UNIVERSITÀ POLITECNICA DELLE MARCHE

FACOLTÀ DI ECONOMIA "GIORGIO FUÀ"

Corso di Laurea Magistrale in Data Science per l'Economia e le Imprese

Tesi di Laurea Magistrale

LM 56 - LM 91

INTERPRETABILITÀ DEI MODELLI DI MACHINE LEARNING

APPROCCIO S.A.F.E. SU DATI FINANZIARI

EXPLAINABLE MACHINE LEARNING

S.A.F.E. APPROACH FOR FINANCIAL DATA

Relatore:
Prof.ssa Maria Cristina Recchioni

Rapporto Finale di:
Alessandro Piergallini

Anno Accademico 2022/2023

Indice

Introduzione **1**

1 FinTech: la rivoluzione dell'IA in ambito finanziario **3**
1.1 Vantaggi, minacce e normative dei Big Data e dell'IA 3
1.2 Gestione del rischio informatico 20

2 Analisi di dati finanziari **33**
2.1 Reti neurali artificiali 33
2.2 Reti neurali in finanza 46
2.3 Problemi di selezione dei modelli 58

3 Interpretabilità dei modelli di machine learning **65**
3.1 Shapley Value 65
3.2 S.A.F.E. AI 72

4 Applicazione: previsione del prezzo del Bitcoin **77**
4.1 Analisi descrittiva 78
4.2 Definizione delle funzioni su R e Python 84
4.3 Applicazione 86
4.3.1 Rete neurale autoregressiva: NNAR 87
4.3.2 Reti neurali ricorrenti: LSTM & GRU 90
4.4 Risultati e confronto delle metodologie 100

Conclusioni **107**

Bibliografia **109**

Sitografia per la raccolta dei dati **113**

Introduzione

Negli ultimi anni è aumentato l'impiego dei Big Data in diversi settori, con la necessità di adottare modelli più complessi di analisi come quelli di *machine* e *deep learning*. Ultimamente è particolarmente aumentato l'interesse per l'adozione di sistemi di intelligenza artificiale (IA). Tali modelli sono, però, complicati sia da definire e modellare, sia da interpretare, diversamente dai classici modelli econometrici. La stessa Commissione Europea ha definito delle linee giuda per valutare e definire se una IA sia affidabile o meno.
In relazione al settore finanziario, verrà effettuata un'analisi sulla previsione del prezzo del Bitcoin con l'impiego delle reti ricorrenti su dati ad alta frequenza. Si dimostrerà che l'impiego delle classiche metriche, come l'RMSE, avranno alcune limitazioni, quali la dipendenza dell'unità di misura della variabile di risposta. Come metodo d'interpretabilità di qualsiasi modello di *machine learning* saranno presentati gli *Shapley-Value*, i quali saranno normalizzati attraverso l'impiego dello zonoide di Lorenz. Risolto il problema della scalabilità dei risultati, si adottano le metriche S.A.F.E. come metodologia di spiegabilità dei vari modelli. Tale procedura permetterà di superare i problemi connessi all'unità di misura delle previsioni e di realizzare dei confronti diretti ed immediati. Sarà possibile determinare una classifica su quali siano i principali regressori che influiscono maggiormente

sulla previsione del prezzo del Bitcoin. In ultima analisi, verranno proposti anche dei grafici *ad hoc* per presentare i risultati degli *Shapley-Value* anche sulle singole osservazioni giornaliere delle previsioni.

FinTech: la rivoluzione dell'IA in ambito finanziario

L'impiego dei Big Data e dell'Intelligenza Artificiale (I.A.) ha apportato numerose trasformazioni nelle modalità di acquisizione, gestione e utilizzo delle informazioni in vari settori, quali quello finanziario. Nel seguente paragrafo verranno trattate le caratteristiche chiave e le minacce associate ai Big Data e all'A.I., insieme alle relative regolamentazioni.

1.1 Vantaggi, minacce e normative dei Big Data e dell'IA

Dalla nascita di Internet (1991) è possibile rendere i dati accessibili a tutti e ovunque nel mondo.
Inizialmente si impiegavano solamente dati strutturati, organizzati sulla base delle regole predefinite del modello relazionale, per passare poi a dati non

strutturati, i quali sono più complessi e variegati. Nella nuova era digitale, i dati sono, dunque, di natura diversa (testi, immagini, video, documenti . . .), più difficili da raccogliere, ricercare e analizzare. Ne segue che tali nuovi dati, i c.d. *Big Data*, hanno introdotto notevoli cambiamenti e miglioramenti, ma anche nuove minacce. La rivoluzione dei *Big Data* non è connessa solo all'aumento della dimensione dei dati, ma soprattutto all'impiego di un'architettura in parallelo che permette di svolgere contemporaneamente più operazioni su più dati. Per chiarire il concetto, si presentano nel dettaglio le principali caratteristiche dei *Big Data*, raggruppabili nelle famose "*5V*":

1. *Volume*: nuovi dispositivi, quali l'IoT[1], riescono a raccogliere continuamente una grande mole di dati, la quale viene salvata in database che raggiungono almeno l'unità petabyte (circa 1 miliardo di byte).

2. *Variety*: vengono raccolti una vasta gamma di formati eterogenei di dati, soprattutto non strutturati, ovvero archiviati senza un formato predefinito.

3. *Velocity*: il flusso di dati viene prodotto molto velocemente e soprattutto in continuazione.

4. *Veracity*: qualità e affidabilità dei dati, soprattutto per sistemi di dati strutturati che sono generati dalle macchine. Quelli non strutturati, i quali possono essere generati solo dall'uomo, devono essere puliti e sistemati per divenire strutturati. Solo dopo tale fase, i dati potranno

[1]IoT è l'acronimo di *"Internet of Things"*, ovvero oggetti o dispositivi connessi ad Internet e a software per l'analisi e raccolta di dati.

essere utilizzati in diversi processi, quali Decision Making, Business Analytics e analisi previsionale.

5. *Value*: Probabilmente la caratteristica più importante, definibile come l'insieme di benefici economici che derivano dalla disponibilità dei dati. Avere ad esempio dei dati che sono continuamente aggiornati e ben strutturati permette alle aziende di prendere delle decisioni tempestivamente ed efficacemente.

È opportuno, inoltre, analizzare quali siano i fattori (Pascuzzi, 2020) che hanno generato il crescente impiego dei *Big Data*:

- *Fattore informativo*: con l'accesso ad Internet e il miglioramento della tecnologia è possibile ricavare una grandissima mole di dati, che vengono continuamente prodotti dall'uomo, come foto e post caricati sui social, e dalle macchine, quali sensori di misurazione della temperatura, informazioni raccolte dai satelliti meteorologici, sistemi di geolocalizzazione.

