomento entimematico in cui la conoscenza comune consente di integrare le ipotesi non esplicite all'interno di una sequenza di inferenza.

La teoria della plausibilità secondo Rescher permette di stabilire se le proposizioni sono dei "candidati" alla verità o no. Si tratta quindi di una plausibilità argomentativa la quale può essere definita come il valore massimo tra i valori plausibilistici minimi delle integrazioni entimematiche che consentono una conclusione deduttiva derivata dalle premesse.

Un altro approccio alla teoria della plausibilità è quello di Collins-Michalscki (CM) [13]. Questo tipo di approccio viene chiamato cognitivo e mira a mostrare come i fattori soggettivi e psicologici siano decisivi nel processo decisionale, formalizzando quindi le inferenze plausibili individuate nelle risposte alle domande alle quali le persone non hanno una risposta pronta.

5.1.1 *Decisioni in contesti di info-incertezza e info-incompletezza*

L'ultimo lavoro, cronologicamente parlando, che descrive la teoria della plausibilità, estendendo le teorie descritte in precedenza, è stato proposto nel 2020 da G.Iovane et al. in [35].

Secondo gli autori, il primo elemento da estrapolare è la definizione della certezza dell'obiettivo. In altre parole, bisogna chiedersi se l'obiettivo è raggiungibile o meno.

Secondo lo studio, i problemi decisionali complessi sono generati dai dualismi:

- determinismo e indeterminismo;
- completezza e incompletezza.

I quali generano un nuovo dualismo: certezza e incertezza dell'obiettivo. Il dualismo incerto è affrontato attraverso un insieme di sfumature linguistiche e si parla quindi di:

- probabile e improbabile;
- plausibile e implausibile;
- credibile e incredibile;
- possibile e impossibile.

L'obiettivo di [35] è capire come utilizzare, quantificare e generalizzare queste nozioni al fine di costruire sistemi di supporto al processo decisionale non solo probabilistici.

La ragione dell'utilizzo di tali termini è dovuta al fatto che alcuni fenomeni o sistemi complessi non sono distribuiti in maniera gaussiana (o altre distribuzioni note), ma hanno code pesanti che non risultano gaussiane.

L' esempio più adatto alla descrizione di questo fenomeno è la distribuzione del prezzo di uno strumento finanziario attorno a un prezzo campione in cui si osserva una distribuzione non normale, con code pesanti. Ciò significa che alcuni eventi improbabili possono ancora verificarsi, quindi è importante stimare se l'evento sia plausibile, credibile o possibile e determinarne l'accadibilità.

L'esempio risulta estremamente calzante in quanto nei mercati finanziari gli operatori hanno diverse esperienze, impressioni

soggettive, emotività e convinzioni che rendono il processo decisionale soggettivo.

Lo scopo dei sistemi di supporto alle decisioni, quindi, deve essere quello di costruire scenari simulati, come quelli della teoria dei giochi. Proprio per questo motivo, in [35] vengono effettuate due simulazioni che mostrano il comportamento dei sistemi complessi in condizioni di info-incertezza e l'esistenza delle "code pesanti".

Le simulazioni sono legate a due ambiti diversi: quello sportivo e quello dell'analisi di un sistema multibiometrico. In entrambi i casi sono stati creati due dataset basati sulle probabilità, aggiungendo della casualità che simula l'incertezza delle informazioni, dimostrando così l'esistenza delle code pesanti.

Dopodiché, applicando al dataset la modellazione descritta, gli autori stimano l'accadibilità degli eventi simulati.

Grazie a questa metodologia, l'operatore interessato può essere portato a decidere razionalmente (operatore razionale indotto) prima del raggiungimento del risultato (tempo di inoltro della decisione).

L'incompletezza informativa e le variabili/informazioni, o l'indeterminazione dei legami funzionali, generano nel decisore emozioni che inducono a convinzioni. Grazie a un sistema di supporto alle decisioni basato sulle simulazioni, l'operatore viene portato ad essere più razionale grazie a una base cognitiva più ampia (conoscenza estesa). Il nuovo momento di realizzazione porterà allo stato di ipercoscienza in cui l'operatore è consapevole dello scenario aumentato chiamato iperscenario.

A differenza di György Pólya o Arthur Dempster e Glenn Shafer, i quali hanno cercato di estrarre, in termini probabilistici, la funzione di credenza dalla componente descrivibile, in [35] si accetta la differenza tra i termini, gerarchizzandoli in termini di forza e descrivendoli con un metodo più generale e non esclusivo di probabilità.

In questa visione, i concetti di Probabilità (Pr), Plausibilità (Pl), Credibilità (Cr) e Possibilità (Po) hanno una forza decrescente, anche se descrivono tutti eventi incerti.

Quindi:

- un evento possibile non è necessariamente credibile;

- un evento credibile è sicuramente possibile;
- un evento credibile non è necessariamente plausibile, ma è sicuramente possibile;
- un evento plausibile non è necessariamente probabile, ma è sicuramente credibile e possibile;
- un evento probabile è sicuramente plausibile, credibile e possibile.

Con questo presupposto è ovvio che, in condizioni di incertezza o incompletezza dell'informazione, l'improbabile verificarsi di eventi apre lo spazio alle emozioni che demarcano in modo significativo il passaggio dai metodi di stretta incertezza (probabilità) al trattamento degli eventi incerti sotto l'azione degli effetti emotivi ed esperenziali.

In [35] quindi questi concetti vengono modellati assegnando una probabilità a un evento in base all'opinione di più persone. La credibilità fa emergere la valutazione personale e soggettiva da condividere, possibile, ma non necessariamente condivisibile. Da queste considerazioni emerge che, poiché il concetto di probabilità è più stringente di quello di plausibilità, nella prima assumiamo che, anche se c'è incertezza, non c'è valutazione guidata dalla soggettività; invece nella definizione di plausibilità, associamo una componente emotiva all'incertezza, ma rappresenta l'opinione della maggior parte delle persone. Nella credibilità, l'emotività e l'esperienza fa emergere l'io, cioè una visione personale e soggettiva che non richiede il confronto con la comunità, ma ha solo bisogno della possibilità.

5.2 INFERENZA BASATA SU INFORMATION FUSION PER SISTEMI INCERTI

Definiamo la funzione di accadibilità a, ottenuta da un'opportuna composizione funzionale in termini di Probabilità (Pr), Plausibilità (Pl), Credibilità (Cr), Possibilità (Po), tale che $a(\emptyset) = 0$ e $\Sigma_{i=1}^{4}$ $a(Ai) = 1$ con $A \in (Pr, Pl, Cr, Po)$.

Definiamo successivamente sette modelli utili a descrivere e calcolare l'accadibilità di un evento. Ogni modello è pensato per soddisfare un determinato tipo di evento, risultando in un'alta

generalizzazione che rende il tutto adattabile a qualsiasi tipo di evento (come poi dimostrato nel capitolo 7 di questo elaborato).

5.2.1 *I: Modello basato sulle medie*

Indichiamo con $P_1 = Pr, P_2 = Pl, P_3 = Cr, P_4 = Po$. Il modello più semplice della funzione di accadibilità che è possibile costruire è il seguente modello, il quale è basato sulle medie:

$$a_1 = a_1(P_1, P_2, P_3, P_4) = \frac{1}{4}\Sigma_{i=1}^{4} P_i$$

. Questo modello assegna la stessa importanza alle diverse distribuzioni di Probabilità, Plausibilità, Credibilità e Possibilità, quindi potrebbe essere sia vantaggioso che svantaggioso a seconda di quanto esperto è l'analista che effettua l'analisi. Questo modello, come nella teoria della probabilità, viene utilizzato per calcolare l'aspettativa di più eventi alternativi. Un esempio di utilizzo di questo modello potrebbe essere quello delle previsioni del tempo, assegnamo:

- P_1=previsione del luogo A dell'aviazione militare;
- P_2=previsione del luogo B effettuata tramite sentiment analysis;
- P_3=previsione degli esperti per il luogo C;
- P_4=previsione degli utenti per il luogo D

Questo modello può essere utilizzato per calcolare la funzione di accadibilità per l'evento "pioverà nel luogo A o nel luogo B o nel luogo C o nel luogo D".

5.2.2 *II: Modello basato sul prodotto*

Un altro modo per stimare l'aspettativa di eventi è attraverso il prodotto P_i, formalmente

$$a_2 = a_1(P_1, P_2, P_3, P_4) = \prod_{i=1}^{4} P_i$$

. In questo modello, come nel precedente, tutti i P_i hanno la stessa importanza. In questo caso facciamo un altro esempio riguardante le previsioni del tempo in un determinato luogo:

- P_1=secondo l'aviazione militare, pioverà;
- P_2=secondo la sentiment analysis il vento soffierà a 31km/h;
- P_3=gli esperti prevedono una temperatura di 18 C°;
- P_4=secondo mia madre ci sarà umidità alta.

Questo modello può essere utilizzato per calcolare la funzione di accadibilità per l'evento "pioverà, il vento soffierà a 31km/h, ci saranno 18 C° e l'umidità sarà dell'80%".

5.2.3 *III: Modello della media ponderata*

Per estendere i modelli precedenti assumiamo che non tutti i P_i abbiano la stessa importanza nella costruzione della funzione di accadibilità. Costruiamo quindi il seguente modello a media ponderata:

$$a_3 = \frac{\sum_{i=1}^{4} \alpha_i P_i}{\sum_{i=1}^{4} \alpha_i},$$

dove α_i sono pesi relativi ai diversi P_i. Un esempio potrebbe essere quello di dare un 50% di peso a P_1 ovvero:
α_1 = 0, 5; del 25% a P_2, quindi α_2 = 0, 25; 15% a P_3, quindi α_3 =0, 15, e 10% a P_4 , α_4 = 0, 1; in questo modo la funzione di accadibilità è $a_3 = \frac{\sum_{i=1}^{4} \alpha_i P_i}{\sum_{i=1}^{4} \alpha_i} = 0,5P_1 + 0,25P_2 + 0,15P_3 + 0,1P_4$.
In entrambi i modelli pesati abbiamo il vincolo di $\sum_{i=1}^{4} \alpha_i = 1$.
Questo modello è quello che utilizzeremo per il calcolo dei best price degli asset finanziari.

5.2.4 *IV: Modello del prodotto ponderato*

In analogia, per estendere la funzione di accadibilità del modello basato sul prodotto, consideriamo il seguente modello pesato:

$$a_4 = \frac{\prod_{i=1}^{4} \alpha_i P_i}{\sum_{i=1}^{4} \alpha_i}.$$

Anche in questo modo avremo la possibilità di pesare il diverso contributo della Probabilità, Plausibilità, Credibilità e Possibilità alla funzione di accadibilità per eventi combinati.

5.2.5 *V: Modello con sovrapposizione basato sullo shift e sulla probabilitá*

Il primo e il terzo rappresentano modelli di sovrapposizione e sovrapposizione ponderata dei diversi contributi dei P_i alla funzione di accadibilità. Nel V modello, gli autori definiscono un'ulteriore possibile generalizzazione la quale include un utilizzo gerarchico di P_i. Sia P_r la probabilità dell'evento, in questo modello è possibile definire Probabilità, Plausibilità, Credibilità e Possibilità come concetti da utilizzare in ordine decrescente di importanza. Ad esempio possiamo considerare

$$a_5 = \begin{cases} P_1, & se \quad 1\% < Pr \leq 100\% \\ P_2, & se \quad 0,1\% < P_r \leq 1\% \\ P_3, & se \quad 0,01\% < P_r \leq 0,1\% \\ P_4, & se \quad P_r \leq 0,01\% \end{cases}$$

In altre parole, fino all'1% della distribuzione di probabilità utilizzeremo solo la Probabilità classica (P_1); oltre tale probabilità definiremo l'evento improbabile, con un coefficiente di Plausibilità (P_2) compreso fra 0% e 100% nell'intervallo in cui $1\% < P_r \leq 0,1\%$; analogamente, nell'intervallo $0.1\% < P_r \leq 0.01\%$ definiamo l'evento improbabile e non plausibile, con Credibilità (P_3) compresa tra 0% e 100%; infine, per valori di probabilità ancora più bassi avremo eventi improbabili e non plausibili, con un coefficiente di Possibilità (P_4) compreso fra 0% e 100%. Alla fine di queste valutazioni c'è l'evento certo $P_1 = 100\%$ e quello impossibile $P1 = 0\%$.

5.2.6 *VI: Modello con sovrapposizione basato su P_i gerarchico*

In alternativa al modello precedente, senza che la probabilità funga da pivot, ma generalizzando i P_i valutando anche gli elementi emotivi e cognitivi, in [35] si definisce un sesto modello di funzione di accadibilità che può essere definita come:

$$a_6 = \begin{cases} P_1, & se \quad \overline{P}_{P_1} - 3\sigma P_1 \leq p_t \leq \overline{P}_{P_1} + 3\sigma P_1 \\ P_2, & se \quad \overline{P}_{P_2} - 3\sigma P_2 \leq P_t \leq \overline{P}_{P_2} + 3\sigma P_2 \\ & e \quad p_t > \overline{P}_{P_1} + 3\sigma P_1 \\ P_3, & se \quad \overline{P}_{P_3} - 3\sigma P_3 \leq P_t \leq \overline{P}_{P_3} + 3\sigma P_3 \\ & e \quad p_t > \overline{P}_{P_2} + 3\sigma P_2 \\ P_4, & se \quad \overline{P}_{P_4} - 3\sigma P_4 \leq P_t \leq \overline{P}_{P_4} + 3\sigma P_4 \\ & e \quad p_t > \overline{P}_{P_3} + 3\sigma P_3 \end{cases}$$

Dove $\overline{P}$ è la media su l'insieme di dati considerato e σ è la deviazione standard; il pedice P_t indica con quale distribuzione P e σ vengono calcolate.

5.2.7 *VII: Modello basato sulle regole di composizione di Dempster*

L'ultimo modello per costruire la funzione di accadibilità fornito in [35] è basato sulla regola che Dempster definisce per la plausibilità [65][20], estendendola all'applicazione del caso P_i calcolando la funzione di massa (m), la funzione credenza(bel) e la plausibilità di Dempster (Dpl) per ogni P:

- $m(P_1)$, $bel(P_1)$ e $Dpl(P_1)$;
- $m(P_2)$, $bel(P_2)$ e $Dpl(P_2)$;
- $m(P_3)$, $bel(P_3)$ e $Dpl(P_3)$;
- $m(P_4)$, $bel(P_4)$ e $Dpl(P_4)$.

Componendo quindi i P_i con le regole di composizione di Dempster. Dai modelli forniti in [35] si comprende quanto sia generale il problema delle decisioni in condizioni di incertezza e quanto sia facile costruire modelli utilizzando la funzione di accadibilità che descrivono i processi di inferenza in termini di incompletezza e incertezza degli input.

Come detto, questa teoria funziona particolarmente bene nei mercati e sarà quindi ripresa nel capitolo 6 per effettuare analisi su asset finanziari.

6

TOKEN EVALUATION SYSTEM: VALUTAZIONE DI ASSET DIGITALI

6.1 INTRODUZIONE

Il TES è stato introdotto da G.Iovane et al. in [39]. In questo lavoro gli autori forniscono un modello che intende valutare il prezzo di un token blockchain e la sua presenza sul mercato, ma lo stesso potrebbe essere utilizzato con successo anche per la valutazione di asset finanziari tradizionali.

Il modello TES è costruito utilizzando il concetto di probabilità estesa [35], descritto nel precedente capitolo, nel quale la probabilità viene estesa con le nozioni di Probabilità (Pr), Plausibilità (Pl), Credibilità (Cr) e Possibilità (Po). I sette modelli per il calcolo della funzione di accadibilità, mostrati nel capitolo 5, fondono i pareri di diverse fonti di informazione, offrendo un valore di aspettativa più adatto rispetto alla probabilità tradizionale in assenza di informazioni in input. Come anticipato, tali modelli sono ottimali in contesti finanziari dove il prezzo degli asset è guidato dalle emozioni e dalle competenze degli operatori.

Il TES sfrutta la terza funzione di accadibilità la quale viene definita in [35] come:

$$a_3 = \frac{(\sum_{i=1}^{4} \alpha_i P_i)}{\sum_{i=1}^{4} \alpha_i}.$$

Dove α_i è il peso delle informazioni i e P_i è il valore i con $P_i \in (Pr, Pl, Cr, Po)$ (ad esempio un esperto pensa che il prezzo di bitcoin sarà 35000$ nel 2024, quindi $Pl = 35000$). In altre parole, il TES è un ampio modello di probabilità e non un tradizionale gaussiano, che riduce la discrepanza tra il prezzo atteso e il prezzo di mercato di una moneta digitale, nonché qualsiasi attività, strumento finanziario, prodotto, utilità, ecc.

Il TES sfrutta questi concetti sviluppando un modello che definisce tre fasi:

- fase di input, che è formata da un insieme di variabili chiamate Critical Success Factor (CSF), gli intervalli di variazione dei CSF;
- fase di valutazione e analisi, dove si costruisce il prezzo più probabile del token e ne viene calcolata l'occorrenza e l'accuratezza;
- fase di output, che restituisce il Token Expectation Value (o best price, il prezzo più probabile del token) rappresentato da una terna (TV, O(TV), R(O)) dove TV è il valore del token o il prezzo di mercato più probabile, O(TV) rappresenta il suo verificarsi e R(O) l'affidabilità dell'occorrenza.

6.1.1 *Input: i fattori Critical Success Factors (CFS)*

In [39] gli autori definiscono 13 variabili le quali rappresentano beni tangibili e non tangibili, questi beni vanno a comporre un token e sono chiamati CSF.

Ogni variabile A_n ha un peso a_i.

I 13 CFS sono definiti come:

$$CFS = (A_1, ..., A_n) \in B^{13} \quad \text{con} \quad n = 1, ..., 13.$$

dove B^{13} è l'insieme booleano $\{0, 1\}$ a 13 dimensioni, e

- A_1 : è la variabile che riguarda gli asset fisici, la tokenizzazione avviene utilizzando uno dei tre metodi per tokenizzare asset reali su una blockchain. Di nostro interesse per questo lavoro è il metodo basato sulle attività umane. Questo CSF ha un peso $0 \leq a_1 \leq 0,5$;
- A_2 : è la variabile che riguarda il valore dell'asset tecnologico, ad esempio la blockchain, gli Smart Contract, l' utilizzo di oracoli, le Applicazioni decentralizzate (dApp), ecc. Questo CSF ha un peso $0 \leq a_2 \leq 0,15$;
- A_3 : comprende l'acquisizione dei diritti per produrre il bene fisico (A_1). Con un peso $0 \leq a_3 \leq 0,2$;
- A_4 : comprende la rete Asset to Market, ovvero la presenza dell'asset in diversi e affidabili exchange (o borse); la sua

promozione e presenza in diversi paesi. Questo CSF ha un peso $0 \leq a_4 \leq 0,15$;

- A_5 : rappresenta l'offerta della moneta digitale definita come $1 - \frac{T_{sold}}{T_{minted}}$ dove T è Token (o moneta digitale). Questo CSF ha un peso $0 \leq a_5 \leq 0,1$;
- A_6 : rappresenta la disponibilità di attività lavorative documentate del processo produttivo. Questo CSF ha un peso $0 \leq a_6 \leq 0,05$;
- A_7 : rappresenta il social sentiment relativo al token e ai suoi casi d'uso. Questo CSF ha un peso $0 \leq a_7 \leq 0,05$;
- A_8 : rappresenta l'opinione degli esperti i quali effettuano l'analisi del token e dei suoi casi d'uso. Questo CSF ha un peso $0 \leq a_8 \leq 0,15$;
- A_9 : rappresenta la reputazione di coloro che hanno ideato e coniato la moneta digitale o il token. Questo CSF ha un peso $0 \leq a_9 \leq 0,05$;
- A_{10} : rappresenta la reputazione di chi emette effettivamente la moneta digitale o il token. Questo CSF ha un peso $0 \leq a_{10} \leq 0,05$;
- A_{11} : rappresenta la reputazione del team di sviluppo e degli advisors. Questo CSF ha un peso $0 \leq a_{11} \leq 0,05$;
- A_{12} : rappresenta l'originalità progetto e i suoi casi d'uso. Questo CSF ha un peso $0 \leq a_{12} \leq 0,03$;
- A_{13} : rappresenta il livello di innovazione del progetto. Questo CSF ha un peso $0 \leq a_{13} \leq 0,03$.