- *Fattore tecnologico*: la possibilità di caricare, analizzare e organizzare continuamente una grande mole di dati è dovuta soprattutto dal miglioramento della tecnologia, che negli ultimi anni è cresciuta in maniera esponenziale. Sono stati sviluppati nuovi sistemi quali il calcolo distribuito[2] e il *cloud computing*[3] utili per lo scambio di risorse informatiche.

[2]Il calcolo distribuito è caratterizzato dalla suddivisione della mole di dati da analizzare tra più computer che si scambiano in continuazione i risultati intermedi.

[3]Il cloud computing è una risorsa che fornisce la possibilità di usufruire di spazi di archiviazione online, quali *Dropbox* o *Google Drive*.

- *Nuove tecniche e metodologie*: l'avvento dei Big Data ha introdotto nuove tecniche denominate *"data analytics"* aventi l'obiettivo di analizzare ed estrapolare informazioni dalla grande mole di dati che vengono generati in continuazione. La nascita di nuove metodologie ha creato anche nuove professioni quali il data scientist, che estrae informazioni di valore dai dati attraverso l'identificazione di modelli di machine learning in grado di risolvere problemi complessi, e il data analyst che invece seleziona le informazioni utili per un determinato processo decisionale e per il raggiungimento di obiettivi specifici.

- *Fattore economico*: l'utilizzo dei Big Data consente una riduzione dei costi e la costruzione di modelli statistico-econometrici utili per prendere delle decisioni ottimali in relazione a uno specifico obiettivo aziendale. Non è un caso che sono nate nuove discipline quali la Business Perfomance Analytics (BPA) con la quale si utilizzano i dati aziendali per analizzare la customer satisfaction e allineare tutta l'organizzazione aziendale al raggiungimento di obiettivi specifici.

Pertanto, i Big Data rappresentano una grande fonte di informazioni utili per varie e molteplici finalità. È necessario comprendere che dietro l'innovazione di tali dati possano nascere diverse minacce, quali la violazione del diritto alla privacy e il c.d. *"Capitalismo della sorveglianza di massa"* (Zuboff, 2019). È nato un sistema, nel quale si ha una vera e propria *collisione* tra libertà e controllo. Sulla rete e sui social network ogni individuo condivide *liberamente* delle informazioni (post, commenti, foto ...) che vengono *estratte* sottoforma di dati, i quali sono utilizzati per attuare una profilazione.
La parola chiave del capitalismo della sorveglianza è appunto l'*inconsapevolezza*.

Si crede di dare ai sistemi un po' di dati, informazioni e in cambio di ricevere anche gratuitamente certi servizi. La realtà, però, è che i dati che vengono consapevolmente ceduti sono una frazione di quello che questi sistemi estraggono da ogni individuo. Da tutto ciò nasce un potere completamente nuovo e differente dal totalitarismo, ovvero il c.d. *potere strumentario*. Tale fenomeno non avviene soltanto nel mondo digitale, ma anche in luoghi fisici, quali le proprie case, automobili, le strade nelle varie città e i negozi (bar, ristoranti ...). Si è sviluppata, dunque, un'architettura digitale che è sempre più globale e più ubiqua[4] e onnipresente. A differenza del potere totalitario, il potere strumentario agisce senza la minaccia del terrore e della violenza. Si affida completamente all'inconsapevolezza di ogni individuo. Estrapola dati e informazioni rendendo ogni persona una materia prima.

Di conseguenza, le nostre informazioni, tradotte in dati, divengono il c.d. *nuovo petrolio*, ovvero nuove fonti di ricchezza e nuovi strumenti per trarre profitti per diverse aziende.

Il processo, in realtà, non termina con la profilazione, quest'ultima serve per fare poi *previsione*. I dati, una volta raccolti, subiscono un processo di trasformazione per divenire strutturati, così da impiegarli per svolgere analisi e predizioni. Alle aziende interessa capire le preferenze e abitudini di ogni soggetto, per poi modificare i suoi comportamenti e spingerlo ad effettuare, ad esempio, ulteriori acquisti. Tutto il processo di estrazione e previsione, è orientato all'ottenimento di profitti per l'azienda. Tale processo è gestito principalmente da algoritmi complessi e dall'impiego dell'IA.

Un esempio pratico, riportato dalla sociologa Zuboff, è la campagna presiden-

[4] *"Ubiquitous computing"*, termine coniato dall'informatico statunitense Mark Weiser (1952 – 1999).

ziale del 2016, dove l'organizzazione elettorale di Donald Trump ha utilizzato proprio questo sistema. I nostri dati, che normalmente vengono venduti agli inserzionisti pubblicitari, sono stati acquisiti dall'organizzazione elettorale di Trump per prendere di mira, nelle grandi città più importanti degli Stati Uniti d'America, gli elettori che avrebbero votato contro Trump. Venne attuata una strategia per far sì che tali elettori adottassero un determinato comportamento: non andare a votare, astenersi dall'affluenza alle urne e scegliere (o almeno credevano di scegliere) di non recarsi a votare per il candidato democratico (ad esempio di non andare a votare per la Clinton). L'obiettivo era dunque sopprimere, reprimere e obliterare il voto grazie a questi sistemi di controllo. Tutto ciò si è realizzato senza minacce, armi e violenza, è un sistema che agisce nell'ombra, fa credere di avere un controllo, un pensiero e un'autonomia nelle scelte, ma in realtà inconsapevolmente ogni soggetto viene plasmato e adattato alle loro esigenze. Avviene una nuova selezione che possiamo definire "innaturale". Non è come quella naturale (Darwin, 1859), poiché in questo caso non si ha un'evoluzione per esigenze naturali di sopravvivenza, ma si è inconsapevolmente costretti a cambiare per soddisfare le volontà di altri soggetti, quali grandi aziende.

Ma come riesce tale sistema a plasmare ogni soggetto a suo piacimento? Tristan Harris[5], nel docufilm *"The Social Dilemma"* di Netflix, analizza il fenomeno della *tecnologia persuasiva*, ovvero un modello applicato all'estremo per mezzo del quale si cerca di modificare il comportamento di un soggetto e convincerlo a fare una determinata azione. Tale tecnica viene accostata con un'altra, il c.d. *rinforzo positivo intermittente* che ha lo stesso funzionamen-

[5]Google Former Design Ethicist e Center for Humane Technology Co-Founder.

to delle slot machine e ha l'obiettivo di mantenerti costantemente connesso. Si cerca innanzitutto di dare sempre più contenuti in continuazione, si vuole innestare un'abitudine inconscia nel soggetto tale da essere programmato. Il problema è che non sai quando otterrai una determinata cosa e neanche se la otterrai. Ad esempio avere il telefono spento sopra il tavolo vicino a te e darti l'idea che se lo prendi potrebbe avere qualcosa di nuovo e interssante per te.