Queste variabili sono utilizzate in [39] per la stima del Token Value (TV), in Figura 6.1 è graficata la composizione di un token in base ai CSF.

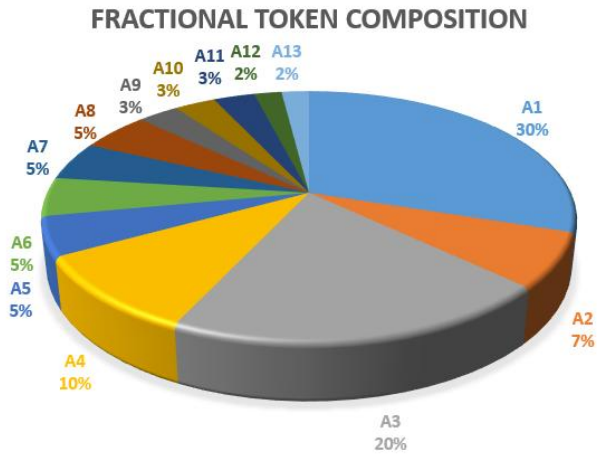

Figure 6.1: Composizione di un token nel modello TES.

6.1.2 *Fase di analisi*

Nella fase di analisi si effettua una scannerizzazione delle caratteristiche del token, similmente all'esecuzione dell'analisi fondamentale sull'equity[1]. Qui, dopo aver analizzato caratteristiche come la tokenomics, il whitepaper, i casi d'uso, il mercato, ecc, a ciascuno dei 13 CSF viene assegnato un valore. In questo caso gli autori definiscono TV come:

$$TV = \sum_{n=1}^{13} a_n A_n$$

dove, $TV = TV(t)$, cioè TV è una funzione al tempo t, poiché il peso $a_n = a_n(t)$ sono funzioni del tempo nel processo di produzione. In questa fase, possiamo trovarci in tre diverse situazioni:

- $\sum_{n=1}^{13} a_n < 0.5$, tipico dei nuovi progetti dove è presente solo un token, il suo whitepaper o una roadmap e qualche sponsorship;
- $0.5 \leq \sum_{n=1}^{13} a_n < 1$, comprende quei progetti che sono innovativi dal punto di vista tecnologico, hanno casi d'uso

1 Mercato azionario

interessanti che ne aumentano la domanda, un'ottima reputazione e marketing, ma non sono coperti da beni fisici;

- $\sum_{n=1}^{13} a_n > 1$, il valore è maggiore di 1 quando un progetto è affermato e coperto anche da beni fisici.

Dopo aver analizzato i fattori critici di successo, Iovane et al. definiscono gli output del TES con una terna $(TV, O(TV), R(O))$; la descriviamo in dettaglio nel seguente paragrafo.

6.1.3 *Output del TES*

Dopo aver analizzato la composizione del token, andiamo a definire gli output del TES. Il primo è il Token Expectation Value (TV), che, come visto nel paragrafo precedente, rappresenta il valore più attendibile dell'asset. TV è definito come:

$$TV(t) = \sum_{n=1}^{13} a_n^t A_n.$$

Secondo la teoria dell'info-incertezza e info-incompletezza, la probabilità non è sufficiente a rappresentare il valore di un asset a causa delle condizioni di incertezza o incompletezza delle informazioni presenti nel mercato. Inoltre il prezzo è formato dall'emotività ed esperienza differente fra tutti gli operatori che prendono decisioni in diverse fasi macroeconomiche.

Gli autori quindi definiscono un valore di occorrenza di TV, utilizzando la Probabilità, la Plausibilità, la Credibilità e la Possibilità [35] al fine di descrivere al meglio un valore stimato dell'asset digitale in analisi. Più precisamente, definiscono il verificarsi di un evento E, $O(E)$ come segue:

$$O(E) = \gamma_1 Pr(E) + \gamma_2 Pl(E) + \gamma_3 Cr(E) + \gamma_4 Po(E),$$

dove $Pr(E)$ è la Probabilità, $Pl(E)$ la Plausibilità, $Cr(E)$ la Credibilità e $Po(E)$ la Possibilità che l'evento E accada, mentre i γ_i sono i relativi pesi i quali generalmente sono decrescenti rispetto i con il vincolo:

$$\sum_{n=1}^{4} \gamma_n = 1.$$

Applicando O a TV, si ottiene $O(TV)$ che rappresenta l'occorrenza con la quale il valore di accadibilità del token TV si verifica. Una buona approssimazione iniziale dei pesi in $O(TV)$ può essere:

$$\gamma_1 = 0,5, \quad \gamma_2 = 0,25, \quad \gamma_3 = 0,15, \quad \gamma_4 = 0,10.$$

Applicando i pesi come descritto otteniamo che:

- Probabilità cioè l'evidenza scientifica oggettiva ha un peso del 50%;
- Plausibilità cioè l'opinione degli esperti ha un peso del 25%;
- Credibilità cioè il social sentiment ha un peso del 15%;
- Possibilità cioè un'opinione di un potenziale acquirente ha un peso del 10%.

L'ultimo membro della terna è l'affidabilità dell'occorrenza, definita come la deviazione standard dei diversi valori di

$$Pr(E), Pl(E), Cr(E), Po(E)$$

.

Ricapitolando, l'output del TES è quindi:

$$(TV, O(TV), R(O)), \tag{6.1}$$

dove TV è il Token Expectation Value e cioè il valore intrinseco del token, O è l'affidabilità del valore TV e $R(O)$ è l'accuratezza dell'occorrenza O.

6.2 APPLICAZIONE DEL TES ALL'ASSET BITCOIN

Come primo esperimento, utilizziamo il modello TES per adattare i dati dell'asset bitcoin alla scala temporale giornaliera relativa al Q3[2] del 2022 e supponiamo che:

- lo stock fisico della moneta digitale bitcoin sia legato al mercato del mining (costi di energia elettrica e hardware). Legando alla probabilità il valore A_1 di TV, e considerando che il costo di mining attuale di un bitcoin è di 22039$[8] (Figura 6.2), otteniamo TV_P=22039 con probabilità $P = 99\%(\gamma_1 = 0.99)$;

2 Penultimo quarto dell'anno

- l'opinione degli esperti, la quale è legata alla plausibilità, analizzando il ciclo macro-economico, stimano un prezzo TV_{Cr} =20000$ con una plausibilità $Pl = 90\%(\gamma_2 = 0.90)$[3];
- la visione comune connessa al sentiment, corrispondente alla credibilità, è più pessimistica e si aspetta una ulteriore capitolazione del prezzo a 15000$. Quindi TV_{pl} =15000$ con una credibilità pari a $Cr = 75\%(\gamma_3 = 0.75)$;
- gli operatori poco esperti (possibilità) sono in preda al panico e hanno paura che il prezzo possa continuare a scendere fino ai 12000$. Quindi TV_{Po}=12000$ con una possibilità pari a $Po = 95\%(\gamma_4 = 0.95)$.

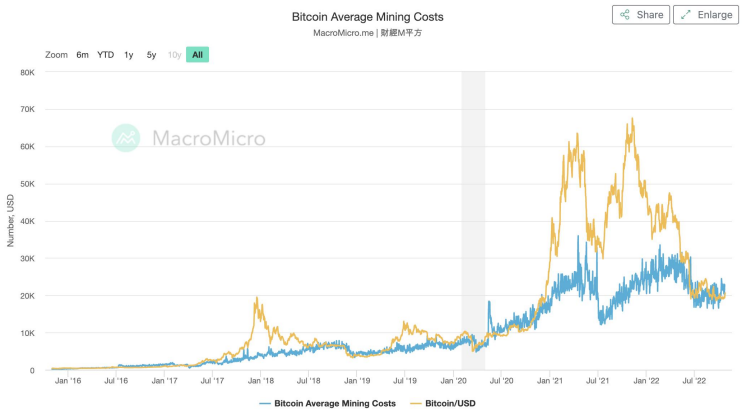

Figure 6.2: Costo del mining per un bitcoin in blu e prezzo di mercato di bitcoin in arancione.

Come l'esempio citato nel TES [39] assegnamo A_1 a TV_{Pr}, A_x a TV_{Pl}, A_7 a TV_{Cr} e A_y a TV_{Po} dove A_x e A_y sono composti da:

$$A_x = A_2 + A_5 + A_8 + A_{12} + A_{13}$$

$$A_y = A_3 + A_4 + A_6 + A_9 + A_{10} + A_{11}$$

.

Definiamo i CSF:

3 https://coinpedia.org/bitcoin/bitcoin-bottom-is-just-around-the-corner-heres-the-timeline/

- $A_1 = 0.45$ in quanto sostenuto dall'industria del mining, ma non con materie prime a copertura;
- $A_2 = 0.05$ siccome si tratta della prima moneta digitale mai creata, ed è la più sicura e testata negli anni, ma risulta poco scalabile;
- $A_3 = 0$, bitcoin non prevede diritti nella produzione di beni fisici;
- $A_4 = 0.15$, bitcoin è presente in tutte gli exchange sul mercato;
- $A_5 = 1 - \frac{19187418}{21000000} = 0.086$
- $A_6 = 0$, non sono previste attività lavorative a copertura;
- $A_7 = 0.05$ il social sentiment verso i casi d'uso della rete Bitcoin è estremamente positivo;
- $A_8 = 0.15$ l'opinione degli esperti verso i casi d'uso della rete Bitcoin è estremamente positiva;
- $A_9 = 0$ Bitcoin è decentralizzato e non è controllato da nessuno. Satoshi Nakamoto ha fatto perdere le proprie tracce nel 2010;
- $A_{10} = 0$ il mining è un processo decentralizzato, pertanto non esiste un agente che emette l'asset;
- $A_{11} = 0$ non esiste un team di advisors ed è un progetto open source alla quale chiunque può contribuire;
- $A_{12} = 0.03$ bitcoin è la prima moneta digitale creata e ha diversi casi d'uso (descritti nell'introduzione di questo elaborato);
- $A_{13} = 0.03$ Bitcoin si è rivelata una rivoluzione in ambito finanziario.

Otteniamo quindi un Token Value (TV) pari a:

$$TV = A_1 TV_{Pr} + A_8 TV_{Pl} + A_7 TV_{Cr} + A_x TV_{Po} =$$

$$= 0.5 * 22039 + 0.35 * 20000 + 0.05 * 15000 + 0.15 * 12000 =$$

$$= 20569.50\$$$

Dall'analisi risulta quindi che il prezzo di bitcoin è in linea con il suo valore o leggermente sottovalutato in quanto il prezzo reale si aggira intorno ai 19000$ al momento della scrittura.

Ora andiamo a calcolare l'occorrenza di TV O(TV):

$$O(TV) = \gamma_1 TV_{Pr} + \gamma_2 TV_{Pl} + \gamma_3 TV_{Cr} + \gamma_4 TV_{Po} =$$

$$= 0.5 * 0.99 + 0.25 * 0.90 + 0.15 * 0.75 + 0.10 * 0.95 = 93\%.$$

Infine calcoliamo l'affidabilità di O(TV) attraverso l'ultima componente della terna R(O) utilizzando la deviazione standard fra Probabilità, Plausibilità, Credibilità e Possibilità:
$O_S = (Pr(O), Pl(O), Cr(O), Po(O)$, quindi:

$$R(O_s) = \sigma^2(O_s) = \sqrt{\frac{\sum_{N=1}^{N}(x_i - \bar{x})^2}{N-1}} =$$

$$= std_{dev}(0.99; 0.90; 0.75; 0.95) = 0.09 = 9\%$$

Il fair value dell'asset bitcoin nel Q3 2022 secondo l'analisi TES è quindi:

$$(TV, O(TV), R(O) = (20569.50\$, 93\%, 9\%)$$

Possiamo dire quindi che con un'occorrenza del 93%, il fair value di bitcoin nel Q3 2022 è compreso fra 18718.25$ e 22420.75$.

Proviamo ora a calcolare il TV di bitcoin durante la fase di euforia del 2021:

- Legando alla probabilità il valore A_1 di TV, e considerando che il costo di mining medio di un bitcoin era di 22500$, otteniamo TV_P=22500 con probabilità $P = 99\%(\gamma_1 = 0.99)$;
- l'opinione degli esperti, la quale è legata alla plausibilità, stimava un prezzo di TV_{Cr} =73000$ con una plausibilità $Pl = 80\%(\gamma_2 = 0.80)$[4];
- la visione comune connessa al sentiment, corrispondente alla credibilità, era convinta che il prezzo di bitcoin si sarebbe andato ad aggirare intorno ai 100000$. Quindi TV_{pl} =100000$ con una credibilità pari a $Cr = 90\%(\gamma_3 = 0.90)$;

4 Fonte: https://www.newsbtc.com/news/bitcoin/jpmorgan-puts-bitcoin-at-150000-in-the-long-term-but-what-about-its-fair-value/

- gli operatori poco esperti (Possibilità) erano in piena euforia e si aspettavano prezzi anche molto al di sopra dei 100000$, stimiamo quindi 200000$. Quindi TV_{Po}=200000$ con una possibilità pari a $Po = 75\%(\gamma_4 = 0.75)$.

Assegnamo A_1 a TV_P, A_x a TV_Pl, A_7 a TV_Cr e A_y a TV_Po
Otteniamo quindi un Token Value (TV) pari a:

$$TV = A_1TV_p + A_xTV_{Pl} + A_7TV_{Cr} + A_yTV_{Po} =$$

$$= 0.45 * 22500 + 0.35 * 73000 + 0.05 * 100000 + 0.15 * 200000 =$$

$$= 70675\$$$

Dall'analisi risulta quindi che il prezzo di bitcoin nel suo all time high era in linea con il suo valore in euforia in quanto il prezzo massimo è stato di 70000$.
Ora andiamo a calcolare l'occorrenza di TV O(TV):

$$O(TV) = \gamma_1TV_P + \gamma_2TV_{Pl} + \gamma_3TV_{Cr} + \gamma_4TV_{Po} =$$

$$= 0.5 * 0.99 + 0.25 * 0.80 + 0.15 * 0.90 + 0.10 * 0.75 = 90\%.$$

Infine calcoliamo l'affidabilità di O(TV) attraverso l'ultima componente della terna R(O) utilizzando la deviazione standard fra Probabilità, Plausibilità, Credibilità e Possibilità:
$O_S = (P(O), Pl(O), Cr(O), Po(O)$, quindi:

$$R(O_s) = \sigma^2(O_s) = \sqrt{\frac{\sum_{N=1}^{N}(x_i - \bar{x})^2}{N-1}} =$$

$$= std_{dev}(0.99; 0.80; 0.90; 0.75) = 0.092 = 9.2\%$$

Il fair value dell'asset bitcoin durante l'anno 2021 secondo l'analisi TES era quindi:

$$(TV, O(TV), R(O)) = (70675\$, 90\%, 9.2\%)$$

Possiamo dire quindi che con un'occorrenza del 90%, il fair value di bitcoin era compreso fra 64172.9$ e 77177.10$.

Infine, mostriamo un esempio con più incertezza. Calcoliamo il TV di bitcoin durante il quarto halving nel 2025:

- Legando alla probabilità il valore A_1 di TV, e considerando che il costo di mining medio nel 2025 dovrebbe raddoppiare a causa dell'halving e si dovrebbe aggirare intorno ai 50000\$, otteniamo TV_{Pr}=50000, con le incertezze relative all'industria del mining, ai costi dell'hardware e dell'elettricità. La probabilità sarà quindi più bassa rispetto gli esempi precedenti: $Pr = 85\%(\gamma_1 = 0.85)$;

- gli esperti, la cui opinione è legata alla plausibilità, sono preoccupati del pattern a triplo massimo a 70000\$ e stimano quindi che il prezzo si fermerà intorno al vecchio All Time High: TV_{Cr} =70000\$ con una plausibilità $Pl = 80\%(\gamma_2 = 0.80)$[5];

- supponiamo che si generi una nuova narrativa su bitcoin e che la visione comune connessa al sentiment, corrispondente alla credibilità, torni positiva grazie al cambio di fondamentali macroeconomici e torni a convincersi che il prezzo di bitcoin si andrà ad aggirare intorno alla resistenza psicologica dei 100000\$. Quindi TV_{pl} =100000\$ con una credibilità pari a $Cr = 60\%(\gamma_3 = 0.60)$;

- supponiamo che torni l'euforia e gli operatori poco esperti (possibilità) torneranno ad aspettarsi prezzi monstre, stimiamo quindi 250000\$. TV_{Po}=250000\$ con una possibilità pari a $Po = 50\%(\gamma_4 = 0.50)$.

Assegnamo A_1 a TV_{Pr}, A_x a TV_Pl, A_7 a TV_Cr e A_y a TV_Po
Otteniamo quindi un Token Value (TV) pari a:

$$TV = A_1 TV_{Pr} + A_x TV_{Pl} + A_7 TV_{Cr} + A_y TV_{Po} =$$

$$= 0.45 * 50000 + 0.35 * 70000 + 0.05 * 100000 + 0.15 * 250000 =$$

$$= 89500\$$$

Ora andiamo a calcolare l'occorrenza di TV O(TV):

$$O(TV) = \gamma_1 TV_{Pr} + \gamma_2 TV_{Pl} + \gamma_3 TV_{Cr} + \gamma_4 TV_{Po} =$$

$$= 0.5 * 0.85 + 0.25 * 0.80 + 0.15 * 0.60 + 0.10 * 0.50 = 76.5\%.$$

5 Fonte: https://libertex.com/blog/bitcoin-price-forecasts

Infine calcoliamo l'affidabilità di O(TV) attraverso l'ultima componente della terna R(O) utilizzando la deviazione standard fra Probabilità, Plausibilità, Credibilità e Possibilità:
$O_S = (Pr(O), Pl(O), Cr(O), Po(O)$, quindi:

$$R(O_s) = \sigma^2(O_s) = \sqrt{\frac{\sum_{N=1}^{N}(x_i - \bar{x})^2}{N-1}} =$$

$$= std_{dev}(0.85; 0.80; 0.60; 0.50) = 0.14 = 14\%$$

Il fair value dell'asset bitcoin durante l'anno 2025 secondo l'analisi TES è quindi:

$$(TV, O(TV), R(O) = (89500\$, 76.5\%, 14\%)$$

Possiamo dire quindi che con un'occorrenza del 76.5%, il fair value di bitcoin nel 2025 sarà compreso fra 76970$ e 102030$.

6.3 UN ESEMPIO CON UN TOKEN COPERTO DA MATERIE PRIME E ATTIVITÀ LAVORATIVE

In questa sezione, analizzeremo completamente un'applicazione in tema industria 4.0, simile ai token royalties generate nel capitolo 4. Utilizziamo come esempio una supply chain nell'industria della produzione di litio, emettendo un token ogni volta che ne estrae un kg [62]. Descriviamo tre epoche nel ciclo di vita di un progetto nell'ambito dell'estrazione del litio:

- Whitepaper;
- Prima fase di promozione globale;
- Vendita del token e fase di startup fino ai 5 anni.