Tutto ciò non è un caso, è una tecnica di progettazione. Un ulteriore esempio, quando vieni taggato in una foto, nella notifica spesso ti arriva solo il messaggio *"sei stato taggato nella foto"* costringendoti ad aprire il cellulare e andare ad aprire quella mail o app per vedere la foto e i commenti ad essa associati. Come mai non si riceve direttamente la foto, ma si riceve un messaggio? Con questo sistema rimani connesso in più occasioni e il prossimo passaggio consiste nell'ottenimento di più coinvolgimento, partecipazione, condivisione e iscrizione in nuove app e siti. Si cerca dunque di far condividere da ogni individuo i propri dati, i quali divengono il *"nuovo petrolio"*. Ciò avviene tramite gli algoritmi *hacking of the will* che cercano di selezionare le notizie che ti interessano di più e gli argomenti dei siti dove passi più tempo. Una volta che entri nel meccanismo, avviene che condividi dei dati e dunque delle informazioni che verranno poi usate per analizzare il tuo comportamento e creare così un modello predittivo. Tutto il sistema ruota a soddisfare grandi aziende che hanno principalmente tre obiettivi:

1. *Coinvolgimento*: aumentare l'uso dei contenuti.

2. *Crescita*: condividere il sito dell'azienda a più individui, tramite ad esempio un passaparola positivo.

3. *Revenue*: Ottenere un profitto.

Nel processo di estrazione delle informazioni, di profilazione e previsione dei comportamenti dei vari individui, un ruolo importante lo acquisisce anche l'IA. Tutti gli algoritmi di apprendimento (machine learning) e quelli predittivi si basano sui Big Data attraverso i quali acquisire informazione e nuove conoscenze.
L'Intelligenza Artificiale è sempre più utilizzata in vari campi, quali quello giuridico (analisi automatica di atti e documenti, analisi predittive sull'esito di una decisione giudiziaria, ...) e finanziario (FinTech[6], *cfr.* *1.2 Gestione del rischio informatico*). La stessa Commissione europea nel 2018 ha elaborato una definizione[7] di IA:

Per "intelligenza artificiale" (IA) si intendono quei sistemi che mostrano un comportamento intelligente analizzando il proprio ambiente e compiendo azioni, con un certo grado di autonomia, per raggiungere obiettivi specifici. [...] *L'aumento della potenza di calcolo e della disponibilità dei dati e il progresso negli algoritmi hanno reso l'IA una delle tecnologie più importanti del 21° secolo.*

In tale comunicazione vengono presentate, anche, alcune delle preoccupazioni connesse all'impiego dell'IA, quali la perdita del proprio lavoro a causa

[6]La *Financial Technology* comprende tutti i servizi finanziari tradizionali che vengono ottimizzati e semplificati attraverso l'uso della tecnologia (ICT). Alcuni esempi di servizi FinTech sono: pagamenti digitali, blockchain, criptovalute, peer-to-peer lending.

[7]Comunicazione della Commissione al Parlamento Europeo, al Consiglio, al Comitato Economico e Sociale Europeo e al Comitato delle Regioni:Piano coordinato sull'intelligenza artificiale, 2018.

dell'automazione, problemi connessi alle responsabilità giuridiche dei sistemi che si basano sull'intelligenza artificiale e la difficoltà per le start up basate sull'IA di trovare risorse umane per gestire tali sistemi.
Per affrontare tali problematiche, la Commissione ha stabilito una strategia europea con l'obiettivo di sfruttare tutte le opportunità scaturite dall'impiego dell'IA.

Tale strategia sostiene un'IA "made in Europe" etica, sicura e all'avanguardia e si basa sui punti di forza scientifici e industriali dell'Europa, articolandosi su tre pilastri:

- *aumentare gli investimenti pubblici e privati nell'IA,*
- *prepararsi ai cambiamenti socioeconomici,*
- *garantire un quadro etico e giuridico adeguato.*

Il coordinamento a livello europeo è essenziale per garantire il successo di tale strategia.

Il c.d. *"Piano coordinato sull'IA"* si pone l'obiettivo di massimizzare l'effetto degli investimenti in intelligenza artificiale in Europa e a migliorare la posizione competitiva dell'UE a livello globale. Gli aspetti più importanti che il piano va ad affrontare sono:

1. Definizione di obiettivi comuni e sforzi complementari.
2. Rafforzamento della cooperazione tra Stati membri, Commissione EU e settore privato per sviluppare un'agenda strategica di ricerca comune

sull'IA e finanziare start-up innovative, così da favorire la ricerca in ambito di intelligenza artificiale.

3. Favorire una diffusione estesa delle tecnologie IA "affidabili" attraverso prove e sperimentazioni in condizioni reali.

4. Adattare i sistemi di formazione in relazione alla rapida evoluzione della tecnologia, in modo tale da generare risorse umane esperte in nuovi campi, quali l'IA, e allineate con il progresso tecnologico. Tutto ciò permette anche di migliorare e risolvere i problemi connessi al livello di occupazione e alla diffusione di nuovi mestieri.

5. Costruzione di un database europeo essenziale per l'IA, disponibile anche per il settore pubblico: è importante realizzare un sistema di dati affidabili, veritieri, disponibili e con un'adeguata infrastruttura. Tale sistema di dati dovrà rispettare il regolamento generale sulla protezione dei dati (GDPR).

6. Sviluppo di orientamenti etici a livello globale: è importante che la tecnologia, inclusa l'IA, rispetti i diritti fondamentali in modo tale da favorire e aumentare la fiducia delle persone nell'impiego di quel nuovo sistema.

7. Garantire la sicurezza delle infrastrutture e delle applicazioni dell'IA: bisogna analizzare in che modo l'IA possa migliorare la sicurezza, proteggere da attacchi informatici dannosi e/o involontari.

La sesta e settima voce del piano sono molto importanti, poiché di seguito nell'elaborato verrà trattato l'impiego della S.A.F.E. AI (*cfr.* *3.2 S.A.F.E. AI*), avente l'obiettivo di aiutare nell'interpretabilità di modelli black box. In particolare le metriche in relazione alla *Fairness* e alla *Sustainability* analizzano rispettivamente se è presente un problema di discriminazione del modello in relazione a differenti gruppi e se il modello è resistente agli attacchi informatici.