Descriviamo le fasi in dettaglio:

1. Whitepaper TV(t-1):
   il valore del token non è correlato all'asset fisico reale, ma è in balia della speculazione. È collegato solo al sentiment relativo alla lettura del whitepaper, le caratteristiche CSF

di spessore in questo momento sono il social sentiment, la prospettiva dell'analisi degli esperti, la credibilità dei fondatori e dell'emittente, la valutazione dei consulenti e del team, l'originalità e l'innovazione;

2. prima fase di promozione globale; $TV(t0)$:
   il valore del token non è ancora correlato all' asset fisico. A differenza dell'epoca precedente, non è legato solo al sentiment e alla speculazione sulle informazioni presenti nel white paper. Il lavoro è documentato, la piattaforma tecnologica è in fase di test, i contratti per la produzione di litio sono stati stipulati, il progetto è presente in diversi paesi e il 12% di token è pronto per essere venduto;

3. prevendita; $TV(t1)$:
   Il valore del token non è ancora correlato all' asset fisico, ma al sentiment e alla speculazione sulle informazioni presenti nel white paper. Il lavoro è documentato, la piattaforma tecnologica è in fase di pubblicazione, i contratti per la produzione di litio sono stati stipulati, il progetto è presente in diversi paesi e il 12% di token è pronto per essere venduto;

4. 1 anno dalla vendita dei primi token; $TV(t2)$:
   Il 20% del litio è stato prodotto e il 20 % di token è stato emesso. Il Token Value è meno influenzato da caratteristiche come i diritti di produzione, la presenza in diversi paesi, la credibilità degli sviluppatori e dell'emittente e degli sponsor rispetto alle epoche precedenti ed inizia a pesare di più il valore effettivo del litio;

5. 2 anni dalla vendita dei primi token; $TV(t3)$:
   Il 40% del litio è stato prodotto e il 40 % di token è stato venduto. Il Token Value è meno influenzato da caratteristiche come i diritti di produzione, la presenza in diversi paesi, la credibilità degli sviluppatori e dell'emittente e degli sponsor rispetto alle epoche precedenti e il valore del litio pesa sempre di più;

6. 3 anni dalla vendita dei primi token; $TV(t4)$:
   Il 60% del litio è stato prodotto e il 60 % del token è stato venduto. Il Token Value è meno influenzato da caratteristiche come i diritti di produzione, la presenza in diversi

paesi, la credibilità degli sviluppatori e dell'emittente e degli sponsor rispetto alle epoche precedenti e il valore del litio pesa sempre di più;

7. 4 anni dalla vendita dei primi token; $TV(t5)$:
   L'80% del litio è stato prodotto e l'80% del token è stato venduto. Il Token Value è meno influenzato da caratteristiche come i diritti di produzione, la presenza in diversi paesi, la credibilità degli sviluppatori e dell'emittente e degli sponsor rispetto alle epoche precedenti e il valore del litio pesa sempre di più;

8. 5 anni dalla vendita dei primi token; $TV(t6)$:
   Il 100% del litio è stato prodotto e tutti i token sono stati venduti. A differenza delle epoche precedenti, il Token Value non è più influenzato da caratteristiche come i diritti di produzione, la presenza in diversi paesi, la credibilità degli sviluppatori e dell'emittente e degli sponsor. Invece, il prezzo è quasi completamente influenzato da quello del litio. I grafici di ogni epoca sono descritti in Figura 6.3.

Calcoliamo ora i CSF e di conseguenza il $TV(t_i)$ per ogni epoca, dove t_i è il numero dell'epoca corrente. Assegnamo a_1 a Pr, a_x a Pl, a_7 a Cr e a_y a P_o con

$$a_x = a_2 + a_3 + a_5 + a_6 + a_8 + a_{12} + a_{13}$$

e

$$a_y = a_4 + a_9 + a_{10} + a_{11}.$$

$TV(t)$ in t_1, t_2, t_3 :
durante la promozione globale, 300.000.000 (6% dell'offerta) token vengono venduti tramite vendita privata al prezzo di 0,027 $, permettendo di investire nelle attività del progetto con l'aspettativa che i token raggiungano il prezzo del litio di 5$ al kg. i) fase 1: promozione globale; a) Prezzo: 0.027$; b) monete emesse: 300.000.000(6%);
c) CSF:

- $a_1 = 0$: non esiste ancora nessuna materia prima estratta tramite le attività legate al progetto;

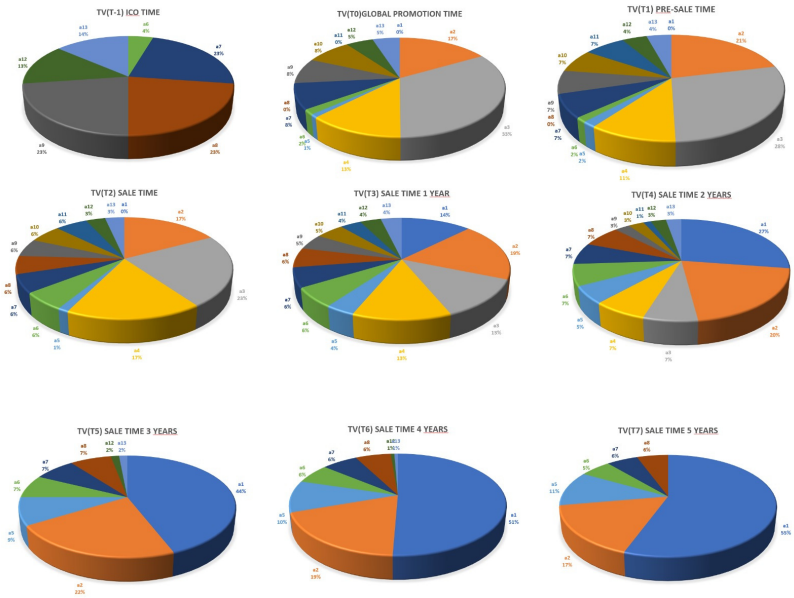

Figure 6.3: Evoluzione della composizione del token preso in esempio durante le diverse fasi del suo ciclo di vita.

- $a_2 = 0,07$: la piattaforma tecnologica è in fase di test;
- $a_3 = 0,2$: esiste un contratto legato ai diritti per la produzione di litio;
- $a_4 = 0,075$: presente solo in alcuni paesi;
- $a_5 = 0,1$: solo il 6% di token è pronto per essere venduto;
- $a_6 = 0,01$: è documentata solo la fase di sviluppo;
- $a_7 = 0,03$: buon sentiment verso il progetto, ma ci si aspetta ulteriori sviluppi;
- $a_8 = 0$: nessuna analisi prospettica da parte di esperti è stata eseguita in questa fase;
- $a_9 = 0,05$: il progetto è fondato da una società a responsabilità limitata registrata;
- $a_{10} = 0,05$: l'emittente è la società stessa;
- $a_{11} = 0$: nessuna valutazione riguardante il team di sviluppo e advisors;
- $a_{12} = 0,03$: progetto di alto livello in termini di originalità;
- $a_{13} = 0,03$: progetto di alto livello in termini di innovazione;

Table 6.1: CSF nella prima fase di vita del token.

a_1^0	a_2^0	a_3^0	a_4^0	a_5^0	a_6^0	a_7^0
0	0,07	0,2	0,075	0,1	0,01	0,03

a_8^0	a_9^0	a_{10}^0	a_{11}^0	a_{12}^0	a_{13}^0
0	0,05	0,05	0	0,03	0,03

$$\sum_{1}^{13} a_i^0 = 0.675 < 1.$$

Dalla somma dei pesi, si evince come il risultato sia molto minore di 1 e possiamo assumere che investire in un token con un buon sentiment, ma basato solo su un'idea è molto rischioso.

Assumiamo che il prezzo del litio, che nel 2020 si trovava a 5.52$ al kg, influenzi comunque Pr,Pl,Cr e Po in quanto le informazioni vengono sempre intrinsecamente inglobate nel prezzo:

$$TV_{Pr} = 5.52\$, TV_{Pl} = 0.5\$, TV_{Cr} = 3\$, TV_{Po} = 0.05\$.$$

e calcoliamo: $A_1 = 0; A_x = 0.10 + 0.2 + 0.1 + 0.01 + 0 + 0.03 + 0.03 = 0.47; A_7 = 0.03; A_y = 0.075 + 0.05 + 0.05 + 0 = 0.175;$ Calcoliamo il TV durante la fase di promozione globale:

$$TV_v = 0 * 5.52 + 0.47 * 0.5 + 0.03 * 3 + 0.175 * 0.05 = 0.33\$.$$

Durante la promozione globale, il Token Value (TV) è di 0.33$, e risulta quindi molto sottovalutato; il rischio, trattandosi di completa speculazione, è ovviamente che il progetto non venga portato avanti dagli sviluppatori e non mantenga le promesse fatte nel whitepaper portando il token a perdere di valore.

ii) Fase 2: pre-sale;

Durante la prevendita, vengono venduti 600.000.000 (12% dell'offerta) di pezzi attraverso una vendita pubblica al prezzo di 0,50$.

a) Prezzo: 0.50$;

b) monete emesse: 600.000.000(12%);

c) CSF:

- $a_1 = 0$: non esiste ancora nessuna materia prima estratta tramite le attività legate al progetto;
- $a_2 = 0,10$: la piattaforma tecnologica è in fase di test;
- $a_3 = 0,2$: esiste un contratto legato ai diritti per la produzione del litio;
- $a_4 = 0,1$: la promozione è stata allargata in diversi paesi;
- $a_5 = 0,1$: solo il 12% di token è pronto per essere venduto;
- $a_6 = 0,01$: è documentata solo la fase di sviluppo;
- $a_7 = 0,04$: buon sentiment verso il progetto, ma ci si aspetta ulteriori sviluppi;

- $a_8 = 0$: nessuna analisi prospettica da parte di esperti è stata eseguita in questa fase;
- $a_9 = 0,05$: il progetto è fondato da una società a responsabilità limitata registrata;
- $a_{10} = 0,05$: l'emittente è la società stessa;
- $a_{11} = 0,05$: buona valutazione degli advisors e del team di sviluppo;
- $a_{12} = 0,03$: progetto di alto livello in termini di originalità;
- $a_{13} = 0,03$: progetto di alto livello in termini di innovazione;

a_1^1	a_2^1	a_3^1	a_4^1	a_5^1	a_6^1	a_7^1
0	0,10	0,2	0,1	0,1	0,01	0,04

a_8^1	a_9^1	a_{10}^1	a_{11}^1	a_{12}^1	a_{13}^1
0	0,05	0,05	0,05	0,03	0,03

$$\sum_{1}^{13} a_i^1 = 0.76 < 1.$$

Dalla somma dei pesi,che è inferiore a 1, possiamo supporre che investire nel token in questo momento sia ancora rischioso.

$$TV_P = 5.52\$, TV_{Pl} = 1\$, TV_{Cr} = 3\$, TV_{Po} = 2\$,$$

quindi, al momento della pre-vendita abbiamo:

$$TV_v = 0 * 5.52 + 0.47 * 1 + 0,04 * 3 + 0,25 * 2 = 1.09\$.$$

Nella seconda epoca, il Token Value (TV) è 1.09$, si noti che il fair value del token è più alto rispetto alla fase precedente in quanto gli sviluppi proseguono e la fiducia degli investitori sale, nella prospettiva che il token si apprezzi durante il tempo e venga coperto dalle riserve di litio. Quella descritta è una price action

molto comune nelle monete lanciate tramite presale (Un esempio con storico sono i Launchpad di Binance). *iii*) Fase 3: vendita e produzione
In quest'epoca, abbiamo simulato il TES cambiando i valori a_i durante i diversi anni di produzione supponendo che il valore del litio al kg si mantenga fra i 5$ e i 20$. Le monete coniate partono dai 600.000.000 dall'epoca di pre-vendita fino ai 4.400.000.000 di token coniati dopo cinque anni.

a) Prezzo: da 0.50$ della presale; *b*) monete emesse: da 600.000.000 a 4.400.000.000(88%);
c) Parametri:

- a_1 =da 0 a 0,5: dal primo al quinto anno le riserve di litio a copertura del token e il suo relativo CSF crescono nel tempo;
- a_2 = da 0,07 a 0.15 : la piattaforma tecnologica è in fase di test all'inizio della vendita, e viene pubblicata durante il primo anno;
- a_3= da 0,2 a 0 : esiste un contratto legato ai diritti di produzione del litio, negli anni successivi questo CSF influenza il prezzo sempre di meno;
- a_4= da 0,1 a 0,15 : durante gli anni avviene la quotazione sui maggiori exchange;
- a_5=da 0,1 a 0 : solo il 12% della supply di token è pronto per essere venduto all'inizio della vendita, dopo 5 anni tutti i token saranno stati venduti;
- a_6= da 0.01 a 0,05 : dopo 5 anni tutte le attività sono documentate;
- a_7=0,05 : ottimo sentiment di mercato;
- $a_8 = 0,05$: gli esperti hanno analizzato il progetto e garantiscono un'ottima prospettiva;
- a_9= 0,05: ottima credibilità dei fondatori;
- a_{10}= 0,05: l'emittente è la società founder del progetto; con il passare del tempo questo CSF diventa sempre meno importante;

- a_{11}= 0,05: ottima valutazione degli advisors e del team; negli anni successivi questo CSF diventa sempre meno importante;
- a_{12}= 0,03: progetto di alto livello in termini di originalità; con il passare del tempo questo CSF diventa sempre meno importante;
- a_{13}= 0,03 : progetto di alto livello in termini di innovazione; negli anni successivi questo CSF diventa sempre meno importante;

a_1^x	a_2^x	a_3^x	a_4^x	a_5^x	a_6^x
$0-0,5$	$0.07-0,15$	$0,2-0$	$0.1-0,15$	$0,1-0$	$0.01-0,05$

a_7^x	a_8^x	a_9^x	a_{10}^x	a_{11}^x	a_{12}^x	a_{13}^x
0.05	$0,05$	$0,05$	$0,05$	$0,05$	$0,03$	$0,03$

$$\sum_{i=1}^{13} a_i^x = [0.79\ ,\ 1.16],$$

dove $0 < x \leq 5$.

Dalla somma dei pesi, è possibile notare come il risultato sia inferiore a 1 alla fine di prevendita, ma superiore alla fine della produzione. Ciò significa che acquistare all'inizio della vendita potrebbe essere un investimento con buon rapporto risk-reward, mentre alla fine della produzione investire nel token è più sicuro, ma con ritorni inferiori in quanto gran parte del valore del token rispecchia l'asset fisico sottostante.

Infine supponiamo di trovarci in due scenari in questa epoca. Nel primo scenario, il litio ha un valore di 5$ per kg: Alla fine della produzione, gli investitori comprendono appieno che il valore del token è per lo più collegato al valore del litio. Quindi, supponiamo che i max TV siano:

$$MaxTV_P = 5\$, MaxTV_{Pl} = 5.50\$,$$

$$MaxTV_{Cr} = 6\$, MaxTV_{Po} = 7.45\$.$$

Nel secondo scenario, il litio ha un valore di 22$ per kg. Supponiamo che i max TV siano:

$$MaxTV_{P} = 22\$, MaxTV_{Pl} = 20.50\$,$$

$$MaxTV_{Cr} = 21\$, MaxTV_{Po} = 22.45\$$$

TV_{Po}. Abbiamo simulato i CSF e il relativo TV nelle tre epoche descritte in precedenza: l'anno 0 è la promozione globale; il 5° è l'ultimo anno di produzione del litio. Il risultato della simulazione è mostrato nella Figura 6.4.

Dalla figura 6.4 è chiaro come il prezzo del token inizialmente sia interamente guidato dalla speculazione anche se il mercato prezza già in parte il valore del litio a copertura. Quando il litio viene prodotto durante gli anni attraverso l' attività lavorativa, il valore del token inizierà a tendere a quello del litio fin quando non lo rispecchierà appieno alla fine del 5° anno quando le attività saranno concluse.

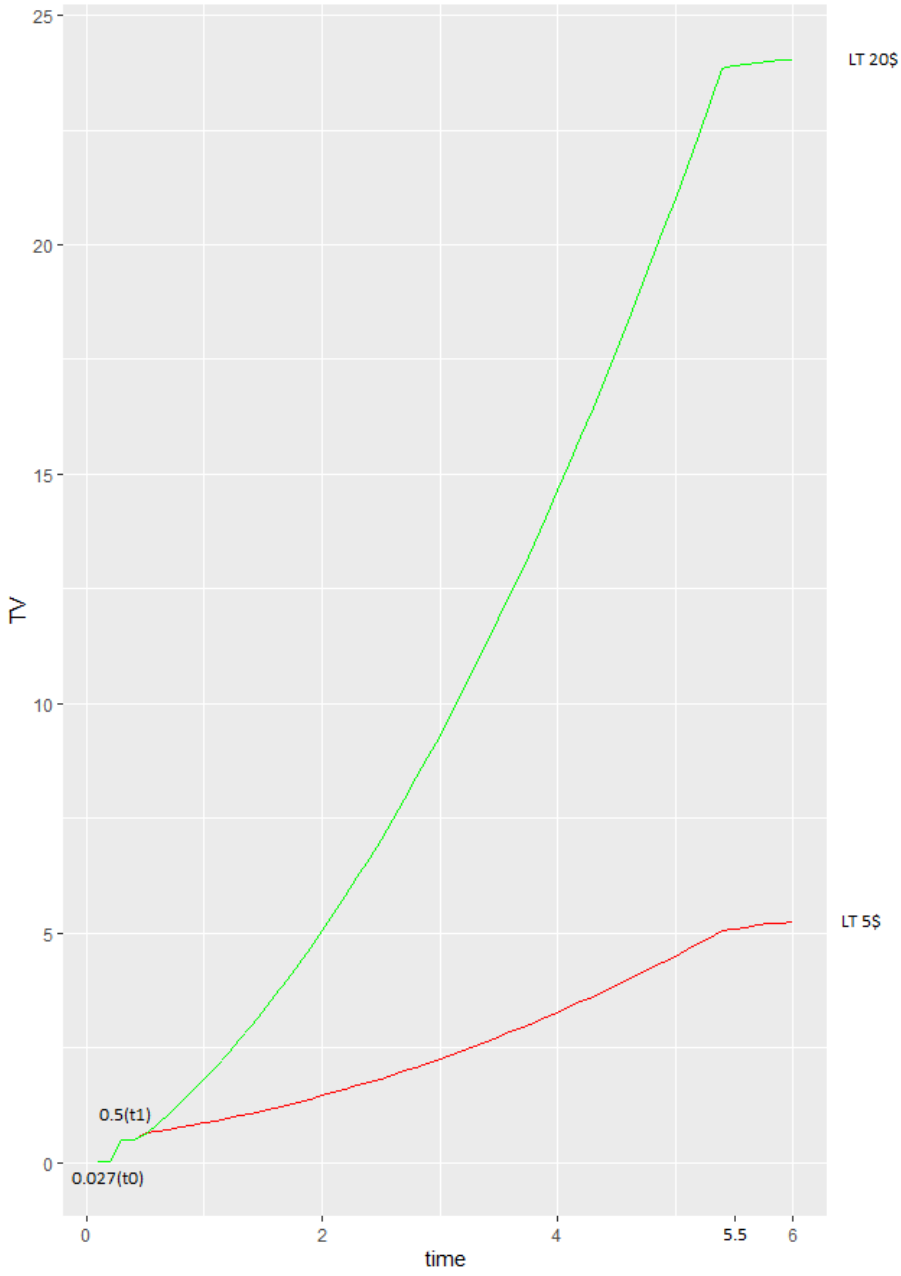

Figure 6.4: Evoluzione di TV(t) con t anni di sviluppo.

Part III

DISCUSSIONE

Nel capitolo 7 utilizziamo la teoria dell'info-incertezza e del best price per valutare asset utilizzando dati e contesti reali, approfondendo e rafforzando allo stesso tempo la validazione delle teorie presentate in precedenza. In questa parte inoltre viene rafforzata la sicurezza della soluzione proposta nel capitolo 4. L'utilizzo di dispositivi IoT durante la generazione dei dati utilizzati pone in serio pericolo la sicurezza di dati altamente sensibili. Per questo motivo, in questa parte proponiamo una soluzione di autenticazione biometrica che permette di ottenere firme autenticate da utilizzare durante lo svolgimento del lavoro. Infine viene analizzata la scalabilità della soluzione proposta, immaginando l'esecuzione di un ciclo di lavoro nel mondo reale, analizzando le varie piattaforme Smart Contract, i relativi costi e la sostenibilitá a lungo termine attraverso una proiezione futura riguardante l'utilizzo delle delle stesse.

7

VALIDAZIONE DEI MODELLI

Al fine di validare i modelli descritti nei capitoli 5 e 6, in [36], è stata sviluppata una rete neurale per stimare i pesi del bast price associato alla funzione di accadibilità a_3 di [35] in un contesto dove è possibile testare il modello su dati reali e con uno storico ben definito.

Sulla base delle ipotesi e delle definizioni mostrate nei capitoli precedenti, l'approccio proposto è stato testato su un caso studio reale riguardante la valutazione di asset appartenenti al mercato sportivo. Tale problema è ideale per studiare la rilevanza delle opinioni in condizioni di incertezza e incompletezza delle informazioni, poiché la previsione del prossimo costo di trasferimento di un atleta è soggetta a diverse fonti di speculazione.

Questo ambito è stato scelto in quanto, grazie all'ampia disponibilità di dati, è possibile valutare il modello proposto sia in condizioni di info-completezza, cioè quando il parere riguardante la Probabilità è disponibile, sia di info-incompletezza, cioè quando il parere relativo alla Probabilità è completamente carente, lasciando intatto l'ambiente definito in questo elaborato e cioè la valutazione di un asset in un mercato alla quale partecipano diversi operatori.

Facendo riferimento alla funzione TV (o, in questi caso, di miglior prezzo/fair value) definita in precedenza, si intende stimare i pesi α_1, α_2, α_3 e α_4, relativi alle opinioni associate rispettivamente a Probabilità, Plausibilità, Credibilità e Possibilità, estraendo dati da fonti diverse.

Date le occorrenze delle quattro opinioni di cui sopra e il relativo ground truth, è possibile approssimare la funzione TV, cioè la migliore opinione, definendo un modello di apprendimento addestrato alla migliore approssimazione dei pesi. Questo approccio consente di ottenere la rilevanza delle opinioni in condizioni di info-incompletezza, insieme alla stima più promettente riguardo un dato evento o fenomeno.

Per raggiungere tale obiettivo, in [36] viene proposto il modello in Figura 7.1, in cui un' Artificial Neural Network (ANN) prende in input quattro features che corrispondono alle opinioni associate alla Probabilità, Plausibilità, Credibilità, Possibilità e un insieme arbitrario di altri attributi rilevanti. Ciascuna delle opinioni viene estratta da una fonte di informazioni dedicata, nonché dal set aggiuntivo di attributi.

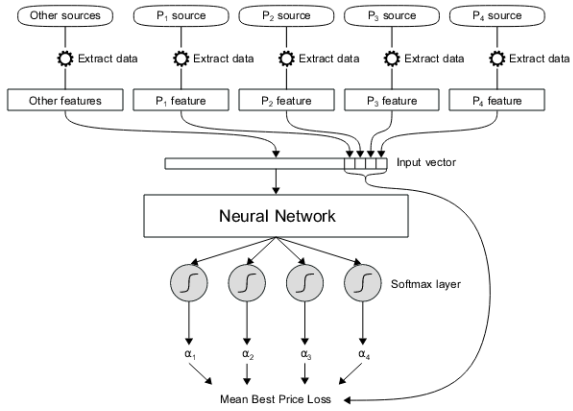

Figure 7.1: Il modello di apprendimento che permette la stima dei pesi e quindi dell'opinione più rilevante.

Il modello produce quattro pesi estratti dai componenti di un softmax layer per ridurre al minimo una loss function personalizzata, chiamata *Mean Best Price Loss*, che è stata definita come

$$L_{MBP} = \frac{1}{N} \sum_{i=1}^{N} |P_{0_i} - (\alpha_{1_i} P_{1_i} + \alpha_{2_i} P_{2_i} + \alpha_{3_i} P_{3_i} + \alpha_{4_i} P_{4_i})|,$$

dove N è la batch size, mentre P_{0_i}, P_{1_i}, P_{2_i}, P_{3_i} e P_{4_i} sono rispettivamente il ground truth e le opinioni di Probabilità, Plausibilità, Credibilità e Possibilità relative alla i-esima occorrenza. Allo stesso modo, i pesi α_{1_i}, α_{2_i}, α_{3_i} e α_{4_i} sono i componenti del softmax layer relativo alla i-esima occorrenza. Questo permette all'ANN di apprendere i parametri che forniscono i pesi ottimali per ridurre al minimo la differenza tra la migliore valutazione del prezzo, cioè la stima della migliore opinione, ed il ground truth. La riduzione è ottenuta attraverso la loss function, la quale può essere considerata come la Mean Absolute Error Lost, in cui

il secondo termine della sottrazione rappresenta la previsione del modello proposto data una certa istanza di caratteristiche. Nella sezione seguente viene valutato l'errore MBPE in base alle variazioni dei sottoinsiemi di opinioni. per valutare le prestazioni del modello in condizioni di info-incompletezza, ad esempio, la caratteristica relativa alla Probabilità è impostata a zero, in modo da testare il caso in cui una decisione dovrebbe essere presa quando solo i pareri di Plausibilità, Credibilità e Possibilità sono pienamente o parzialmente disponibili.

7.1 ESPERIMENTI SU UN CASO DI STUDIO: VALUTAZIONE IN AMBITO SPORTIVO

Come discusso nella sezione precedente, al fine di fornire un caso di studio su cui verificare l'efficacia dei modelli proposti, è stata costruita un' ANN che permette di stimare il costo di trasferimento di un atleta a partire dai suoi attributi e dalle opinioni associate alla Probabilità, Plausibilità, Credibilità e Possibilità, estratte dal web. La presenza di diversi simulatori, forum in cui gli utenti esperti definiscono il valore degli sportivi, negoziatori e speculatori permettono di raccogliere una vastissima mole di dati da utilizzare per la validazione.

Formalmente, nel presente caso d'uso le opinioni sono definite come $P_1, P_2, P_3, P_4 \in \mathbb{R}$, poiché i prezzi sono quantità unidimensionali.

Per la validazione, sono state scelte quattro fonti di informazione:

1. Valutazione degli atleti sul sito Transfermarkt, la principale fonte di informazione riguardante l'ambito calcistico; associando l'informazione al parere della Probabilità ed evidenza scientifica [32, 67], le informazioni raccolte sono relative all'anno 2021/ 2022;

2. valutazione degli atleti sul simulatore Football Manager per l'opinione relativa alla Plausibilità; Football Manager è il simulatore calcistico più completo per quanto riguarda i dati siccome del personale specializzato per ogni società si occupa di analizzare e costruire i relativi dataset, per questo motivo associamo questi dati all'opinione esperta;

3. per l'opinione relativa alla credibilità viene effettuata la sentiment analysis sulla descrizione degli atleti presenti sul portale Wikipedia. Il sentiment viene normalizzato combinando l'offerta e cioè l'ammontare che la società richiede per la vendita del proprio tesserato. Questa informazione è stata prelevata dal database di Football Manager;

4. valutazione presente sul simulatore FIFA21 per l'opinione relativa alla Possibilità.

Ogni fonte di informazione è stata normalizzata in un valore di prezzo, ad esempio, le valutazioni di un atleta su Transfermarkt (P), Football Manager (Pl) e FIFA21 (Po) potrebbero essere rispettivamente di 16, 15 e 25 milioni di euro, mentre la società proprietaria del cartellino richiede 20 milioni di euro, combinata con un punteggio di sentiment di 0,95 (Cr).

L'estrazione dei dati è stata eseguita attraverso la tecnica dello scraping, sviluppando quattro script distinti in linguaggio Python. Ogni script permette di raccogliere i dati dalle pagine web relative ai principali campionati calcistici europei, ovvero Serie A, La Liga, Premier League, Ligue 1 e Bundesliga. Il ground truth è stato acquisito recuperando il costo dei trasferimenti dalla stessa fonte scelta per raccogliere le opinioni relative alla Probabilità (Transfermarkt) in quanto dispone delle informazioni complete sui trasferimenti. Per quanto riguarda gli attributi dei giocatori, si è deciso di acquisire le loro caratteristiche dalla stessa fonte associata alle opinioni relative alla Possibilità (FIFA21). Il processo di estrazione dei dati può essere riassunto, per ogni atleta, definendo l'esecuzione parallela dei seguenti passi:

- estrazione del valore di mercato ottenuto da Transfermarkt, un anno prima del successivo trasferimento, per ottenere la Probabilità P_{1_i};

- estrazione del valore di mercato ottenuto da Football Manager, un anno prima del successivo trasferimento, per ottenere la Plausibilità P_{2_i};

- esecuzione del calcolo del punteggio di sentiment dalla descrizione di Wikipedia relativa allo sportivo, ponderando il risultato con l'offerta della società per la vendita del

tesserato estratta da Football Manager (un anno prima del trasferimento successivo), per ottenere la Credibilità P_{3_i};

- estrazione del valore di mercato ottenuto da FIFA21, un anno prima del successivo trasferimento, per ottenere la Possibilità P_{4_i};
- estrazione degli attributi ottenuti da FIFA21, un anno prima del successivo trasferimento, per ottenere altre features da dare in input alla rete neurale.

7.1.1 *Sentiment analysis*

La sentiment analysis, come detto, è stata condotta sui testi acquisiti dalle pagine di Wikipedia relative agli atleti.

Il modello di classificazione del sentiment testuale utilizzato è basato sul BERT (Bidirectional Encoder Representations from Transformers) [21] e addestrato sull'IBM Claim Stance Dataset [33, 34]. Il classificatore riceve una stringa di testo come input, e restituisce un punteggio nell'intervallo $[0, 1]$, in cui 0 rappresenta emozioni negative ed 1 positive.

Il modello è caratterizzato da due fasi operative:

- *pre-training*, nella quale avviene l'addrestramento del modello su più compiti, attraverso un approccio non supervisionato;
- *fine-tuning*, che permette di ottimizzare i parametri presenti nel task precedente alla sentiment analysis, attraverso un approccio supervisionato.

In [36] è descritto in dettaglio il training di questo classificatore, il quale raggiunge in termini di accuratezza, il 94% e fornisce una buona stima del sentiment in un testo.

7.1.2 *Dataset*

Come anticipato, i dati sono stati estratti dalle fonti Transfermarkt, FIFA21 e Football Manager disponibili sul web attraverso la tecnica dello scraping. In particolare, Transfermarkt è stata la prima fonte esplorata. Il processo di scraping ha permesso

di estrarre da Transfermarkt i nomi degli atleti, i relativi valori di mercato e i costi di trasferimento nelle sessioni di negoziazione precedenti nel caso in cui sia avvenuto un trasferimento. Dopodiché per ogni nome estratto, sono state prese in considerazione le fonti relative a FIFA21 e Football Manager per ottenere attributi e valori riguardanti le opinioni di Plausibilità, Credibilità e Possibilità.

Parallelamente, le pagine di Wikipedia riguardanti i nomi estratti sono state elaborate attraverso il classificatore per la sentiment analysis.

I costi di trasferimento dei giocatori forniti da Transfermarkt (min = 0,1 milioni di euro, max = 125 milioni di euro) sono stati utilizzati come ground truth per i processi di training e test in quanto è il valore dello sportivo sulla quale le società negozianti convergono. In questo caso sono stati presi in considerazione un totale di 398 occorrenze di dati di atleti trasferiti.

La figura 7.2 mostra le distribuzioni di frequenza doppia delle caratteristiche di Probabilità, Plausibilità, Credibilità e Possibilità ottenute costruendo il dataset.

La Figura, sulla prima diagonale, mostra le distribuzioni di densità relative alle opinioni. Si può notare che le osservazioni, espresse in milioni di euro, relative a Credibilità ($m = 13,33$, $SD = 29,47$) e Possibilità ($m = 26,19$, $SD = 101,72$) presentano la varianza più alta. La distribuzione riguardante la Plausibilità, invece, è caratterizzata dalla varianza più bassa ($m = 6,62$, $SD = 10$).

I pareri di Plausibilità, che fanno riferimento alle opinioni degli esperti, tendono ad occupare un'area definita di ipotesi caratterizzate da minore incertezza[1]. Man mano che il campo di opinioni considerato si espande verso le aree del sentiment e altre fonti di informazione meno rilevanti, l'incertezza sulla valutazione dei giocatori aumenta, poiché le opinioni sono più eterogenee e dettate dalle emozioni.

Le altre features considerate sono: abilità, età ($m = 24$, $SD = 3,66$, anni), ruolo, salario ($m = 14,53$, $SD = 36,18$, milioni di euro), altezza ($m = 182,48$, $SD = 7,13$, cm), peso ($m = 76,81$, $SD = 7,74$, kg), piede preferito e ruoli preferiti. Le caratteristiche non numeriche, come la posizione o il piede preferito, sono state

1 Questo è visibile anche negli esempi mostrati nel capitolo precedente

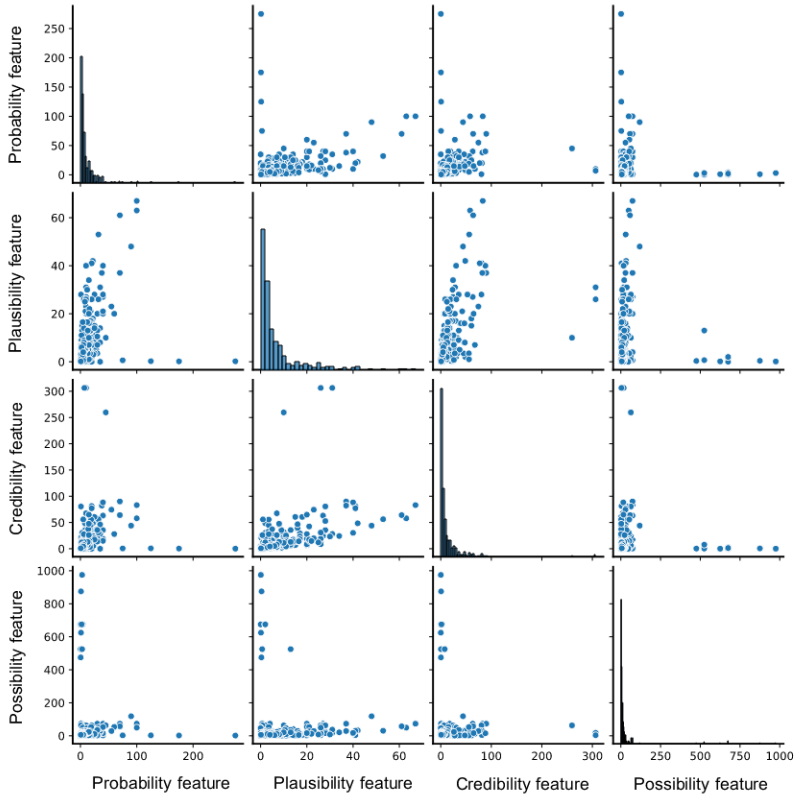

Figure 7.2: Distribuzioni di frequenza doppia delle caratteristiche di Probabilità, Plausibilità, Credibilità e Possibilità.

enumerate. Nella Figura 7.3, vengono mostrati i boxplot relativi alle distribuzioni delle abilità. Come organizzato in FIFA21, le statistiche per le abilità sono specificate come numeri interi nell'intervallo $[0, 100]$.

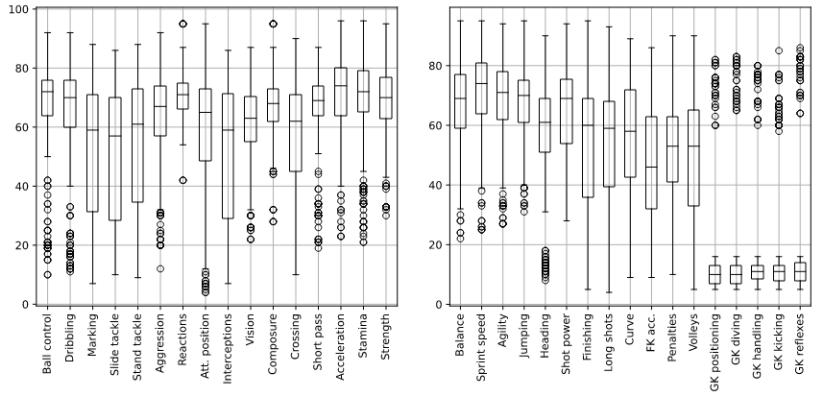

Figure 7.3: Distribuzioni relative alle abilità degli sportivi.

Gli attributi che presentano la più alta dispersione sono *making*, *slide tackle* e *interceptions*, mentre quelli caratterizzati dalla più alta quantità di valori anomali sono le dimensioni riservate ai portieri, ovvero *GK positioning*, *GK diving*, *GK handling*, *GK kicking* e *GK reflexes*. Queste anomalie sono causate della ovvia presenza limitata di portieri nel dataset. Nella prossima sezione analizziamo i dati ottenuti dalla rete neurale. I dettagli dell'implementazione della rete sono descritti in [36].

7.1.3 *Risultati*

La tabella 7.1 mostra l'errore medio del miglior prezzo, espresso in milioni di euro, relativo a diversi insiemi di opinioni dopo la features selection. Fatta eccezione per la valutazione del modello considerando individualmente le opinioni di Plausibilità, Credibilità e Possibilità, è stato riscontrato un aumento significativo delle prestazioni dopo la features selection. Tuttavia, l'insieme delle dimensioni relative a Credibilità e Possibilità aumenta l'incertezza nella stima del prezzo.

Model evaluation considering the most important features		
Chosen opinions	**Mean Best Price Error (mln)**	**Normalized error (%)**
Probability, Plausibility, Credibility, Possibility	1.01	0.72
Plausibility, Credibility, Possibility	2.25	1.72
Probability, Plausibility	0.91	0.64
Credibility, Possibility	2.41	1.85
Probability	0.86	0.61
Plausibility	2.48	1.90
Credibility	2.34	1.79
Possibility	2.87	2.21

Table 7.1: Performance nella predizione del miglior prezzo considerando i diversi set di opinioni.

In questo caso di studio le migliori prestazioni sono state ottenute considerando l'insieme composto da Probabilità e Plausibilità, nonché la Probabilità considerata individualmente. I risultati identificano differenze sostanziali tra i test decisionali condotti utilizzando i diversi insiemi di opinioni.

Per il problema affrontato, riguardante la previsione del costo di trasferimento dei giocatori, abbiamo ottenuto che l'errore minore, in fase di previsione, si raggiunge utilizzando la decisione associata alla Probabilità (0,86 MBPE, 0,61% di errore normalizzato). Le peggiori prestazioni, tuttavia, sono state trovate individualmente considerando le opinioni di Plausibilità, Credibilità e Possibilità.

Questo risultato corrisponde alla teoria prodromica della decisione e del ragionamento in condizioni di incertezza e info-incompletezza [35] presentata nel capitolo 5, poiché l'ottenimento, l'uso ed il confronto con un ground truth, di opinioni particolarmente rilevanti riguardanti la sfera della Probabilità, rappresenta una condizione di info-completezza.

Al contrario, eliminando l'evidenza diretta, cioè trascurando l'opinione di Probabilità, si ottiene un errore maggiore in fase di previsione. La decisione in condizioni di incompletezza delle informazioni può introdurre una maggiore incertezza nella fase decisionale, in quanto la mancanza di prove dirette costringe il decisore a valutare le opinioni derivanti da esperti, sentiment e di minor rilevanza.

Il problema di previsione affrontato nel presente studio può essere ricondotto alla caratterizzazione gerarchica del *modello di sovrapposizione con shift basato sulla probabilità* ripreso nel capi-

tolo 5. La gerarchia decisionale è rispettata dai risultati basati sull'aumento dell'errore in funzione del tipo di opinione valutata; per il problema attuale, la priorità riflette l'ordine crescente dell'errore medio del miglior prezzo sul set di test: i) Probabilità (0,86 MBPE, 0,61 % di errore normalizzato), ii) Credibilità (2,34 MBPE, 1,79 % di errore normalizzato); iii) Plausibilità (2,48 MBPE, 1,90 % di errore normalizzato); iv) Possibilità (2,87 MBPE, 2,21 % di errore normalizzato). Seguendo la logica nell'approccio del *modello di sovrapposizione con shift basato sulla probabilità*, per ogni giocatore, viene presa la decisione finale più promettente sulla base dei seguenti passaggi:

1. la decisione finale viene presa in base al parere di Probabilità, se disponibile, con un errore atteso pari a 0,86 MBPE (0,61% di errore normalizzato), proporzionale al peso α_1 stimato dal modello;

2. se il parere di Probabilità non è disponibile, la decisione finale viene presa in base al parere di Credibilità, se disponibile, con un errore previsto di 2,34 MBPE (1,79% di errore normalizzato), in proporzione al peso α_3 stimato dal modello;

3. se i pareri di Probabilità e Credibilità non sono disponibili, la decisione finale viene presa sulla base del parere di Plausibilità, se disponibile, con un errore atteso di 2,48 MBPE (1,90% di errore normalizzato), in proporzione al peso α_2 stimato dal modello;

4. se i pareri di Probabilità, Credibilità e Plausibilità non sono disponibili, la decisione finale viene presa in base al parere della Possibilità, se disponibile, con un errore previsto di 2,87 MBPE (2,21% di errore normalizzato), in proporzione al peso α_4 stimato dal modello.