Per analizzare i vari aspetti, la Commissione UE (Pascuzzi,2020) ha istituito un gruppo di esperti dell'IA, il quale mira a sviluppare un progetto di orientamenti etici dell'IA. Gli esperti presentarono alla Commissione la versione finale degli orientamenti l'8 aprile del 2019 (*Ethics guidelines for trustworthy AI, 2019*), all'interno della quale si presentano sette requisiti chiave che una tecnologia IA deve soddisfare per essere considerata affidabile:

1. *Supervisione umana*: è fondamentale garantire dei meccanismi di supervisione adeguati quali:

    - Human in the loop: utilizzo dell'IA come supporto ad una serie di operazioni principalmente gestite dall'uomo.
    - Human on the loop: una tecnolgia, come l'IA, gestisce un intero processo, ma contemporaneamente l'esecuzione viene monitorata dall'uomo.
    - Human in command: tutte le fasi decisionali sono gestite dall'uomo.

Attraverso una supervisione umana si avrà più consapevolezza sull'IA impiegata, e sarà così possibile prendere decisioni informate, chiare e meno incerte.

2. *Sicurezza e robustezza tecnica*: necessità di sistemi resilienti, accurati e sicuri in caso di incidenti dannosi o involontari.

3. *Privacy e governance dei dati*: garantire il rispetto del diritto alla privacy e delle normative del GDPR, e definire un adeguato sistema di governance dei dati in relazione alla qualità e integrità dei dati. È anche necessario garantire un accesso legittimato ai dati.

4. *Trasparenza*: i sistemi di IA devono essere dotati di meccanismi di tracciabilità. In questa fase rientra il problema di interretabilità dei modelli black box, quali le reti neurali artificiali, per comprendere i risultati ottenuti. È, inoltre, fondamentale che ogni individuo sia consapevole di interagire con un sistema IA e comprenda i limiti di tale tecnologia.

5. *Equità*: i sistemi IA devono evitare di discriminare e devono essere accessibili a tutti, indipendentemente da qualsiasi disabilità.

6. *Rispetto sociale e ambientale*: i sistemi di intelligenza artificiale dovrebbero apportare benefici a tutti gli esseri umani, comprese le generazioni future. Occorre che siano sostenibili e rispettosi dell'ambiente e deve anche essere considerato il loro impatto sociale.

7. *Responsabilità*: riguarda essenzialmente l'affidabilità del funzionamento e dei risultati dell'IA. Si necessita, dunque, di una fase di validazione e verifica del sistema.

Analizzando i punti chiave stabiliti dal gruppo di esperti in IA istituito dalla Commissione UE, possiamo riassumere che per essere considerata affidabile, l'intelligenza artificiale deve essere:

- *Legale*: L'IA che si sta sviluppando deve rispettare le norme e regolamentazioni applicabili.
- *Etica*: deve rispettare tutti i principi e valori etici. Verificare, ad esempio, se l'IA in sviluppo non discrimini in base a informazioni quali etnia, identità di genere, orientamento sessuale di un individuo.
- *Robusta*: affidabile sia a livello tecnico (analisi sul bias e qualità dei risultati) che sociale.

In seguito, nel 2020[8] la Commissione UE ha stabilito, attraverso il gruppo di esperti, una nuova e più completa definizione di intelligenza artificiale:

"I sistemi di intelligenza artificiale (IA) sono sistemi software (ed eventualmente hardware) progettati dall'uomo che, dato un obiettivo complesso, agiscono nella dimensione fisica o digitale percependo il proprio ambiente attraverso l'acquisizione di dati, interpretando i dati strutturati o non strutturati raccolti, ragionando sulle conoscenze, o elaborando le informazioni derivate da questi dati e decidendo le migliori azioni da intraprendere per raggiungere l'obiettivo dato. I sistemi di IA possono usare regole simboliche o apprendere un modello numerico, e possono anche adattare il loro comportamento analizzando come l'ambiente è influenzato dalle loro azioni precedenti."

[8] Commissione europea, *Libro bianco sull'intelligenza artificiale - Un approccio europeo all'eccellenza e alla fiducia*, 2020.

Tale definizione è più chiara e comprensiva delle molteplici funzionalità dell'impiego dell'IA. Si discute anche dell'applicazione dell'IA in ambiti trasversali quali robotica e si pone l'obiettivo del miglioramento dell'interpretabilità di tali sistemi:

"Parallelamente, l'Europa continuerà a guidare il progresso per quanto riguarda le basi algoritmiche dell'IA, basandosi sulla propria eccellenza scientifica. È necessario creare collegamenti tra discipline che attualmente lavorano separatamente, come l'apprendimento automatico (machine learning) e l'apprendimento profondo (deep learning), che sono caratterizzati da una limitata interpretabilità e dalla necessità di una grande quantità di dati per addestrare i modelli e apprendere mediante correlazioni, e gli approcci simbolici, in cui le regole sono create mediante l'intervento umano. Combinare il ragionamento simbolico con le reti neurali profonde può aiutarci a rendere maggiormente spiegabili i risultati dell'IA."

Per avere un'esaustiva panoramica dell'intelligenza artificiale e dei Big Data, è interessante analizzare anche la questione relativa al diritto di privacy.
La disciplina del trattamento dei dati personali ha l'obiettivo di garantire a ogni individuo il controllo sulle informazioni che lo riguardano. La principale fonte è il regolamento (UE) 2016/679, meglio conosciuto come *"General Data Protection Regulation (GDPR)"*, i cui articoli presentano una definizione di dati personali e del loro trattamento (dalla fase di raccolta e registrazione, organizzazione e modifica, circolazione, per arrivare poi alla conservazione

o cancellazione/distruzione dei dati). Il trattamento deve essere inoltre lecito, corretto e trasparente nei confronti dell'individuo interessato, e deve essere limitato alla finalità per cui vengono raccolti i suoi dati. Dunque, se l'interessato accetta che vengano raccolti dei suoi dati per una determinata finalità, ad esempio nel caso in cui si accettano dei cookies di un sito, il titolare del trattamento dei dati deve raccogliere solo i dati utili per l'obiettivo concordato con l'interessato. I suoi dati non possono essere impiegati in maniera generica. Dal dodicesimo al ventunesimo articolo si presentano i diritti riconosciuti a ogni individuo in merito ai suoi dati personali, quali quello di informazione da parte del titolare del trattamento, di accesso ai dati, di rettifica, ovvero ottenere dal titolare del trattamento i propri dati aggiornati e resi veritieri. Una questione delicata è il diritto alla cancellazione/oblio dei propri dati dal titolare del trattamento. La propria identità personale e collettiva è generata e modellata dalla cooperazione tra memoria e oblio. In varie situazioni l'ordinamento introduce l'obbligo di ricordare per mantenere viva la memoria collettiva su eventi di grande importanza, ad esempio il Giorno della Memoria per commemorare le vittime dell'Olocausto. Il concetto della memoria è connesso con la libertà di informazione e il diritto di cronaca. Il diritto all'oblio è, invece, connesso con il diritto alla riservatezza ed ha subito diverse modifiche nel corso degli anni, scontrandosi con il diritto di cronaca.