Per ottimizzare le prestazioni in condizioni di info-incompletezza, come nell'ambito finanziario oggetto di questo elaborato, è possibile aggiungere un'estensione ai suddetti passaggi che prevedono il contributo simultaneo di più opinioni. In particolare, nel caso in cui siano disponibili i pareri di Plausibilità, Credibilità e Possibilità, è consigliabile prendere la decisione considerando la

somma ponderata dei relativi contributi, in quanto tale soluzione prevede un Errore Medio del best price inferiore a quello che si otterrebbe considerando solo il parere relativo alla Credibilità.

Basandoci sull'errore medio stimato dal modello, possiamo ottenere i pesi: stabiliamo $x_{min} = 0$ e $x_{max} = 2.5$.
calcoliamo il valore dell'errore normalizzato α_{n_i} su x_{min} ed x_{max}:

$$\alpha_{n_i} = 1 - \frac{(x_i - x_{min})}{(x_{max} - x_{min})} =$$

- $\alpha_{n_1} = 1 - \frac{0.61}{2.5} = 0.76$
- $\alpha_{n_2} = 1 - \frac{1.90}{2.5} = 0.24$
- $\alpha_{n_3} = 1 - \frac{1.79}{2.5} = 0.28$
- $\alpha_{n_4} = 1 - \frac{2.21}{2.5} = 0.12$

Calcoliamo $\alpha_{tot} = \sum_{i=1}^{4} \alpha_i = 1.4$.
e infine i pesi:

$$\alpha_i = \frac{\alpha_{n_i}}{\alpha_{tot}} =$$

- $\alpha_1 = \frac{0.76}{1.4} = 0.54$;
- $\alpha_2 = \frac{0.24}{1.4} = 0.17$
- $\alpha_3 = \frac{0.28}{1.4} = 0.20$
- $\alpha_4 = \frac{0.12}{1.4} = 0.09$

Estendendo la logica dell'approccio del *modello di sovrapposizione con shift basato sulla probabilità* alla valutazione congiunta di più pareri, la decisione finale più promettente viene ottimizzata sulla base dei seguenti passaggi:

1. la decisione finale viene presa in base al parere di Probabilità, se disponibile, con un errore atteso pari a $0,86$ MBPE (0,61% di errore normalizzato), proporzionale al peso α_1 stimato dal modello;

2. se il parere di Probabilità non è disponibile, la decisione finale viene presa sulla base dei pareri di Plausibilità, Credibilità e Possibilità, se disponibili, con un errore atteso pari

a 2,25 MBPE (1,72% di errore normalizzato), in base alla somma delle opinioni ponderate per i valori α_2, α_3, e α_4 identificati dal modello;

3. se i pareri di Probabilità e Plausibilità non sono disponibili, la decisione finale viene presa sulla base del parere di Credibilità, se disponibile, con un errore atteso di 2,48 MBPE (1,90% di errore normalizzato), in proporzione al peso α_3 stimato dal modello;

4. se i pareri di Probabilità, Plausibilità e Credibilità non sono disponibili, la decisione finale viene presa in base all'opinione di Possibilità, se disponibile, con un errore previsto di 2,87 MBPE (2,21% di errore normalizzato), in proporzione al peso α_4 stimato dal modello.

È inoltre interessante notare che, a differenza del modello teorico descritto nel capitolo 5 e del TES, in cui la Plausibilità era caratterizzata da una priorità più alta della Credibilità, in questo caso è vero il contrario. Per quanto riguarda il problema della valutazione degli sportivi, l'opinione degli esperti è più debole rispetto al sentiment.

Per fornire un esempio visivo delle prestazioni di previsione, la Figura 7.4 mostra un confronto tra il ground truth e il miglior prezzo previsto dal modello addestrato con il sottoinsieme delle features più importanti (1,01 MBPE, 0,72 % di errore normalizzato).

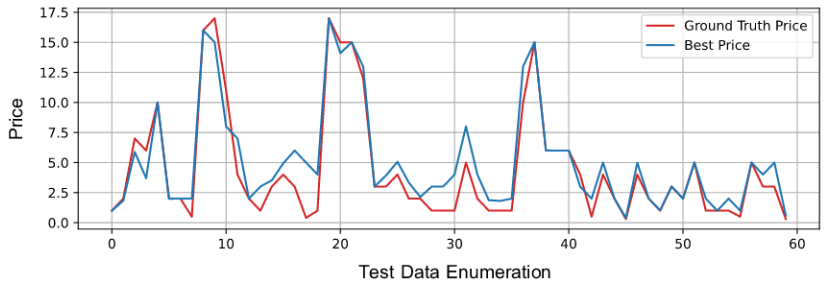

Figure 7.4: Confronto tra il ground truth e la migliore previsione del prezzo ottenuta attraverso il modello proposto, dopo la selezione delle funzionalità, in base alle opinioni di Probabilità, Plausibilità, Credibilità e Possibilità.

Al fine di validare l'evidenza statistica dei risultati ottenuti, è stato eseguito il test di ipotesi riguardanti le differenze tra i modelli addestrati attraverso i diversi insiemi di opinioni. Al fine di calcolare i P-value, sono stati generati duemila set di bootstrap dalle previsioni ottenute valutando i modelli sul set di test. Ogni set generato viene confrontato con il ground truth per calcolare un punteggio MBPE, che a sua volta viene sottratto con il punteggio MBPE ottenuto valutando un altro modello. Per ogni coppia di modelli, è stata generata una distribuzione delle differenze e ne viene calcolato il P-value. La tabella 7.2 mostra i risultati dell'analisi considerando 0.05 come soglia di significatività.

P-values of the pair differences								
	P_1, P_2, P_3, P_4	P_2, P_3, P_4	P_1, P_2	P_3, P_4	P_1	P_2	P_3	P_4
P_1, P_2, P_3, P_4	> 0.05	0.002	0.29	< 0.0001	0.08	< 0.0001	< 0.0001	< 0.0001
P_2, P_3, P_4	0.002	> 0.05	< 0.0001	0.512	< 0.0001	0.1	0.67	0.067
P_1, P_2	0.29	< 0.0001	> 0.05	< 0.0001	0.620	< 0.0001	< 0.0001	< 0.0001
P_3, P_4	< 0.0001	0.512	< 0.0001	> 0.05	< 0.0001	0.8	1.204	0.04
P_1	0.08	< 0.0001	0.620	< 0.0001	> 0.05	< 0.0001	< 0.0001	< 0.0001
P_2	< 0.0001	0.1	< 0.0001	0.8	< 0.0001	> 0.05	0.50	0.23
P_3	< 0.0001	0.67	< 0.0001	1.204	< 0.0001	0.50	> 0.05	0.109
P_4	< 0.0001	0.067	< 0.0001	0.04	< 0.0001	0.23	0.109	> 0.05

Table 7.2: Test di significatività statistica dei risultati riguardanti la differenza nella predizione del best price utilizzando coppie di opinioni.

I risultati mostrano che, per il problema affrontato nel presente studio, non vi è alcuna significatività statistica nell'adottare tutte e quattro le opinioni invece di considerare solo la Probabilità; per ottenere prestazioni ottimali, ci sono forti prove statistiche che validano la considerazione della Probabilità, oppure della stessa in combinazione con Plausibilità, Credibilità e Possibilità. Inoltre, la scelta di considerare sia la Credibilità che la Possibilità è significativamente migliore rispetto alla sola Possibilità. Per quanto riguarda altre coppie di ipotesi, non è stata trovata alcuna ulteriore significatività statistica.

I risultati ottenuti attraverso la sperimentazione del modello proposto applicato a questo ambito dimostrano che le opinioni relative a Plausibilità, Credibilità e Possibilità non sono utili in condizioni di info-completezza. Sono state trovate forti evidenze nell'utilizzo della sola Probabilità al fine di prendere la decisione

finale. Questo risultato è coerente con [67] e [32], in quanto il crowdsourcing, dall'introduzione di Transfermarkt, è diventato la principale fonte per la valutazione degli atleti. Al contrario, le opinioni riguardanti Plausibilità, Credibilità e Possibilità sono essenziali in condizioni di incompletezza delle informazioni; in questo contesto, la decisione ottimale può essere raggiunta adottando un insieme arbitrario di opinioni, ad eccezione della coppia Credibilità-Possibilità, che è stata trovata più promettente della Possibilità considerata esclusivamente.

Attraverso questi esperimenti, abbiamo quindi dimostrato come in condizioni di info-incertezza, come nel caso dell'ambito finanziario e della valutazione di asset, sia necessario considerare altre fonti di informazione rispetto alla sola evidenza scientifica in quanto tutti gli attori che partecipano al mercato modellano il prezzo di un asset.

8

SCALABILITÀ E SICUREZZA

8.1 VALUTAZIONE DELLE PIATTAFORME E SCALABILITÀ

Dopo la fase di progettazione e sviluppo degli Smart Contract, dobbiamo distribuirli su una rete blockchain principale al fine di utilizzarli in un ambiente reale.

In letteratura, la scalabilità della piattaforma è un problema molto sottovalutato. Tutti i lavori pubblicati al tempo della scrittura distribuiscono i loro Smart Contract su Ethereum [9]. Questa piattaforma soddisfa solo due caratteristiche del trilemma della blockchain: gode di buona sicurezza e decentralizzazione[1] a discapito della scalabilità che genera costi di transazione molto elevati durante la congestione del network.

Durante il 2021, i costi del gas di Ethereum si aggiravano in media intorno ai 100 gwei[2], questo significa che una transazione costava in media 40$ in Ether.

Per quanto in letteratura possa essere accettabile non dare peso ai costi di transazione, un sistema simile a quello descritto nel capitolo 4 risulta inutilizzabile nel mondo reale.

Alla luce della discussione di cui sopra, abbiamo analizzato diverse piattaforme blockchain EVM per la distribuzione della nostra soluzione. Le piattaforme analizzate sono mostrate nella tabella 8.1.

In particolare siamo partiti dall'ipotesi della distribuzione della soluzione su una piattaforma centralizzata e passando poi attraverso l'ipotesi di utilizzo di diverse soluzioni decentralizzate EVM compatibili. La nostra analisi si è conclusa come segue:

- archiviazione file su cloud: con una soluzione centralizzata possiamo beneficiare di alta velocità e scalabilità, ma l'affidabilità della soluzione è molto bassa. Questo tipo di soluzione non è in grado di certificare il dato in quanto

1 Con il passaggio a PoS anche la sicurezza e la decentralizzazione ne hanno risentito

2 1 gwei=0,000000001 ETH per passo computazionale

Table 8.1: Abbiamo preso in considerazione diverse piattaforme a Smart Contract EVM compatibili. Per quanto riguarda il nostro obiettivo, la sicurezza e l'anti-manipolabilità delle informazioni sono imprescindibili. Abbiamo escluso la tecnologia Cloud File Storage dalla nostra scelta proprio per la mancanza di queste caratteristiche. Ethereum è la piattaforma più utilizzata e sicura, pecca in scalabilità, risultando non sostenibile dal punto di vista dei costi. Quindi, valuteremo nella tabella 8.2 e 8.3 alcune piattaforme alternative EVM compatibili cercando di massimizzare il rapporto sicurezza/costi in base ad una prospettiva sull'utilizzo futuro di una determinata blockchain. Nella presente sono analizzate le caratteristiche di ogni soluzione.

Type	**Speed**	**Security**	**Scalability**	**Decentralization**	**Anti-Manipulability**	**EVM**
Cloud File Storage	High	Low	High	No	Low	No
Ethereum	Low	High	Low	High	High	Yes
Binance Smart Chain	High	Medium	High	Low	High	Yes
Fantom	High	Medium	High	Medium	High	Yes
xDai	High	Medium	High	Medium-Low	High	Yes

manipolabile ed è soggetta al problema del *single point of failure*;

- Ethereum: è la soluzione più sicura e decentralizzata ma, come discusso sopra, non risulta utilizzabile in produzione a causa della sua scarsa scalabilità e degli alti costi di processamento delle transazioni;

- Binance Smart Chain: è la seconda piattaforma di Smart Contract più utilizzata dopo Ethereum. Risolve il problema della scalabilità grazie ad un algoritmo di consenso chiamato PoA risulta più sicuro di una soluzione completamente centralizzata. Il network rimane comunque molto centralizzato a causa della limitazione del numero di validatori dei blocchi. Essi al momento della scrittura sono 26 di cui la maggior parte è legata a Binance[3].

3 l'ente che ha svluppato la Binance Smart Chain

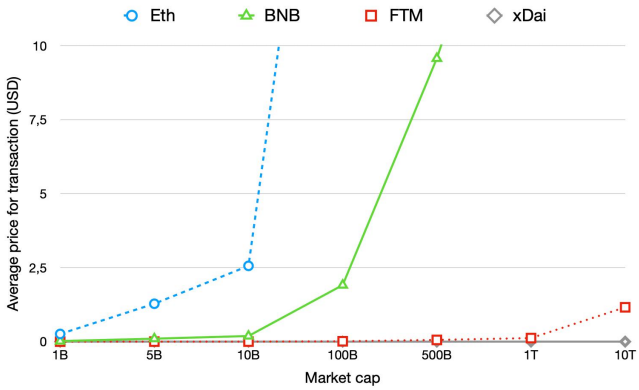

Figure 8.1: Proiezione in Tabella 8.3 graficata.

- Fantom[4] e xDai: si tratta di altre due blockchain compatibili con EVM che utilizzano la PoS per la validazione delle transazioni. In particolare Fantom non utilizza una blockchain, ma un grafico aciclico diretto per lo storage delle transazioni. La PoS di queste piattaforme è più decentralizzata della PoA di BSC. Nella tabella 8.1 diamo a xDai un punteggio di decentralizzazione inferiore rispetto a Fantom poiché fa utilizzo di meno validatori (che possono comunque aumentare nel tempo).

Affinchè un sistema sia sicuro, resistente ai fault e non manipolabile, con le tecnologie attualmente disponibili, si ha inevitabilmente bisogno di sacrificare la scalabilità in favore delle suddette caratteristiche ed è quindi necessaria una soluzione decentralizzata.

Utilizzando come riferimento gli Smart Contract proposti nei capitoli 4 e A, è stata eseguita un'analisi dei costi di esecuzione delle funzioni implementate.

Questa analisi è mostrata nella Tabella 8.2.

Nell'analisi è stato considerato il prezzo di *Gwei* medio effettivo per ciascuna piattaforma ed è stato calcolato il costo di esecuzione in termini di coin nativa.

4 Nonostante sia stato scelto Fantom per questa proiezione, lo stesso discorso è equivalente per qualsiasi altra piattaforma EVM compatibile alternativa (Polygon, Avalanche, ecc.)

Table 8.2: Analizzando le alternative alla piattaforma Ethereum, utilizziamo un costo Gwei medio per ciascuna di esse in fase di utilizzo intensivo. I valori medi attuali sono descritti nella seconda colonna. Per ogni piattaforma abbiamo calcolato il costo di ogni operazione degli Smart Contract sviluppati. I costi sono riportati in termini di moneta nativa della piattaforma. La prospettiva dei costi in dollari del contratto è riportata nella tabella 8.3 e nella figura 8.1. Il codice del contratto è stato ottimizzato a 200 round al fine di diminuirne i costi e i passi da eseguire. Nella tabella è riportato il costo della funziona acceptJob con un'unica consegna in sospeso.

Type	Medium Gwei	Constructor	one Inst.	addDelivery	acceptJob	makeReview
Ethereum	100	0.111749 Eth	0.004602 Eth	0.02297 Eth	0.004562 Eth	0.003955 Eth
Binance Smart Chain	10	0.011175 BnB	0.00046 BnB	0.002297 BnB	0.000456 BnB	0.000396 BnB
Fantom	1	0.001117 Ftm	0.000046 Ftm	0.00023 Ftm	0.000046 Ftm	0.000040 Ftm
xDai	1	0.001117 xDai	0.000046 xDai	0.00023 xDai	0.000046 xDai	0.000040 xDai

Table 8.3: Prezzi medi in Dollari USA delle transazioni degli Smart Contract. I prezzi vengono calcolati tramite la capitalizzazione degli asset digitali. La capitalizzazione è analizzata dal valore di 1 miliardo di dollari (1B) ai 10000 miliardi di dollari (10T). Si noti che la capitalizzazione di Ethereum al momento della scrittura è di 100 miliardi di dollari, questo significa che gli Smart Contract sviluppati, distribuiti su Ethereum al momento costerebbero in media 25,61$. I valori di questa tabella sono riportati graficamente in Figura 8.1.

	1B	5B	10B	100B	500B	1T	10T
Ethereum	0.26	1.28	2.56	25.61	128.08	256.17	2561.73
Binance Smart Chain	0.019	0.10	0.19	1.91	9.57	19.14	191.41
Fantom	0.000116	0.00058	0.00116	0.0116	0.058	0.12	1.16
xDai	0.0002958	0.0002958	0.0002958	0.0002958	0.0002958	0.0002958	0.0002958

Già da un'analisi superficiale, è chiaro che la nostra scelta si riduce a sole tre piattaforme: BSC, Fantom e xDai, poiché il prezzo del gas di Ethereum non è al momento sostenibile e

Table 8.4: Confronto con lavori in letteratura che utilizzano gli Smart Contract per i pagamenti automatici. Il confronto verte su: costi medi in Dollari USA, scalabilità in termini di velocità e costi, antimanipolabilità e trasparenza.

Ref.	Avg. op. cost ($)	**Scalability**	**Anti-manipulability**	**Transparency**
This	0.0002958	High	High	High
[40]	150.50	Low	High	High
[56]	Not provided	Not provided	High	High
[11]	Not provided	Low	High	High
[78]	Not Provided	Not provided	High	High
[5]	0	High	No	No

abbiamo scartato la tecnologia cloud a causa della mancanza delle caratteristiche indicate in precedenza.

Sebbene il costo e la scalabilità di queste tre piattaforme sembrino sostenibili a prima vista, potrebbero non esserlo in futuro, sia per limitazioni in termini di Transazioni Per Secondo (TPS)[5], sia, come vedremo studiando la proiezione delle capitalizzazioni, per i costi causati dall'aumento dei prezzi delle monete digitali native delle piattaforme.

E' stata quindi effettuata una proiezione dei prezzi tenendo conto della capitalizzazione di mercato degli asset utilizzati per il pagamento dei costi di transazione. Questa proiezione è mostrata in Tabella 8.3 e graficamente in Figura 8.1.

Per prima cosa abbiamo calcolato il prezzo medio di costi di transazione dello Smart Contract sviluppato:

$$\overline{P} = Gwei \frac{\sum_{i=0}^{5} Fg_i}{5}.$$

Dove *Gwei* è il GASPRICE e Fg_i è il GASLIMIT della funzione i.

Dopodiché abbiamo calcolato il prezzo per transazione media in dollari sulla base della proiezione della capitalizzazione di mercato della moneta digitale come segue:

$$X_i = \frac{Cs}{M_i} \overline{P}.$$

5 Limitazione però superabile grazie alla maturazione futura della tecnologia

Dove X è il prezzo previsto, Cs è l'offerta, cioè l'ammontare di moneta circolante, M è la capitalizzazione di mercato presa in analisi con:

$$1B \leq M \leq 10T.$$

Da questa proiezione è emerso che su BSC i costi di uno Smart Contract utilizzato in produzione iniziano a essere insostenibili quando $M > 10B$.

Continuando l'analisi, in tabella 8.3 è possibile notare come l'utilizzo di Fantom (FTM) per il sistema proposto nel capitolo 4 risulta sostenibile fino a $M < 1T$, la differenza con BSC ed Ethereum è dovuta ai bassi costi del Gwei e all'ampia offerta di FTM. Quando $M > 1T$ i costi per transazione iniziano a essere moderati.

Una menzione a parte merita xDai. Questa piattaforma utilizza come metodo di pagamento per le transazioni la moneta digitale Dai, un token legato al valore del dollaro USA. Per questo motivo, il suo prezzo rimane costante all'aumentare della capitalizzazione di mercato ed è quindi perfetto per il nostro lavoro in quanto ogni transazione costerà in media 0.00030$ indipendentemente dalla capitalizzazione e crescita dell'ecosistema.