Inizialmente per oblio si intendeva (Cassazione, 1998, n. 3679):

"Il giusto interesse di ogni persona a non restare indeterminatamente esposta ai danni ulteriori che arrecano al suo onore e alla sua reputazione la reiterata pubblicazione di una notizia in passato legittimamente divulgata."

Con la nascita di Internet, le informazioni rimangono sempre online, e il diritto alla riservatezza acquisisce un nuovo significato: non più come diritto a essere dimenticato, ma come diritto di contestualizzazione degli avvenimenti al fine di *"mantenere i caratteri di verità ed esattezza e quindi di liceità e correttezza"* (Cassazione, 2012, n. 5525). Si ha, dunque, sia il diritto alla cancellazione dei dati obsoleti e il diritto alla deindicizzazione dei dati da parte dei motori di ricerca. Quest'ultimi devono evitare di rimandare un risultato obsoleto ad una determinata ricerca (o query), ed evitare di ripresentare vecchie notizie come se fossero attuali.
Il ventunesimo articolo tratta, invece, del diritto di opposizione dell'interessato al trattamento dei suoi dati personali, compresa la profilazione. Il titolare del trattamento dovrà astenersi dal trattamento di quei dati, salvo che riesca a dimostrare l'esistenza di motivi legittimi per procedere al trattamento dei dati, quali l'accertamento o la difesa di un diritto in sede giudiziaria. Nel dettaglio, l'art. 21 specifica che se i dati sono utilizzati per finalità di marketing diretto, allora l'interessato potrà opporsi in qualsiasi momento, e i suoi dati non dovranno più essere oggetto di trattamento dal titolare per tali finalità. In contesti sociali informativi, l'interessato potrà esercitare l'opposizione solo con mezzi automatizzati, mentre per contesti di ricerca scientifica o storica

o per fini statistici, potrà opporsi salvo se il trattamento è necessario per l'esecuzione di un compito di interesse pubblico. Pertanto il diritto alla privacy è riassumibile non come un semplice *"diritto ad essere lasciati soli"*, o diritto alla riservatezza, ma bensì come un diritto di mantenere il controllo sulle proprie informazioni. Il GDPR, ovviamente, non si limita solo a definire le regole del trattamento e i diritti dell'interessato, ma specifica anche la protezione delle persone nel trattamento dei propri dati personali attraverso il c.d. *"diritto cogente"* (hard law), il quale è direttamente applicabile e la cui violazione è seguita da una specifica sanzione.

La continua diffusione e utilizzo di Internet ha amplificato le minacce sulla tutela dei dati personali, ad esempio, il problema menzionato in precedenza relativo al capitalismo della sorveglianza di massa. L'art. 40 del regolamento tratta appunto dei codici di condotta per garantire la corretta applicazione del GDPR in vari settori e piccole e medie imprese. Molti siti web definiscono, inoltre, una propria *privacy policy*, ovvero un proprio regolamento sul trattamento dei dati personali dei navigatori. Per rafforzare e rendere sicuro il proprio sito, vengono impiegati anche dei sigilli (*privacy seals*), ovvero dei marchi che certificano al navigatore che tale sito ha raggiunto determinati standard di affidabilità, e che sia dunque sicuro.

Questi concetti sono fondamentali per avere un'adeguata panoramica dei vantaggi e minacce relative ai Big Data e all'IA. Nel prossimo paragrafo analizzeremo, invece, i rischi informatici che un'azienda incontra nella gestione di una grande mole di dati, mostrando particolare attenzione anche per la FinTech.

1.2 Gestione del rischio informatico

La diffusione di Internet, dei Big Data e dell'IA, ha generato una modernizzazione di vari settori, nei quali la tecnologia dell'informazione (IT) è divenuta una componente fondamentale. In ambito finanziario il connubio con l'IT ha dato vita alla *FinTech*.
Nella Tecnofinanza esistono due fenomeni (Osservatori Digital Innovation): il primo consiste nell'ottimizzazione dei servizi finanziari (Fintech-Fin), l'altro nello sviluppo di nuove tecnologie (Fintech-Tech). Indipendentemente dalla tipologia, entrambi hanno generato molti vantaggi quali:

- Un aumento del grado di interconnessione tra differenti imprese.
- Personalizzazione della customer experience: attraverso l'impiego dell'IA e l'analisi dei Big Data è possibile offrire soluzioni finanziarie personalizzate alle esigenze specifiche di ogni cliente.
- Sistemi più economici ed efficienti rispetto al sistema bancario tradizionale.
- Maggiore potenziale di crescita e diffusione nel mercato: le società fintech non sono vincolate dalle normative delle banche tradizionali, riescono così ad espandersi più velocemente.
- Sistemi automatizzati grazie all'impiego di tecnologie innovative, quali l'IA e il cloud computing. Tali sistemi sono anche più rapidi e soggetti a meno errori.

- Elevata versatilità: ci sono molteplici utilizzi per la fintech. I principali trend sono:

    1. Integrated finance: combinare servizi finanziari con altri di altra natura (dunque non finanziari).
    2. BNPL: implementazione nei siti di e-commerce di sistemi di prestito "Buy Now Pay Later", sempre più richiesti dagli utenti.
    3. Intelligenza Artificiale: molto utilizzata per la gestione del rischio informatico, quali la lotta contro le frodi bancarie. Viene impiegata anche per analizzare una grande mole di dati sulle transazioni.
    4. Blockchain e criptocurrency: inerente principalmente al campo del trading.
    5. Vertical banking: soddisfare le richieste specifici segmenti di clientela.

- Una regolamentazione più flessibile, rispetto alla banca tradizionale. Questo aspetto è però molto delicato, da un lato è un vantaggio perché velocizza diverse procedure e l'innovazione, ma allo stesso tempo rende il sistema più rischioso.

Diverse aziende di piccole, medie e grandi dimensioni, pubbliche e private, hanno iniziato così ad implementare sempre più sistemi IT.
Precedentemente è stato analizzato come la definizione dell'intelligenza artificiale è cambiata, divenendo più accurata e precisa grazie anche alla creazione e intervento di un gruppo di esperti. La Commissione UE, però, ha anche

definito una regolamentazione sull'IA (Commissione Europea, 2021) con l'obiettivo di ridurre gli oneri amministrativi e finanziari per le piccole e medie imprese. Nonostante la definizione delle caratteristiche tali per considerare una IA affidabile, rimane comunque un problema: i rischi e le conseguenze indesiderate. Il problema è connesso alla poca interpretabilità e trasparenza dei risultati, ovvero non sempre sono chiare le decisioni e previsioni attuate dall'intelligenza artificiale. Questa mancanza di trasparenza rende difficile valutare se qualcuno sia stato danneggiato ingiustamente, come ad esempio in processi decisionali per l'assunzione o nell'erogazione di servizi pubblici.

É stato definito un approccio basato su quattro tipologie di rischio:

1. *Minimal Risk*: è possibile impiegare il sistema basato sull'intelligenza artificiale. In tale categoria rientrano spesso le applicazioni, i videogiochi e filtri antispam.

2. *Limited Risk*: si richiede un obbligo di trasparenza, poichè sono principalmente sistemi IA che interagiscono con l'uomo, quali il chat-bot. Gli utenti devono essere informati di "rapportarsi" con una macchina, in modo tale da poter decidere se proseguire e affidarsi oppure se preferire un'interazione diversa. Il fornitore del servizio è, dunque, obbligato ad informare la persona fisica che tale sistema venga utilizzato, ad esempio, per rilevare emozioni, manipolare audio, immagini o video.

3. *High Risk*: è consentito il loro utilizzo solo dopo una valutazione *ex ante* e qualora soddisfi determinati requisiti, in base al campo nel quale vengono impiegati. Tutti i sistemi di identificazione biometrica, ovvero sulla base di una o più caratteristiche fisiologiche o comportamenta-

li, sono considerati ad alto rischio e richiedono norme severe. Sono, inoltre, vietati negli spazi pubblici anche per fini di sicurezza. Alcuni degli obblighi da rispettare, prima che venga introdotto nel mercato, sono le valutazioni dei rischi, la loro mitigazione, l'analisi della qualità dei dati che impiegherebbero per svolgere i vari compiti, la necessità di una documentazione dettagliata relativa al funzionamento e allo scopo di tale IA, informazioni per l'utente, sistema di robustezza, sicurezza e precisione, un sistema di tracciabilità dei risultati. Alcuni settori nei quali l'IA potrebbe risultare ad alto rischio sono nelle infrastrutture critiche[9], nella formazione professionale, sistemi automatici di selezione dei curriculum vitae per le procedure di assunzione e amministrazione della giustizia.

Nel momento in cui il sistema viene introdotto nel mercato, le autorità sono incaricate della vigilanza del mercato, gli utenti devono assicurare una sorveglianza costante e i fornitori sono tenuti a monitorare anche dopo la commercializzazione. In caso di gravi inconvenienti e malfunzionamenti, i fornitori e gli utenti devono segnalarli.

4. *Unacceptable Risk*: considerati una minaccia per la sicurezza di ogni individuo e per tale ragione vietati. Ad esempio, sistemi pubblici che scansionano il volto e identificano automaticamente le persone, oppure anche giocattoli dotati di assistenza vocale che incoraggia comportamenti pericolosi.

[9]Le infrastrutture critiche sono quei sistemi fisici e virtuali la cui compromissione avrebbe effetti significativi sulla salute e sicurezza degli individui. Caratteristiche fondamentali sono la resilienza e la conitnuità operativa per poter garantire il servizio in seguito ad incidenti ed interruzioni. Alcuni esempi sono: settore dei trasporti, distribuzione, sanità, protezione civile, banche e telecomunicazioni.

Di conseguenza, in caso di alto rischio, avviene un costante monitoraggio per verificare se effettivamente quell'intelligenza artificiale sia gestibile. In tali casi, dopo aver sviluppato un sistema IA classificato ad alto rischio, si necessita di valutarne la conformità tramite un organismo notificato. Nel caso cui risultasse conforme, l'organismo notificato procederà con la registrazione di quel determinato sistema di intelligenza artificiale in un database europeo e l'assegnazione di un marchio *CE* che ne certifichi la conformità. Successivamente viene immesso nel mercato per essere adoperato in differenti settori. Solo in caso di segnalazioni di malfunzionamenti specifici, l'organismo notificato effettuerà un'ulteriore esame di conformità per quel determinato sistema IA. É necessario verificare l'efficacia di un approccio basato sul rischio, attraverso la valutazione d'impatto intrapresa dall'UE. In quest'ultima si sviluppano quattro differenti opzioni, la prima inerente alla definizione dell'IA solo su base volontaria attraverso un'etichettatura, a livello UE, per consentire ai fornitori di applicazione di intelligenza artificiale di certificare la conformità e affidabilità dei propri sistemi. La seconda opzione valuta i rischi e determina una definizione di IA specifica per ogni settore, prevedendo la stesura di leggi *ad hoc* per risolvere i rischi specifici legati a determinate applicazioni. In terzo luogo viene, invece, proposta una definizione orizzontale e una metodologia per:

- *Valutazione del rischio elevato*: uno strumento legislativo orizzontale applicabile a tutti i sistemi IA immessi nel mercato e impiegati dall'UE con un approccio proporzionato al rischio.

- *Codici di condotta ad alto e basso rischio*: combina i requisiti e obblighi vincolanti per i sistemi ad alto rischio con i codici di condotta volontari

di quelli a rischio non elevato.

La quarta e ultima opzione è l'impiego di una definizione orizzontale senza una differenzazione, diversamente dalla precedente proposta.
Come quest'ultima, ogni titolare e utente avranno gli stessi requisiti, ma potranno in questo caso essere applicati a tutti i sistemi IA indipendentemente dal livello di rischio.
Dopo aver approfondito ulteriormente la regolamentazione dell'UE sull'impiego dell'intelligenza artificiale per le piccole e medie imprese, è fondamentale specificare che la rivoluzione di questi nuovi sistemi ha anche prodotto nuove minacce per le aziende, in particolare il c.d. *"cyber risk"*.
Il rischio informatico (Aldasoro et al., 2022) si traduce in perdite finanziarie, interruzioni e danni all'immagine o reputazione dell'organizzazione. Molte imprese si affidano, anche, ai sitemi cloud per aumentare il grado di interdipendenza con altre aziende, con la possibilità, però, di accrescere l'esposizione ai rischi informatici. Il cloud computing permette alle aziende di affittare potenza di calcolo e spazio di archiviazione, ne segue che alcuni costi fissi divengono marginali e l'impresa acquisisce una maggiore flessibilità nella gestione delle proprie operazioni. Il vantaggio è soprattutto per le piccole-medie imprese, che riescono a dotarsi di meno risorse in IT, rispetto alle grandi aziende. L'impiego di sistemi cloud permette di ridurre il consumo energetico e l'emissione di carbonio.
Tale sistema, dunque, è ottimale anche per l'ambiente, però è caratterizzato anche da rischi, quali la nascita di nuove opportunità per i criminali informatici e una crescente complessità delle infrastrutture digitali. Quest'ultimo aspetto potrebbe avere come effetto un aumento della probabilità di guasti