8.2 SICUREZZA

L'utilizzo della tecnologia IoT per automatizzare i processi di gestione attraverso i dati forniti dai sensori e per monitorare e controllare da remoto i dispositivi digitali sta prendendo sempre più piede, ma la sicurezza dei dati sensibili ottenuti da tali dispositivi, i quali risultano deboli contro eventuali attacchi, è a elevato rischio. In questo elaborato, come visto nel capitolo 4, proponiamo di utilizzare dispositivi IoT al fine di tracciare il ciclo di vita di un task e l'ambiente che circonda il lavoratore.

Gli elementi funzionali chiave dell'IoT includono rilevamento, calcolo, comunicazione e controllo. Queste funzionalità sono realizzate attraverso una combinazione di dispositivi embedded, tecnologie di comunicazione wireless, sensori e attuatori. Un'attenta configurazione e distribuzione di componenti hardware, software e di rete sono essenziali per raggiungere gli obiettivi dell'applicazione.

Il "connected things" layer è costituito da sensori, attuatori, dispositivi embedded, smartphone ed elettrodomestici intelligenti in grado di raccogliere dati dal loro ambiente operativo o di attivare apparecchi elettronici tra cui serrature digitali, lampade, porte da garage, ecc. Sviluppare applicazioni o servizi robusti utilizzando le "connected things", è essenziale al fine di consegnare i dati agli end-point dell'applicazione desiderati.

A seconda dei requisiti dell'applicazione, i dati fluiscono dai dispositivi IoT al livello perimetrale o al livello cloud (o come nel nostro caso in uno Smart Contract). In alcuni casi, il livello perimetrale aggrega i dati e li segnala al livello cloud per ulteriori elaborazioni, analisi e visualizzazione. Nel caso dell'utilizzo di tali dispositivi con uno Smart Contract è sufficiente che il dispositivo o il "connected things" layer contenga una chiave privata e che la utilizzi per firmare una transazione contenente i dati da inserire nel contratto.

Il livello di comunicazione consente ai dispositivi di trasmettere e ricevere dati da altri dispositivi nello stack dell'applicazione.

I sistemi che fanno utilizzo di dispositivi IoT possono essere estremamente complessi. Ad esempio, gli algoritmi di apprendimento automatico e intelligenza artificiale possono utilizzare i dati dei sensori ambientali per comprendere meglio il cambiamento climatico oppure per fondere i dati ambientali con i dati di un sensore di traffico per studiare i pattern e raccomandare percorsi che permettano di ridurre l' inquinamento.

Questo tipo di applicazione richiede dati provenienti da una vasta collezione di sensori posizionati in ampie zone geografiche. L'implementazione di sensori e l'infrastruttura di comunicazione necessaria a scala urbana, però, richiedono risorse finanziarie significative, introducendo al contempo un'elevata complessità di gestione e manutenzione, oltre a moltiplicare le vulnerabilità del sistema.

Nel campo dell'IoT, in genere, i dispositivi sono limitati in termini di capacità di elaborazione, archiviazione e rete e risultano più vulnerabile agli attacchi rispetto a dispositivi più complessi come smartphone, tablet o computer.

La sicurezza è altrettanto importante nel mercato delle monete digitali.

In particolare, negli ultimi anni, diversi sono gli attacchi effettuati ai danni di protocolli che fanno utilizzo di Smart Contract. Il risultato è l'impressionante cifra di 5.9 miliardi di dollari in controvalore di fondi saccheggiati.[6]

Questo è dovuto a causa della natura open source degli Smart Contract, della giovinezza della tecnologia e dell'elevato incentivo ad attacchi ai protocolli che contengono centinaia di milioni di dollari in controvalore.

Utilizzando a capacità limitata come i dispositivi IoT per accedere ai protocolli e firmare le transazioni, il rischio si moltiplica.

E' quindi importante difendere i protocolli sviluppati attraverso tecniche di protezione dei fondi e i dispositivi IoT.

In [45] e [14], gli autori descrivono alcuni problemi dei dispositivi IoT e le relative soluzioni. In particolare da queste survey, emerge che la Blockchain è la migliore tecnologia in grado di proteggere i problemi di sicurezza dei dispositivi IoT.

In questa sezione, viene presentato un algoritmo di information fusion che permette di generare chiavi biometriche basate sull'iride e di fonderle con chiavi RSA autenticate.

8.3 IOT E BLOCKCHAIN

Le reti IoT sono reti decentralizzate come la blockchain, per questo motivo le due tecnologie vengono spesso utilizzate in coppia.

Inoltre, la blockchain presta le sue caratteristiche e i vantaggi alle reti IoT, le quali sono generalmente vulnerabili, che diventano affidabili, tolleranti ai guasti, a elaborazione immutabile e consente allo stesso tempo di generare e scambiare valore attraverso la raccolta di informazioni.

L'esempio più famoso, basato su una evoluzione della blockchain orientata ai sensori, cioè su tangles è IOTA [59].

L' esempio di applicazione della blockchain all'IoT più diffuso in letteratura è il tracciamento della supply chain. Con una blockchain, ogni step di un processo produttivo risulta certificato e questo aumenta il valore del prodotto e del brand.

6 Fonte aggiornata in tempo reale: https://defillama.com/hacks

Questa è una delle applicazioni che maggiormente esplicano le potenzialità della tecnologia blockchain applicata alle reti IoT, ma le applicazioni sono potenzialmente infinite.

In quanto abbiamo già mostrato nel capitolo 4 come un dispositivo IoT potrebbe aiutare a tracciare un task lavorativo[7], in questa sezione il nostro obiettivo è proteggere le transazioni dei dispositivi IoT ed evitare la potenziale doppia spesa o comportamenti malevoli degli operatori.

L'idea è di utilizzare una chiave biometrica estratta dall'iride e di fonderla con algoritmi di firma basati su numeri primi [37, 38].

La chiave biometrica piò essere utilizzata al fine di controllare l'identità degli utilizzatori del sistema, per poi essere fusa con chiavi private blockchain e utilizzata per firmare le transazioni. La chiave biometrica viene resa pubblica solo se l'attacco è comprovato, disincentivando questi ultimi, mantenendo quindi sicurezza e privacy.

8.4 CHIAVI POST QUANTUM BASATE SU BIOMETRIA

In [38], gli autori costruiscono una chiave che impiega una componente biometrica al fine di conservare l'identità dell'utente, rendendo una eventuale blockchain basata su queste chiavi, autenticata.

Applicando tale paradigma al sistema proposto in questo elaborato, otteniamo dei contratti di lavoro autenticati; ciò che viene depositato sugli Smart Contract descritti nel capitolo 4 e tutte le operazioni effettuate su esso, sono a loro volta autenticate, permettendo di intervenire in caso di azioni malevole.

Queste chiavi sono ottenute fondendo una chiave generata da un algoritmo di crittografia asimmetrica (RSA) e una chiave biometrica. In particolare, nel lavoro [37], viene acquisita l'iride e viene descritto l'algoritmo della generazione della chiave, il quale è formato dai seguenti passaggi (Fig. 8.2):

- acquisizione dell'iride;

7 Favorendo allo stesso tempo la gestione della supply chain

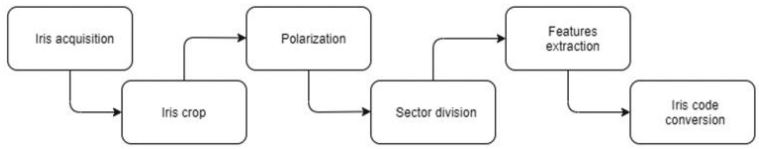

Figure 8.2: Visual abstract delll'estrazione delle features dell'iride

- pre-elaborazione dell'immagine dell'iride: in questo passaggio essa viene filtrata e segmentata al fine di estrarre le features nel successivo passo;
- estrazione delle features dell'iride;
- conversione delle features in un vettore numerico attraverso l'algoritmo IIF[37];
- calcolo della chiave RSA;
- produzione della matrice quadrata composta da vettore numerico e chiave privata RSA.

Il risultato del calcolo è una chiave con un valore NIST che è stato calcolato per 100 iridi. Di questi, tutti i P-value sono maggiori di 0,01.

Un valore P maggiore di 0,01 certifica che una sequenza di bit è casuale. In [38], è stata sviluppata una PoC che comprende due componenti:

- il client, cioè un'applicazione mobile che esegue l'acquisizione dell'iride, la generazione della chiave RSA, l'information fusion ed esegue le firme sullo Smart Contract/blockchain;
- il backend, ovvero un backend decentralizzato Smart Contract/blockchain che permette di gestire depositi e firme.

8.5 UNO SMART CONTRACT PER PREVENIRE IL DOUBLE SPENDING E GESTIRE I DISPOSITIVI IOT

In questa PoC, viene utilizzato il metodo descritto in [37] al fine di effettuare una transazione Smart Contract sicura, autenticando i dispositivi IoT.

Quando viene effettuata una transazione, l'applicazione client, in [38], si occuperà di generare uno Smart Contract tra le parti, permettendo al mittente di firmare la transazione verso lo Smart Contract con la sua chiave IIF.

La PoC in [38] comprende uno Smart Contract escrow[8] che sviluppa i seguenti passaggi:

- il client crea una chiave RSA;
- il client esegue l'acquisizione dell'iride, calcolando successivamente il codice IIF; memorizza il prodotto di due numeri primi, la chiave RSA e successivamente istanzia il contratto;
- il client crea lo Smart Contract ed entrambe le parti firmeranno il contratto inserendo l'hash del proprio codice IIF;
- lo Smart Contract entra in fase di esecuzione;
- l'acquirente depositerà le monete digitali nel contratto;
- entrambe le parti accetteranno la transazione;
- quando le parti accetteranno la transazione, i fondi verranno trasferiti dal contratto all'indirizzo del venditore.

La parte biometrica è memorizzata su una blockchain privata ed è accessibile solo se la doppia spesa o l'azione illegale viene provata ed esiste consenso riguardo il rilascio della chiave. Questo tipo di approccio, chiaramente, è in visione futura ed occorrebbe una regolamentazione chiara di questa tecnologia. Al fine di far rispettare questo "regolamento", una Smart City o un'entità legale di terze parti potrebbe essere coinvolta.

Quando la transazione avviene, una futura doppia spesa o qualsiasi azione dannosa diventa proibita e legalmente perseguibile, poiché nel sistema, grazie ai dati biometrici, viene tracciata l'identità dell'utilizzatore di un indirizzo. In assenza di regolamentazione, come detto in precedenza, è possibile lasciare la decisione al network.

8 contratto per lo scambio di beni, la tecnica può essere utilizzata per qualsiasi tipo di transazione o di accesso a uno Smart Contract (Nel caso di questo elaborato la tecnica viene applicata per proteggere le aziende che depositano fondi nel contratto descritto nel capitolo 4)

In quel caso, qualora un'azione malevola venga provata, al fine di recuperare la chiave biometrica per ottenere l'identità dell'attaccante, ci sarà una fase di voto in cui il network voterà per svelarla.

La chiave IIF viene salvata nella blockchain tramite hash, attraverso uno Smart Contract che rende quindi le transazioni effettuate al suo interno, autenticate.

Lo Smart Contract in [38] viene creato dall'applicazione quando una transazione viene eseguita ed è sviluppato come segue:

- quando viene avviata una transazione, l'hash dell'IIF dell'acquirente viene memorizzato nel contratto attraverso il costruttore;
- il contratto ha una funzione deposit() che permette all'acquirente, dopo essere stato identificato, di depositare i fondi nel contratto. Questa funzione è accessibile solo all'acquirente;
- successivamente, il venditore registra il suo IIF e il proprio indirizzo sul contratto attraverso la funzione register();
- a questo punto, l'acquirente deve accettare la transazione tramite funzione accept() che sblocca e trasferisce i fondi al venditore;
- le parti, nel caso di problemi con lo scambio, possono anche distruggere il contratto attraverso la funzione cancel() la quale restituisce i fondi all'acquirente.

In questo modo, le parti firmano lo Smart Contract e i fondi vengono congelati al suo interno fino a quando esse non accetteranno la transazione. Di conseguenza, il venditore è protetto dallo Smart Contract e dall'autenticazione dell'acquirente che può essere legalmente perseguito in caso di azioni malevole.

In Fig. 8.3 mostriamo come il client è sviluppato. Quando il mittente deposita i fondi, il contratto verrà automaticamente distribuito e firmato con la chiave IIF del mittente. Successivamente, lo stesso deposita il fondi sul contratto attraverso la funzione deposit(). D'altra parte, il deposito del mittente viene notificato al destinatario, che viene identificato attraverso l'iride. Quando il destinatario accetta i fondi dopo essersi registrato sullo Smart

Contract attraverso la funzione register(), il client emette una notifica al mittente. Infine, se quest'ultimo decide di ultimare il trasferimento, si effettua una seconda identificazione tramite iride ed i fondi vengono sbloccati e trasferiti al destinatario.

8.6 BLOCKCHAIN POST-QUANTUM

Una questione aperta nella ricerca nell'ambito di questa tecnologia è quello di rendere le chiavi utilizzate per la firma di transazioni blockchain sicura contro gli attacchi post quantum. [23] è una survey sullo stato dell'arte delle blockchain post quantum. Come descritto in [23], le caratteristiche di una chiave blockchain post quantum sono:

- lunghezza della chiave limitata;
- lunghezza degli hash e delle firme limitata;
- alta velocità di esecuzione;
- bassa complessità computazionale;
- basso dispendio di energia.

Una soluzione potrebbe essere quella di utilizzare le chiavi IIF come base randomica delle firme basate su hash, come ad esempio gli schemi di Lamport[81]. In questo caso gli schemi di Lamport risultano "pesanti" per le chiavi blockchain e uno studio per l'ottimizzazione di tali schemi combinati con le chiavi IIF è in corso.

8.7 SICUREZZA DELLE CHIAVI

Il risultato che determina la casualità di una sequenza è il P-value, che è un valore nell'intervallo [0,1]. Utilizzando questo valore come metro di giudizio, una sequenza è considerata casuale quando il P-value è maggiore di 0,01. In questo test, Iovane et al. in [37] hanno utilizzato 1024 bit come dimensione della chiave privata RSA, mentre la sequenza biometrica estratta dall'iride è formata da 2048 bit. La Figura 8.4 mostra i risultati degli esperimenti sul P-value di 100 sequenze estratte da altrettante iridi.

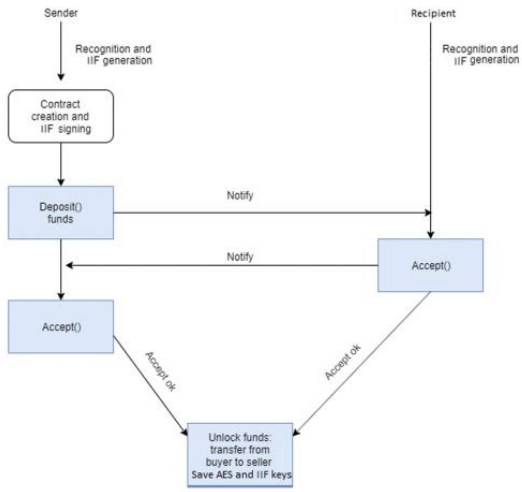

Figure 8.3: Schema del ciclo di transazione. Il client crea lo Smart Contract e richiama le sue funzioni; i diagrammi blu rappresentano le funzioni Smart Contract. 1)Il sender crea lo Smart Contract e lo firma con IIF. 2) Il mittente deposita fondi. 3) Il destinatario accetta la transazione. 4) Il mittente accetta la transazione. 5) Il i fondi vengono sbloccati e la parte biometrica dell' IIF del mittente viene salvata al fine di prevenire la doppia spesa.

Test NIST	*P-value*
Frequency	0,574873
Block Frequency	0,794211
Cumulative Sums1	0,483279
Cumulative Sums2	0,412537
Runs	0,293751
Longest Run	0,426564

Figure 8.4: NIST Test per P-value

Tutti i P-value sono maggiori di 0,01, ne consegue che con una chiave a 2048 bit può essere considerata una sequenza casuale. La casualità di questo risultato dimostra che questo tipo di chiavi si presta particolarmente bene in uno schema di firma basato su hash e utilizzabile in una blockchain ex-novo autenticata.

8.8 PRIVACY

Grazie al P-value, possiamo dire che la sequenza IIF è praticamente un numero casuale. Secondo lo schema definito in [38], la parte biometrica viene salvata in blockchain e può essere estratto solo se c'è consenso sul fatto che un evento sia stato malevolo. In questo modo, la privacy degli utenti è garantita fino a quando essi non assumeranno comportamenti dannosi per la rete.

8.9 SICUREZZA BASATA SUL FRONTRUNNING

Come ultimo spunto, descriviamo un altro metodo per la mitigazione di transazioni dannose. Oltre alla protezione basata sulle chiavi IIF, in questo elaborato proponiamo un nuovo tipo di difesa agli Smart Contract utilizzati in un sistema simile a quello mostrato nel capitolo 4. Questa protezione fa utilizzo del Miner Extractable Value (MEV) definito per la priva volta in [17]. Il MEV è una tecnica che sfrutta una delle caratteristiche delle blockchain più apprezzate: la trasparenza.

Avendo a disposizione un nodo blockchain, è possibile intercettare le transazioni che sono in fase di validazione. Grazie a

questa possibilità, diversi programmatori sfruttano le transazioni di utenti ignari al fine di trarne profitto [60].

In [71], gli autori propongono delle difese contro questi attacchi; classificando le transazioni attraverso il machine learning, è possibile definire se si tratta di un attacco e, nel caso, reagire attivando le funzioni di emergenza degli Smart Contract manipolando il prezzo del GAS in maniera tale da posizionare la transazione di difesa prima di quella dell'attacco.

Part IV

CONCLUSIONI E UNO SGUARDO AL FUTURO

9 CONCLUSIONI E FUTURO REGOLAMENTATIVO

In questo elaborato abbiamo mostrato una PoC che permette di tracciare un processo lavorativo, emettere pagamenti senza intermediari e produrre royalties.

Questo sistema implica l'apertura di diverse strade di ricerca nel campo della tecno-regolamentazione, in quanto sarebbe uno dei primi utilizzi di Smart Contract per la gestione dei pagamenti aziendali. Come analizzato in [26], gli Stati Uniti d'America sono il primo stato ad essersi mosso nal campo della regolamentazione di queste tecnologie, tuttavia, pur avendo le potenzialità regolative dei classici contratti, siamo ancora lontani dall'avere delle leggi chiare e precise sull'utilizzo degli Smart Contract per la regolamentazione dei pagamenti ed occorre approfondire le ricerche in tale ambito.

Per questo motivo, al momento, il giusto utilizzo di questa tecnologia (in combinazione con l'IoT, è quello della distribuzione di royalties al fine di tracciare e gestire il lavoro umano. Nell'elaborato è stato trattato anche il problema della scalabilità e della sicurezza di una soluzione di questo tipo utilizzata in un contesto reale.

Successivamente è stata presentata la teoria della plausibilità e un modello di supporto alle decisioni in contesti di info-incertezza ed info-incompletezza. La validazione di tale teoria è stata effettuata con un esperimento in ambito sportivo dove è stata utilizzata una ANN al fine di calcolare il valore del prezzo di negoziazione degli atleti fra società sportive. L'esperimento dimostra come in assenza della fonte associata alla Probabilità, utilizzando i modelli descritti in [35] e fondendo le informazioni provenienti dalla Plausibilità, Credibilità e Possibilità, siamo in grado di migliorare le performance di decisione rispetto alle singole fonti.

Infine è stato presentato il TES, sistema di supporto alle decisioni basato sulla teoria dell'infoincertezza che permette di calcolare il *fair value* di un token. In particolare è stato mostrato il

calcolo del *fair value* di bitcoin in stato di euforia e depressione e il valore di un token royalty basato sul lavoro umano e l'estrazione di materie prime.

Part V

APPENDIX

APPENDIX TEST

A.1 SMART CONTRACT CODE

A.1.1 *Context.sol*

```
pragma solidity ^0.6.0;

abstract contract Context {
  function _msgSender() internal view virtual returns (address
      payable) {
    return msg.sender;
  }

  function _msgData() internal view virtual returns (bytes memory
      ) {
    this; // silence state mutability warning without generating
        bytecode - see https://github.com/ethereum/solidity/
        issues/2691
    return msg.data;
  }
}
```

A.1.2 *Ownable.sol*

```
pragma solidity ^0.6.0;

import "Context.sol";

contract Ownable is Context {
  address private _owner;

  event OwnershipTransferred(address indexed previousOwner,
      address indexed newOwner);

  constructor () internal {
    address msgSender = _msgSender();
```

```
    _owner = msgSender;
    emit OwnershipTransferred(address(0), msgSender);
  }

  function owner() public view returns (address) {
    return _owner;
  }

  modifier onlyOwner() {
    require(_owner == _msgSender(), "Ownable: caller is not the
        owner");
    _;
  }

  function renounceOwnership() public virtual onlyOwner {
    emit OwnershipTransferred(_owner, address(0));
    _owner = address(0);
  }

  function transferOwnership(address newOwner) public virtual
      onlyOwner {
    require(newOwner != address(0), "Ownable: new owner is the
        zero address");
    emit OwnershipTransferred(_owner, newOwner);
    _owner = newOwner;
  }
}
```

A.1.3 *safeMath.sol*

```
pragma solidity ^0.6.0;

/**
* @dev Wrappers over Solidity's arithmetic operations with added
    overflow
* checks.
*
* Arithmetic operations in Solidity wrap on overflow. This can
    easily result
* in bugs, because programmers usually assume that an overflow
    raises an
* error, which is the standard behavior in high level programming
     languages.
* `SafeMath` restores this intuition by reverting the transaction
     when an
* operation overflows.
```

```
 *
 * Using this library instead of the unchecked operations
     eliminates an entire
 * class of bugs, so it's recommended to use it always.
 */
library SafeMath {
  /**
   * @dev Returns the addition of two unsigned integers, reverting
       on
   * overflow.
   *
   * Counterpart to Solidity's `+` operator.
   *
   * Requirements:
   * - Addition cannot overflow.
   */
  function add(uint256 a, uint256 b) internal pure returns (
      uint256) {
    uint256 c = a + b;
    require(c >= a, "SafeMath: addition overflow");

    return c;
  }

  /**
   * @dev Returns the subtraction of two unsigned integers,
       reverting on
   * overflow (when the result is negative).
   *
   * Counterpart to Solidity's `-` operator.
   *
   * Requirements:
   * - Subtraction cannot overflow.
   */
  function sub(uint256 a, uint256 b) internal pure returns (
      uint256) {
    return sub(a, b, "SafeMath: subtraction overflow");
  }

  /**
   * @dev Returns the subtraction of two unsigned integers,
       reverting with custom message on
   * overflow (when the result is negative).
   *
   * Counterpart to Solidity's `-` operator.
   *
   * Requirements:
   * - Subtraction cannot overflow.
```

```
    */
    function sub(uint256 a, uint256 b, string memory errorMessage)
        internal pure returns (uint256) {
      require(b <= a, errorMessage);
      uint256 c = a - b;

      return c;
    }

    /**
    * @dev Returns the multiplication of two unsigned integers,
        reverting on
    * overflow.
    *
    * Counterpart to Solidity's `*` operator.
    *
    * Requirements:
    * - Multiplication cannot overflow.
    */
    function mul(uint256 a, uint256 b) internal pure returns (
        uint256) {
      // Gas optimization: this is cheaper than requiring 'a' not
          being zero, but the
      // benefit is lost if 'b' is also tested.
      // See: https://github.com/OpenZeppelin/openzeppelin-
          contracts/pull/522
      if (a == 0) {
        return 0;
      }

      uint256 c = a * b;
      require(c / a == b, "SafeMath: multiplication overflow");

      return c;
    }

    /**
    * @dev Returns the integer division of two unsigned integers.
        Reverts on
    * division by zero. The result is rounded towards zero.
    *
    * Counterpart to Solidity's `/` operator. Note: this function
        uses a
    * `revert` opcode (which leaves remaining gas untouched) while
        Solidity
    * uses an invalid opcode to revert (consuming all remaining gas
        ).
    *
```

```
93      * Requirements:
        * - The divisor cannot be zero.
        */
        function div(uint256 a, uint256 b) internal pure returns (
            uint256) {
          return div(a, b, "SafeMath: division by zero");
98      }

        /**
        * @dev Returns the integer division of two unsigned integers.
            Reverts with custom message on
        * division by zero. The result is rounded towards zero.
103     *
        * Counterpart to Solidity's `/` operator. Note: this function
            uses a
        * `revert` opcode (which leaves remaining gas untouched) while
            Solidity
        * uses an invalid opcode to revert (consuming all remaining gas
            ).
        *
108     * Requirements:
        * - The divisor cannot be zero.
        */
        function div(uint256 a, uint256 b, string memory errorMessage)
            internal pure returns (uint256) {
          // Solidity only automatically asserts when dividing by 0
113       require(b > 0, errorMessage);
          uint256 c = a / b;
          // assert(a == b * c + a % b); // There is no case in which
              this doesn't hold

          return c;
118     }

        /**
        * @dev Returns the remainder of dividing two unsigned integers.
             (unsigned integer modulo),
        * Reverts when dividing by zero.
123     *
        * Counterpart to Solidity's `%` operator. This function uses a
            `revert`
        * opcode (which leaves remaining gas untouched) while Solidity
            uses an
        * invalid opcode to revert (consuming all remaining gas).
        *
128     * Requirements:
        * - The divisor cannot be zero.
        */
```

```
    function mod(uint256 a, uint256 b) internal pure returns (
        uint256) {
      return mod(a, b, "SafeMath: modulo by zero");
    }

    /**
    * @dev Returns the remainder of dividing two unsigned integers.
         (unsigned integer modulo),
    * Reverts with custom message when dividing by zero.
    *
    * Counterpart to Solidity's `%` operator. This function uses a
        `revert`
    * opcode (which leaves remaining gas untouched) while Solidity
        uses an
    * invalid opcode to revert (consuming all remaining gas).
    *
    * Requirements:
    * - The divisor cannot be zero.
    */
    function mod(uint256 a, uint256 b, string memory errorMessage)
        internal pure returns (uint256) {
      require(b != 0, errorMessage);
      return a % b;
    }
}
```

A.1.4 *IERC20.sol*

```
pragma solidity ^0.8.0;

/**
* @dev Interface of the ERC20 standard as defined in the EIP.
*/
interface IERC20 {
  /**
  * @dev Returns the amount of tokens in existence.
  */
  function totalSupply() external view returns (uint256);

  /**
  * @dev Returns the amount of tokens owned by `account`.
  */
  function balanceOf(address account) external view returns (
      uint256);
```

```
    /**
     * @dev Moves `amount` tokens from the caller's account to `
        recipient`.
     *
     * Returns a boolean value indicating whether the operation
        succeeded.
     *
     * Emits a {Transfer} event.
     */
    function transfer(address recipient, uint256 amount) external
        returns (bool);

    /**
     * @dev Returns the remaining number of tokens that `spender`
        will be
     * allowed to spend on behalf of `owner` through {transferFrom}.
         This is
     * zero by default.
     *
     * This value changes when {approve} or {transferFrom} are
        called.
     */
    function allowance(address owner, address spender) external
        view returns (uint256);

    /**
     * @dev Sets `amount` as the allowance of `spender` over the
        caller's tokens.
     *
     * Returns a boolean value indicating whether the operation
        succeeded.
     *
     * IMPORTANT: Beware that changing an allowance with this method
         brings the risk
     * that someone may use both the old and the new allowance by
        unfortunate
     * transaction ordering. One possible solution to mitigate this
        race
     * condition is to first reduce the spender's allowance to 0 and
         set the
     * desired value afterwards:
     * https://github.com/ethereum/EIPs/issues/20#issuecomment
        -263524729
     *
     * Emits an {Approval} event.
     */
    function approve(address spender, uint256 amount) external
        returns (bool);
```

```
    /**
    * @dev Moves `amount` tokens from `sender` to `recipient` using the
    * allowance mechanism. `amount` is then deducted from the caller's
    * allowance.
    *
    * Returns a boolean value indicating whether the operation succeeded.
    *
    * Emits a {Transfer} event.
    */
    function transferFrom(
    address sender,
    address recipient,
    uint256 amount
    ) external returns (bool);

    function mint(address to, uint256 amount) external;

    function burn(uint256 amount) external;

    /**
    * @dev Emitted when `value` tokens are moved from one account (`from`) to
    * another (`to`).
    *
    * Note that `value` may be zero.
    */
    event Transfer(address indexed from, address indexed to, uint256 value);

    /**
    * @dev Emitted when the allowance of a `spender` for an `owner` is set by
    * a call to {approve}. `value` is the new allowance.
    */
    event Approval(address indexed owner, address indexed spender, uint256 value);
}
```

A.1.5 *ERC20.sol*

```
/**
*Submitted for verification at Etherscan.io on 2018-11-12
```

```
*/


pragma solidity ^0.6.0;

import "SafeMath.sol";

contract owned {
  address public owner;

  function owned() public {
    owner = msg.sender;
  }

  modifier onlyOwner {
    require(msg.sender == owner);
    _;
  }

  function transferOwnership(address newOwner) onlyOwner public {
    owner = newOwner;
  }
}

interface tokenRecipient { function receiveApproval(address _from
    , uint256 _value, address _token, bytes _extraData) external;
     }

contract TokenERC20 is owned, SafeMath {
  // Public variables of the token
  string public name;
  string public symbol;
  uint8 public decimals = 18;
  // 18 decimals is the strongly suggested default, avoid
      changing it
  uint256 public totalSupply;

  // This creates an array with all balances
  mapping (address => uint256) public balanceOf;
  mapping (address => mapping (address => uint256)) public
      allowance;

  mapping (address => bool) public frozenAccount;

  /* This generates a public event on the blockchain that will
      notify clients */
  event FrozenFunds(address target, bool frozen);
```

```
46
    // This generates a public event on the blockchain that will
        notify clients
    event Transfer(address indexed from, address indexed to,
        uint256 value);

    // This generates a public event on the blockchain that will
        notify clients
51  event Approval(address indexed _owner, address indexed _spender
        , uint256 _value);

    // This notifies clients about the amount burnt
    event Burn(address indexed from, uint256 value);

56  /**
    * Constrctor function
    *
    * Initializes contract with initial supply tokens to the
        creator of the contract
    */
61  function TokenERC20(
    uint256 initialSupply,
    string tokenName,
    string tokenSymbol
    ) public {
66    totalSupply = initialSupply * 10 ** uint256(decimals);  //
          Update total supply with the decimal amount
      balanceOf[msg.sender] = totalSupply;                // Give
          the creator all initial tokens
      name = tokenName;                                   // Set
          the name for display purposes
      symbol = tokenSymbol;                               // Set
          the symbol for display purposes
    }
71

    /**
    * Transfer tokens
    *
    * Send `_value` tokens to `_to` from your account
76  *
    * @param _to The address of the recipient
    * @param _value the amount to send
    */
    function transfer(address _to, uint256 _value) public returns (
        bool success) {
81    // Prevent transfer to 0x0 address. Use burn() instead
      require(_to != 0x0);
      require(_value > 0);
```

```
    // Check if the sender has enough
    require(balanceOf[msg.sender] >= _value);
    // Check for overflows
    require(balanceOf[_to] + _value > balanceOf[_to]);
    // Check if sender is frozen
    require(!frozenAccount[msg.sender]);
    // Check if recipient is frozen
    require(!frozenAccount[_to]);
    // Subtract from the sender
    balanceOf[msg.sender] = SafeMath.safeSub(balanceOf[msg.sender
        ], _value);
    // Add the same to the recipient
    balanceOf[_to] = SafeMath.safeAdd(balanceOf[_to], _value);
    emit Transfer(msg.sender, _to, _value);
    return true;
  }

  /**
  * Transfer tokens from other address
  *
  * Send `_value` tokens to `_to` in behalf of `_from`
  *
  * @param _from The address of the sender
  * @param _to The address of the recipient
  * @param _value the amount to send
  */
  function transferFrom(address _from, address _to, uint256
      _value) public returns (bool success) {
    // Prevent transfer to 0x0 address. Use burn() instead
    require(_to != 0x0);
    require(_value > 0);
    // Check if the sender has enough
    require(balanceOf[_from] >= _value);
    // Check for overflows
    require(balanceOf[_to] + _value > balanceOf[_to]);
    // Check if sender is frozen
    require(!frozenAccount[_from]);
    // Check if recipient is frozen
    require(!frozenAccount[_to]);
    // Check allowance
    require(_value <= allowance[_from][msg.sender]);
    // Subtract from the sender
    balanceOf[_from] = SafeMath.safeSub(balanceOf[_from], _value)
        ;
    // Add the same to the recipient
    balanceOf[_to] = SafeMath.safeAdd(balanceOf[_to], _value);
```

```
    allowance[_from][msg.sender] = SafeMath.safeSub(allowance[
        _from][msg.sender],_value);
    emit Transfer(_from, _to, _value);
    return true;
  }

  /**
  * Set allowance for other address
  *
  * Allows `_spender` to spend no more than `_value` tokens in
      your behalf
  *
  * @param _spender The address authorized to spend
  * @param _value the max amount they can spend
  */
  function approve(address _spender, uint256 _value) public
  returns (bool success) {
    require(_value > 0);
    allowance[msg.sender][_spender] = _value;
    emit Approval(msg.sender, _spender, _value);
    return true;
  }

  /**
  * Set allowance for other address and notify
  *
  * Allows `_spender` to spend no more than `_value` tokens in
      your behalf, and then ping the contract about it
  *
  * @param _spender The address authorized to spend
  * @param _value the max amount they can spend
  * @param _extraData some extra information to send to the
      approved contract
  */
  function approveAndCall(address _spender, uint256 _value, bytes
       _extraData)
  public
  returns (bool success) {
    tokenRecipient spender = tokenRecipient(_spender);
    if (approve(_spender, _value)) {
      spender.receiveApproval(msg.sender, _value, this,
          _extraData);
      return true;
    }
  }

  /**
  * Destroy tokens
```

```
    *
    * Remove '_value' tokens from the system irreversibly
    *
    * @param _value the amount of money to burn
    */
    function burn(uint256 _value) public returns (bool success) {
      // Check if the sender has enough
      require(balanceOf[msg.sender] >= _value);
      require(_value > 0);
      // Subtract from the sender
      balanceOf[msg.sender] = SafeMath.safeSub(balanceOf[msg.sender
          ], _value);
      // Updates totalSupply
      totalSupply = SafeMath.safeSub(totalSupply, _value);
      emit Burn(msg.sender, _value);
      return true;
    }

    /**
    * Destroy tokens from other account
    *
    * Remove '_value' tokens from the system irreversibly on behalf
        of '_from'.
    *
    * @param _from the address of the sender
    * @param _value the amount of money to burn
    */
    function burnFrom(address _from, uint256 _value) public returns
         (bool success) {
      require(balanceOf[_from] >= _value);                // Check
          if the targeted balance is enough
      require(_value > 0);
      require(_value <= allowance[_from][msg.sender]);    // Check
          allowance
      // Subtract from the targeted balance
      balanceOf[_from] = SafeMath.safeSub(balanceOf[_from], _value)
          ;
      // Subtract from the sender's allowance
      allowance[_from][msg.sender] = SafeMath.safeSub(allowance[
          _from][msg.sender], _value);
      // Update totalSupply
      totalSupply = SafeMath.safeSub(totalSupply, _value);
      emit Burn(_from, _value);
      return true;
    }

    /// @notice Create 'mintedAmount' tokens and send it to 'target
        '
```

```
    /// @param target Address to receive the tokens
    /// @param mintedAmount the amount of tokens it will receive
    function mintToken(address target, uint256 mintedAmount)
        onlyOwner public {
      balanceOf[target] = SafeMath.safeAdd(balanceOf[target],
          mintedAmount);
      totalSupply = SafeMath.safeAdd(totalSupply, mintedAmount);
      emit Transfer(0, this, mintedAmount);
      emit Transfer(this, target, mintedAmount);
    }

    /// @notice ‘freeze? Prevent | Allow‘ ‘target‘ from sending &
        receiving tokens
    /// @param target Address to be frozen
    /// @param freeze either to freeze it or not
    function freezeAccount(address target, bool freeze) onlyOwner
        public {
      frozenAccount[target] = freeze;
      emit FrozenFunds(target, freeze);
    }
}
```

A.1.6 *employmentContract.sol*

Il codice che comprende l'utilizzo dell'oracolo è commentato al fine di ridurre la complessità dell'implementazione.

```
pragma solidity >=0.6.0 <0.8.0;

import './safemath.sol';
import './IERC20.sol';
//import "https://raw.githubusercontent.com/smartcontractkit/
    chainlink/develop/evm-contracts/src/v0.6/ChainlinkClient.sol
    ";

contract employmentContract is /*ChainlinkClient,*/Ownable{
  using SafeMath for uint256;

  address payable [] employees;
  mapping (address => bool) hasWorked;
  IERC20 royalty;
  //uint256 blockEnd;
  //uint256 numberOfDeliveries;
  uint256 minimumPaymentPerHour=11; //set to 11 Eur, this is the
      payment and not the royalty
  struct delivery{
```

```
    address employee;
    bool isUnfavorableWeather;
    uint256 startBlock;
    uint256 endBlock;
    uint256 latitude;
    uint256 longitude;
    uint256 start;
    uint256 end;
    uint256 fatigue;
    bool accident;
    bool isNight;
    bool isHoliday;
  }
  delivery [] deliveries;

  //address private oracle;
  //bytes32 private jobId;
  //uint256 private fee;
  //bytes32 private apiKey;
  mapping (address => bool) registeredIoT;
  mapping (address => uint256) royaltyAmount;

  event depositMade(uint256 amount);
  event reviewMade(uint256 numberOfDeliveries, string review);
  event paymentMade(uint256 amount,address employee);

  constructor (address [] memory _registeredIoT, IERC20 _royalty)
       public{
    royalty=_royalty;

    //blockEnd=_blockEnd;
    //setPublicChainlinkToken();
    //numberOfDeliveries=_numberOfDeliveries; //no fixed umber of
        deliveries
    //oracle = 0x2f90A6D021db21e1B2A077c5a37B3C7E75D15b7e;
    //jobId = "29fa9aa13bf1468788b7cc4a500a45b8";
    //fee = 0.1 * 10 ** 18; // 0.1 LINK
    //apiKey= "63cdc582c5324d258f0153931211103";
    for(uint256 i=0;i<_registeredIoT.length;i++){
      registeredIoT[_registeredIoT[i]]=true;
    }
  }

  function employeeRegistration() external{
    employees.push(msg.sender);
  }
```

```
    //params are collected from IoT devices -> smartband with nfc
        -> smartphone app
    function addDelivery(uint256 latitude, uint256 longitude,
        uint256 fatigue, bool accident, bool isNight, uint256 start
        , uint256 end, bool isUnfavorableWeather, bool isHoliday,
        address worker) external{
      //Chainlink.Request memory request = buildChainlinkRequest(
          jobId, address(this), this.fulfill.selector);
      require(registeredIoT[msg.sender],"IoT device is not
          registered");
      //only registered IoT devices can sign the transaction. A
          checksum on the device integrity is reccomended
      // Set the URL to perform the GET request on
      //request.add("get", "https://*request parameters*");
      //return sendChainlinkRequestTo(oracle, request, fee);
      delivery memory del;
      del.employee=worker;
      del.isUnfavorableWeather=isUnfavorableWeather;
      del.isNight=isNight;
      del.end=end;
      del.start=start;
      del.latitude=latitude;
      del.longitude=longitude;
      del.fatigue=fatigue;
      del.accident=accident;
      del.isHoliday=isHoliday;
      deliveries.push(del);
      hasWorked[worker]=true;
    }

    //accept from the employer, but the employee could be payed
        after each delivery
    function acceptJob (address payable employee) external
        onlyOwner {
      uint256 calculatePay=0;
      for(uint256 i=0;i<deliveries.length;i++){
        if(deliveries[i].employee==employee){
          uint256 workHours=(deliveries[i].end-deliveries[i].start)
              .div(uint256(60).mul(60));
          if(deliveries[i].isNight && deliveries[i].isHoliday &&
              deliveries[i].isUnfavorableWeather)
          calculatePay=calculatePay+(minimumPaymentPerHour+
              minimumPaymentPerHour.mul(20).div(100)).mul(workHours
              );
          /*if(deliveries.accident)
          send compensation*/
          else if((deliveries[i].isNight && deliveries[i].isHoliday
              ) || (deliveries[i].isNight && deliveries[i].
```

```
            isUnfavorableWeather) || (deliveries[i].isHoliday &&
            deliveries[i].isUnfavorableWeather))
        calculatePay=calculatePay+(minimumPaymentPerHour+
            minimumPaymentPerHour.mul(15).div(100)).mul(workHours
            );
        else if(deliveries[i].isNight || deliveries[i].isHoliday
            || deliveries[i].isUnfavorableWeather)
        calculatePay=calculatePay+(minimumPaymentPerHour+
            minimumPaymentPerHour.mul(10).div(100)).mul(workHours
            );
      }
    }
    //the fatigue could be used to calculate the royalties

    royaltyAmount[employee]=royaltyAmount[employee]+calculatePay;
         //it could be also based on the price, it mesures the
        work of the employee
    employee.transfer(calculatePay);
    emit paymentMade(calculatePay,employee);
  }

  function deposit() external payable onlyOwner {
    emit depositMade(msg.value);
  }


  function addIoT(address IoT) external onlyOwner{
    registeredIoT[IoT]=true;
  }

  function removeIoT(address IoT) external onlyOwner{
    registeredIoT[IoT]=false;
  }

  function makeReview(string memory reviewReference) external {
    require(hasWorked[msg.sender],"you are not an employee");
    royalty.mint(msg.sender, royaltyAmount[msg.sender]); // mint
        and send the royalties in ERC20 to the reviewer,
        chainlink to retrieve the price, minted at review for
        incentivize it
    emit reviewMade(deliveries.length,reviewReference);
  }

}
```

BIBLIOGRAPHY

[1] S N. Ali, N. Haghpanah, X. Lin, and R. Siegel. «How to Sell Hard Information.» In: *The Quarterly Journal of Economics* (2021). DOI: 10.1093/qje/qjab024.