e interruzioni, dunque di problemi nella gestione del rischio informatico.
In generale gli attacchi informatici sono più probabili in settori con una concorrenza di mercato poco intensa, quali il commercio all'ingrosso, i trasporti e le comunicazioni, poichè c'è più opportunità di crescita e una maggiore interconnessione tra aziende. Il settore finanziario è spesso mirato ed esposto a un elevato numero di attacchi informatici a causa della sua elevata esposizione all'IT e per la sua posizione di intermediazione creditizia.
Per aumentare il grado di sicurezza è, innanzitutto, necessaria una regolamentazione. In ambito finanziario si ha, ad esempio, il *Comitato di Basilea* (BCBS, 2018) per la gestione del rischio informatico e il miglioramento della vigilanza bancaria.
Nella newsletter del 2018, Pablo Hernández de Cos, presidente del Comitato e governatore della Banca di Spagna, ha mostrato la crescente preoccupazione in merito al rischio informatico, aggravatosi con l'avvento del Covid-19, e la necessità delle banche di migliorare la propria resilienza alle minacce e agli incidenti legati alla sicurezza informatica.
L'obiettivo consiste nel definire misure per rafforzare la sicurezza informatica delle banche, integrandole con i principi di resilienza e rischio operativo. Il Comitato definisce la resilienza operativa come la capacità di una banca di riuscire a svolgere le proprie operazioni critiche anche in caso di interruzioni, dunque di difendersi da minacce e possibili disfunzioni e minimizzare tutte le possibili perdite e problematiche. In tale contesto è importante definire anche la tolleranza alle perturbazioni, ovvero il livello di disturbo provocato da vari rischi operativi che la banca è disposta a tollerare in scenari gravi, ma plausibili.

Il Comitato ha definito una serie di principi di resilienza operativa in ambito di governance, gestione del rischio operativo, pianificazione e sperimentazione della business continuity, mappatura delle operazioni critiche, gestione delle dipendenze da terzi, gestione degli incidenti, tecnologie ICT resilienti e la cyber security.

In relazione alla nosra analisi, specificheremo solo il settimo e ultimo principio, il quale afferma che le banche devono garantire una solida infrastruttura informatica con programmi di protezione, rilevamento, risposta e ripristino regolarmente testati, e devono fornire informazioni tempestive e rilevanti per la gestione del rischio e il supporto del processo decisionale, per agevolare l'esecuzione delle operazioni critiche.

Ogni istituto bancario deve possedere e dotarsi di un documento di *ICT policy*, inerente ai requisiti di governance, alla sicurezza informatica, sulla base della valutazione dei rischi, e ai piani di continuità operitiva. Serve, ad esempio, per gestire i controlli di accesso ai dati e per la protezione degli asset informativi più importanti.

Riprendiamo in dettaglio l'approfondimento sul cyber risk per verificare se effettivamente un maggior investimento in sistemi IT, possa ridurre i danni e le perdite derivanti da danni informatici. In generale, si ha che a fronte di una maggiore spesa in sistemi IT per l'impresa, quali l'impiego dell'intelligenza artificiale e sistemi di cloud computing, risulta esserci una futura riduzione dei costi derivanti da incidenti informatici.

In un'analisi condotta su 137 164[10] incidenti informatici, dei quali l'86% degli episodi sono avvenuti in America, (Aldasoro et al.,2022) si mostra un

[10]Dati raccolti e forniti dall'*Advisen cyber loss database*.

fenomeno di crescita della frequenza di incidenti informatici fino al 2016, per poi stabilizzarsi. Per ogni evento registrato, vengono riportate una serie di informazioni, quali: tipologia di incidente informatico, quanti dati sono stati rubati, la data dell'incidente, fonte e tipologia della perdita, l'attore (es. terroristi), tipologia di società (pubblica o privata), il numero di dipendenti, codici NAICS per identificare il settore e sotto-settori economici, il luogo in cui è avvenuto l'evento, l'importo della perdita e la dimensione, in termini di ricavi, dell'impresa.
É importante considerare che il costo generato da un evento informatico è caratterizzato da quattro componenti: costo diretto, indiretto e opportunità del danno, e costi di mitigazione relativi all'investimento in IT.

É stato riscontrato che generalmente la frequenza degli eventi informatici è caratterizzata da un trend positivo nel corso degli anni, però quest'ultima, e anche i relativi costi, differiscono in base al settore in cui opera un'impresa. Per il settore finanziario e assicurativo si ha una frequenza più alta, ma un costo medio non molto elevato dovuto all'alto livello di resilienza, adottato in merito al Comitato di Basilea. Il settore del commercio e dell'ingrosso è caratterizzato dai costi più alti. Il trend positivo della frequenza potrebbe essere giustificato da due aspetti: numerosi quadri normativi che incoraggiano la segnalazione degli incidenti informatici, oppure il fatto che per eseguire un attacco (*cyber attack*), non sono più necessarie competenze informatiche di alto livello. Ne segue che attualmente anche chi non possiede determinate competenze, è in grado di realizzare un cyber attack.

Le principali tipolgie di attacchi informatici, considerate nell'analisi, sono:

- *Security Incident*: evento di compromissione dei sistemi di un'azienda, oppure il fallimento delle precauzioni adottate per proteggerli. I principali incidenti di sicurezza sono: tentativi di accesso non autorizzati al sistema, negazione o interruzione involontaria di un servizio, archiviazione e modifica non autorizzata dei dati e malware.

- *Data Breach*: consiste nella violazione dei dati personali, attraverso la loro distruzione, perdita, modifica e divulgazione non autorizzata.

- *Phishing*: truffe online attraverso messaggi di posta elettronica ingannevoli. Sono spesso mail apparentemente provenienti da istituti bancari e siti web, i cui messaggi invitano il ricevente a fornire i propri dati riservati, quali le credenziali di accesso al conto bancario, così da potervi accedere. Per convincere l'utente, di solito, il messaggio presenta un collegamento ipertestuale che sembra dirigere al sito web autentico dell'ente di credito o del servizio a cui ci si è precedentemente registrati. Tuttavia, il sito a cui si accede è stato deliberatamente creato per assomigliare perfettamente a quello originale. Se l'utente inserisce le proprie informazioni riservate su questo sito contraffatto, tali dati finiranno nelle mani dei criminali.