[2] S. Ammous. *The Bitcoin Standard: The Decentralized Alternative to Central Banking*. Wiley, 2018. ISBN: 9781119473916. URL: https://books.google.it/books?id=Sw5TDwAAQBAJ.

[3] E. W. Anderson, C. Fornell, and S. K. Mazvancheryl. «Customer Satisfaction and Shareholder Value.» In: *Journal of Marketing* (2004). DOI: 10.1509/jmkg.68.4.172.42723.

[4] A. Back. *Hashcash - A Denial of Service Counter-Measure*. 2002.

[5] L. Baresi, S. Guinea, and D. F. Mendonca. «A3Droid: A framework for developing distributed crowdsensing.» In: *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. 2016. DOI: 10.1109/PERCOMW.2016.7457103.

[6] J.A. Barnett. «Calculating Dempster-Shafer plausibility.» In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13.6 (1991), pp. 599–602. DOI: 10.1109/34.87345.

[7] B. Biswas and R. Gupta. «Analysis of barriers to implement blockchain in industry and service sectors.» In: *Computers & Industrial Engineering* 136 (2019), pp. 225–241. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2019.07.005.

[8] *Bitcoin average mining cost*. Accessed: 2022-10-21. URL: https://en.macromicro.me/charts/29435/bitcoin-production-total-cost.

[9] V. Buterin. *Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform*. 2014. URL: https://ethereum.org/en/whitepaper/.

[10] C. Cascarilla. *Paxos Standard*. 2019. URL: https://account.paxos.com/whitepaper.pdf.

[11] D. Chatzopoulos, S. Gujar, B. Faltings, and P. Hui. «Privacy preserving and cost optimal mobile crowdsensing using smart contracts on blockchain.» In: *2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. 2018. DOI: 10.1109/MASS.2018.00068.

[12] S. Clemhout. «The Ratio Method of Productivity Measurement: A Reply.» In: *The Economic Journal* (1965). DOI: 10.2307/2229457.

[13] A. Collins and R. Michalski. «The logic of plausible reasoning: A core theory.» In: *Cognitive Science* 13.1 (1989), pp. 1–49. ISSN: 0364-0213. DOI: https://doi.org/10.1016/0364-0213(89)90010-4.

[14] M. Cui, Y. Fei, and Y. Liu. «A Survey on Secure Deployment of Mobile Services in Edge Computing.» In: *Security and Communication Networks* 2021 (Jan. 2021), pp. 1–8. DOI: 10.1155/2021/8846239.

[15] P. Cuzzolin. «On the properties of relative plausibilities.» In: *2005 IEEE International Conference on Systems, Man and Cybernetics*. Vol. 1. 2005, 594–599 Vol. 1. DOI: 10.1109/ICSMC.2005.1571211.

[16] W. Dai. *b-money, an anonymous, distributed electronic cash system*. 1998. URL: http://www.weidai.com/bmoney.txt.

[17] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels. «Flash Boys 2.0: Frontrunning in Decentralized Exchanges, Miner Extractable Value, and Consensus Instability.» In: *2020 IEEE Symposium on Security and Privacy (SP)*. 2020, pp. 910–927. DOI: 10.1109/SP40000.2020.00040.

[18] J. L. Darby. «Tools for evaluating risk of terrorist acts using fuzzy sets and belief/plausibility.» In: *NAFIPS 2009 - 2009 Annual Meeting of the North American Fuzzy Information Processing Society*. 2009, pp. 1–5. DOI: 10.1109/NAFIPS.2009.5156446.

[19] S. De Angelis, L. Aniello, F. Lombardi, A. Margheri, and V. Sassone. «PBFT vs proof-of-authority: applying the CAP theorem to permissioned blockchain.» In: *Italian Conference on Cybersecurity*. Jan. 2018.

[20] A. P. Dempster. «Upper and Lower Probabilities Induced by a Multivalued Mapping.» In: *Ann. Math. Statist., vol. 38, no. 2, pp. 325–339* (1967).

[21] J. Devlin, M. Chang, K. Lee, and K. Toutanova. «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.» In: (2018). DOI: 10.48550/ARXIV.1810.04805.

[22] J. Dezert and F. Smarandache. *An introduction to DSmT*. Infinite Study, 2009.

[23] T. M. Fernández-Caramès and P. Fraga-Lamas. «Towards Post-Quantum Blockchain: A Review on Blockchain Cryptography Resistant to Quantum Computing Attacks.» In: *IEEE Access* 8 (2020), pp. 21091–21116. DOI: 10.1109/ACCESS.2020.2968985.

[24] MakerDAO Foundation. *The Maker Protocol: MakerDAO's Multi-Collateral Dai (MCD) System*. 2017. URL: https://makerdao.com/en/whitepaper/.

[25] Tether Foundation. *Tether: Fiat currencies on the Bitcoin blockchain*. 2016. URL: https://tether.to/wp-content/uploads/2016/06/TetherWhitePaper.pdf.

[26] J. Gilcrest and A. Carvalho. «Smart Contracts: Legal Considerations.» In: *2018 IEEE International Conference on Big Data (Big Data)*. 2018, pp. 3277–3281. DOI: 10.1109/BigData.2018.8622584.

[27] B. Graham. *intelligent investor: The Classic Text on Value Investing*. HarperCollins, 2005. ISBN: 9780060752613. URL: https://books.google.it/books?id=QFdUX7xFpZ0C.

[28] X. Guo and X. Xiao. «Influencing Factors of Asset Evaluation Quality.» In: *Proceedings of the 2020 3rd International Conference on Humanities Education and Social Sciences (ICHESS 2020)*. Atlantis Press, 2020. DOI: 10.2991/assehr.k.201214.628.

[29] S. Haber and W. S. Stornetta. «How to Time-Stamp a Digital Document.» In: *Journal of Cryptology* 3 (1999). DOI: 10.1007/BF00196791.

[30] R. E. Hall. «Struggling to understand the stock market.» In: *American Economic Review* (2001). DOI: 10.1257/aer.91.2.1.

[31] J. Y. Halpern and N. Friedman. «Plausibility measures and default reasoning: an overview.» In: *Proceedings. 14th Symposium on Logic in Computer Science (Cat. No. PR00158)* (1999). DOI: 10.1109/LICS.1999.782603.

[32] S. Herm, H. Callsen-Bracker, and H. Kreis. «When the crowd evaluates soccer players' market values: Accuracy and evaluation attributes of an online community.» In: *Sport Management Review* 17 (2014). DOI: 10.1016/j.smr.2013.12.006.

[33] *"IBM Developer Model Asset Exchange: Text Sentiment Classifier,"* Accessed: 2021-11-23. URL: https://github.com/IBM/MAX-Text-Sentiment-Classifier.

[34] *IBM Research, "IBM Project Debater,"* Accessed: 2021-11-23. URL: https://research.ibm.com/haifa/dept/vst/debating_data.shtml.

[35] G. Iovane, P. Di Gironimo, M. Chinnici, and A. Rapuano. «Decision and Reasoning in Incompleteness or Uncertainty Conditions.» In: *IEEE Access* (2020). DOI: 10.1109/ACCESS.2020.3003726.

[36] G. Iovane, R. Landi, A. Rapuano, and R. Amatore. «Assessing the Relevance of Opinions in Uncertainty and Info-Incompleteness Conditions.» In: *Applied Sciences* 12 (Dec. 2021). DOI: 10.3390/app12010194.

[37] G. Iovane, M. Nappi, M. Chinnici, A. Petrosino, A. Castiglione, and S. Barra. «A Novel Blockchain Scheme Combining Prime Numbers and Iris for Encrypting Coding.» In: Aug. 2019, pp. 609–618. DOI: 10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00117.

[38] G. Iovane, A. Rapuano, and P. Di Gironimo. «Blockchain-Based Iris Authentication in Order to Secure IoT Access and Digital Money Spending.» In: Feb. 2021, pp. 427–441. ISBN: 978-3-030-68820-2. DOI: 10.1007/978-3-030-68821-9_37.

[39] G. Iovane, L. Sensini, P. Di Gironimo, A. Rapuano, and A. Briscione. «TES: Token Evaluation System for Blockchain contexts.» In: *Interdisciplinary Mathematics* (2022). DOI: 10.1080/09720502.2021.1932852.

[40] M. Kadadha, H. Otrok, R. Mizouni, S. Singh, and A. Ouali. «SenseChain: A blockchain-based crowdsensing framework for multiple requesters and multiple workers.» In: *Future Generation Computer Systems* (2019). DOI: 10.1016/j.future.2019.12.007.

[41] C. Kandel, M. Klumpp, and T. Keuschen. «GPS based track and trace for transparent and sustainable global supply chains.» In: July 2011, pp. 1 –8.

[42] J. W. Kendrick. «Productivity Trends in the United States.» In: *Princeton University Press* (1961).

[43] J. W. Kendrick and B. N. Vaccara. «New Developments in Productivity Measurement and Analysis.» In: *University of Chicago Press* (1980).

[44] E. Kereiakes, D. Kwon, M. Di Maggio, and N. Platias. *Terra money: Stability and Adoption*. 2019. URL: https://whitepaper.io/document/587/terra-whitepaper.

[45] M. A. Khan and K. Salah. «IoT security: Review, blockchain solutions, and open challenges.» In: *Future Generation Computer Systems* 82 (2018), pp. 395–411. ISSN: 0167-739X. DOI: 10.1016/j.future.2017.11.022.

[46] H. Li, H. Zhou, and Q. Cai. «An asset evaluation method based on neural network.» In: *2010 2nd IEEE International Conference on Information Management and Engineering*. 2010. DOI: 10.1109/ICIME.2010.5477927.

[47] X. Li and C. Camerer. «Predictable Effects of Visual Salience in Experimental Decisions and Games.» In: *The Quarterly Journal of Economics* (2022). DOI: 10.1093/qje/qjac025.

[48] Y. Liu, X. Hao, W. Ren, R. Xiong, T. Zhu, K.R. Choo, and G. Min. «A Blockchain-based Decentralized, Fair and Authenticated Information Sharing Scheme in Zero Trust Internet-of-Things.» In: *IEEE Transactions on Computers* (2022). DOI: 10.1109/TC.2022.3157996.

[49] Y. Lo and F. Medda. «Assets on the blockchain: An empirical study of Tokenomics.» In: *Information Economics and Policy* (2020). DOI: 10.1016/j.infoecopol.2020.100881.

[50] H. A. G. Lopez and M. A. P. Cisneros. «Industry 4.0 & Internet of Things in Supply Chain.» In: *Proceedings of the 8th Latin American Conference on Human-Computer Interaction*. CLIHC '17. Antigua Guatemala, Guatemala: Association for Computing Machinery, 2017. ISBN: 9781450354295. DOI: 10.1145/3151470.3156646.

[51] Y. Lu and Y. Lin. «Using Hybrid Classifiers to Conduct Intangible Assets Evaluation.» In: *Int. J. Appl. Metaheuristic Comput.* (2016). DOI: 10.4018/IJAMC.2016010102.

[52] J. Madsen, A. Minniti, and F. Venturini. «Wealth Inequality in the Long Run: A Schumpeterian Growth Perspective.» In: *The Economic Journal* (June 2020). DOI: 10.2139/ssrn.3222420.

[53] R. Marcuzzo, J. Rech Graciano dos Santos, J. Siluk, D. Chaves, and V. Gerhardt. «Modelling for intangible assets evaluation in technology-based companies.» In: *International Joint Conference - CIO-ICIEOM-IIE-AIM (IJC)*. 2016.

[54] S. Nakamoto. «Bitcoin: A Peer-to-Peer Electronic Cash System.» In: *Cryptography Mailing list at https://metzdowd.com* (2009).

[55] L. Montiel Olea, P. Ortoleva, M. M. Pai, and A. Prat. «Competing Models.» In: *Econometrics: Econometric & Statistical Methods - General eJournal* (2021).

[56] A. Pinna and S. Ibba. «A Blockchain-Based Decentralized System for Proper Handling of Temporary Employment Contracts.» In: (2019). DOI: 10.1007/978-3-030-01177-2_88.

[57] G. Polya. «Mathematics and Plausible Reasoning: Induction and Analogy in Mathematics.» In: *Princeton, NJ, USA: Princeton Univ. Press* (1954).

[58] G. Polya. «Mathematics and Plausible Reasoning: Patterns of plausible inference.» In: *Princeton, NJ, USA: Princeton Univ. Press* (1990).

[59] S. Popov. «The tangle.» In: *White paper* 1.3 (2018).

[60] K. Qin, L. Zhou, and A. Gervais. «Quantifying Blockchain Extractable Value: How dark is the forest?» In: *2022 IEEE Symposium on Security and Privacy (SP)*. 2022, pp. 198–214. DOI: 10.1109/SP46214.2022.9833734.

[61] N. Rescher. «Plausible reasoning: An introduction to the theory and practice of plausibilistic inference.» In: *Philosophy and Rhetoric* 13.3 (1980).

[62] Digital Atmosphere SRLS. *Smart Lithium Coin (SLC), The Blockchain supply for Lithium Mining*. Accessed: 2022-04-20. 2019. URL: www.smartlithiumcoin.com.

[63] G. Sazandrishvili. «Asset tokenization in plain English.» In: *Journal of Corporate Accounting and Finance* 31.2 (2020), pp. 68–73. DOI: 10.1002/jcaf.22432.

[64] H. Scholten, T. Bartram, A. Kassahun, S. Kläser, R. Tröger, R.J.M. Hartog, A. Schilings-Schmitz, S. Meier, and R. Reiche. «Enabling Transparency in Meat Supply Chains: tracking & tracing for supply chain partners, consumers and authorities.» In: *Referate der 34.GIL-Jahrestagung*. Ed. by M. Clasen, M. Hamer, S. Lehnert, B. Petersen, and T. Brigitte. Gesellschaft für Informatik e.V.: Bonn, 2014, pp. 181–184. ISBN: 9783885796206.

[65] G. Shafer. «A Mathematical Theory of Evidence.» In: *Princeton, NJ, USA: Princeton Univ. Press* (1976).

[66] A. Shved and Y. Davydenko. «The analysis of uncertainty measures with various types of evidence.» In: *2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP)*. 2016, pp. 61–64. DOI: 10.1109/DSMP.2016.7583508.

[67] P. Singh and P. Singh Lamba. «Influence of crowdsourcing, popularity and previous year statistics in market value estimation of football players.» In: *Journal of Discrete Mathematical Sciences and Cryptography* 22.2 (2019), pp. 113–126. DOI: 10.1080/09720529.2019.1576333.

[68] N. Stifter, M. Eckhart, B. Brenner, and E. Weippl. «Avoiding Risky Designs When Using Blockchain Technologies in Cyber-Physical Systems.» In: (2019), pp. 1623–1626. DOI: 10.1109/ETFA.2019.8869163.

[69] J. Sunny, N. Undralla, and V. M. Pillai. «Supply chain transparency through blockchain-based traceability: An overview with demonstration.» In: *Computers & Industrial Engineering* 150 (2020), p. 106895. DOI: 10.1016/j.cie.2020.106895.

[70] C. Tsai, Y. Lu, Y. Hung, and D. Yen. «Intangible Assets Evaluation: The Machine Learning Perspective.» In: *Neurocomputing* (2015). DOI: 10.1016/j.neucom.2015.10.041.

[71] M. Varun, B. Palanisamy, and S. Sural. «Mitigating Frontrunning Attacks in Ethereum.» In: *Proceedings of the Fourth ACM International Symposium on Blockchain and Secure Critical Infrastructure*. 2022. DOI: 10.1145/3494106.3528682.

[72] C. Wang, Y. Huang, M. Shao, and D. Chen. «Uncertainty measures for general fuzzy relations.» In: *Fuzzy Sets Syst.* 360 (2019), pp. 82–96. DOI: 10.1016/j.fss.2018.07.006.

[73] Y. Wang, Z. Su, J. Li, N. Zhang, K. Zhang, K. R. Choo, and Y. Liu. «Blockchain Based Secure and Cooperative Private Charging Pile Sharing Services for Vehicular Networks.» In: *IEEE Transactions on Vehicular Technology* (2022). DOI: 10.1109/TVT.2021.3131744.

[74] W. E. Weber. «A Bitcoin standard: Lessons from the gold standard.» In: (2016). DOI: 10.34989/swp-2016-14.

[75] K. Wen, Y. Song, C. Wu, and T. Li. «A Novel Measure of Uncertainty in the Dempster-Shafer Theory.» In: *IEEE Access* 8 (2020), pp. 51550–51559.

[76] D.A. Wijaya, J.K. Liu, R. Steinfeld, D. Liu, F. Junis, and D.A. Suwarsono. «Designing Smart Contract for Electronic Document Taxation.» In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2019). DOI: 10.1007/978-3-030-31578-8_11.

[77] Y. Yan, B. Duan, Y. Zhong, and X. Qu. «Blockchain technology in the internet plus: The collaborative development of power electronic devices.» In: *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*. 2017, pp. 922–927. DOI: 10.1109/IECON.2017.8216159.

[78] M. Yang, T. Zhu, K. Liang, W. Zhou, and R. H. Deng. «A blockchain-based location privacy-preserving crowdsensing system.» In: *Future Generation Computer Systems* (2019). DOI: 10.1016/j.future.2018.11.046.

[79] L. A. Zadeh. «A Fuzzy-Algorithmic Approach to the Definition of Complex or Imprecise Concepts.» In: *Systems Theory in the Social Sciences: Stochastic and Control Systems Pattern Recognition Fuzzy Analysis Simulation Behavioral Models* (1976). DOI: 10.1007/978-3-0348-5495-5_11.

[80] S.Y.A. Zaidi, M.A. Shah, H.A. Khattak, C. Maple, H.T. Rauf, A.M. El-Sherbeeny, and M.A. El-Meligy. «An attribute-based access control for IoT using blockchain and smart contracts.» In: *Sustainability (Switzerland)* 13.19 (2021). DOI: 10.3390/su131910556.

[81] D.l Zentai. «On The Efficiency on The Lamport Signature Scheme.» In: *Land Forces Academy Review* 25.3 (2020), p. 99. DOI: 10.2478/raft-2020-0033.

[82] R. Zhang, R. Xue, and L. Liu. «Security and Privacy on Blockchain.» In: *ACM Comput. Surv.* (2019). DOI: 10.1145/3316481.

[83] N. S. Zivic. «Distributed Ledger Technology for Automotive Production 4.0.» In: *2020 28th Telecommunications Forum (TELFOR)* (2020), pp. 1–3. DOI: 10.1109/TELFOR51502.2020.9306594.