- *Skimming*: installazione di dispositivi illegali sui bancomat per poter catturare i dati della carta di pagamento.

- *Privacy Violation*: trattamento illecito di informazioni private, come le password dei clienti.

Gli attacchi informatici più frequenti riguardano la violazione della privacy, mentre il phishing e la scrematura (o skimming) sono quelli caratterizzati da costi più elevati.

É, però, necessario valutare se un attacco è dannoso, ovvero volontario e realizzato da soggetti specifici con l'obiettivo di rubare dati. In media gli attacchi dannosi hanno costi inferiori, ma è opportuno evidenziare che nel caso in cui l'attaccante non venisse fermato, i danni possono diventare molto estesi e dunque molto elevati.

La tipologia di attacco è uno dei driver in merito al cyber risk. Altri fattori da considerare sono la tipologia di eventi e la relazione con la dimensione dell'azienda.

Nel caso di eventi connessi, ovvero che colpiscono più aziende, si riscontrano in media costi attesi più elevati. In particolare, un aumento unitario (Aldasoro, 2022) del numero di aziende interessate si traduce in un aumento di circa il 2,6% dei costi attesi.

L'ultimo driver è la correlazione positiva tra dimensione aziendale, in termini di ricavi, e i costi dei cyber attack, caratterizzati però da un'elasticità non molto elevata.

Sicuramente un aumento dell'investimento nella sicurezza informatica può aiutare a gestire il rischio informatico, mitigare i costi dei possibili incidenti e ridurre le perdite derivanti da una violazione informatica, ma non possiamo considerare questa correlazione come causalità. Infatti sembra esserci una correlazione tra spesa in IT e protezione dell'azienda da incidenti non dolosi, ma non possiamo parlare di causalità poiché l'investimento, ad esempio, in un nuovo hardware potrebbe non essere un investimento diretto alla

sicurezza.

Ovviamente l'investimento in IT è anche connesso all'utilizzo dell'intelligenza artificiale. Dopo aver approfondito tutti i regolamenti sull'impiego dell'IA e sulla gestione del rischio informatico, nel prossimo capitolo verrà trattato in dettaglio il funzionamento della rete neurale, a confronto con i modelli classici (white-box). L'obiettivo sarà poi presentare delle metriche per aiutare l'interpretabilità dei sistemi black-box, in grado di garantire una valutazione complessiva dell'affidabilità delle applicazioni dell'IA in finanza.

Analisi di dati finanziari

In questo capitolo verranno approfondite le principali tecniche di analisi e previsione dei dati in serie storica, in particolare per quelli finanziari ad alta frequenza. L'obiettivo sarà confrontare i modelli classici, con le reti neurali feed-forward (NNAR) e ricorrenti (RNN).

2.1 Reti neurali artificiali

La previsione delle serie storiche, diversamente da altre tipologie di dati, deve tener conto del concetto di memoria, ovvero la dipendenza da situazioni passate. In generale, si ha che un fenomeno analizzato al tempo t sia più simile al valore raccolto all'istante *t-1*, rispetto a quelli risalenti a periodi più "lontani", in un orizzonte temporale. Per le serie temporali si utilizza come strumento di analisi il *processo stocastico*. Nel caso in cui si utilizzino i modelli classici, è fondamentale verificare che un processo stocastico sia stazionario.

Esistono due concetti di stazionarietà (Ruey S. Tsay, 2005):

1. *Stazionarietà forte*: è una proprietà sulle distribuzioni, secondo la quale dato un insieme di n osservazioni, non necessariamente consecutive, di una serie storica Y_t, la distribuzione congiunta $(y_{t_1}, \ldots, y_{t_n})$ è identica a quella di $(y_{t_1+s}, \ldots, y_{t_n+s})$, per ogni n e s. Ne segue, dunque, che la funzione di densità rimane inalterata e non dipende dal tempo stesso. Tale condizione, però, è difficile che si verifichi empiricamente, per questo esiste una seconda definizione.

2. *Stazionarietà debole*: Un processo è "debolmente" stazionario se valgono tre condizioni:

   - Il valore atteso del processo non cambia nel corso del tempo:

     $$\mathrm{E}(Y_t) = \mu$$

   - Il processo ha una varianza costante, e dunque non dipendente dal tempo:

     $$\mathrm{V}(Y_t) = \sigma^2$$

     Graficamente è come se il processo fosse un *White-Noise*.

   - Vale il concetto di persistenza. Ad esempio, la coviarianza tra oggi e ieri deve essere uguale a quella tra un altro qualsiasi giorno e quello ad esso antecedente o successivo. Generalizzando il concetto, invece di far riferimento a "ieri ed oggi", è possibile considerare un intero k:

     $$COV(y_t, y_{t-k}) = \gamma_k \qquad \forall k$$

     Questa espressione dipende solo da k, dunque ciò che è successo k volte fa (se $k = 1$ e i dati sono raccolti giornalmente, allora si

intende "un giorno fa"), condiziona il valore di oggi. Ne consegue che esiste una memoria.

La seconda e terza condizione, sono riassumibili attraverso il concetto di autocorrelazione, indicando la varianza di Y_t con γ_0:

$$\rho_k = \frac{\gamma_k}{\gamma_0}.$$

Dunque due osservazioni distanti k, hanno la stessa covarianza. Il processo sarà considerato stazionario se avrà memoria breve e, di conseguenza, all'aumentare del valore di k le covarianze tenderanno al valore nullo. Il concetto di stazionarietà è fondamentale per le analisi con i modelli classici. Per verificare se una serie possa essere considerata stazionaria, vengono realizzati dei test di radice unitaria, in particolare: *"Augmented Dickey-Fuller"* (ADF), *"Phillips-Perroh"* (PP) e il *KPSS*[11] test. Nel caso in cui il processo venga confutato dai test come non stazionario, o integrato, è possibile risolvere il problema attraverso la differenzazione della serie. In finanza si possono riscontrare delle difficoltà nel confronto dei prezzi degli asset, in particolare un'azienda più grande potrebbe avere un prezzo delle azioni più elevato rispetto a un concorrente, ma i prezzi di quest'ultimo potrebbero aumentare rapidamente, mentre quelli dell'impresa più grande potrebbero rimanere relativamente stabili. Ne segue che per effettuare un confronto, è fondamentale considerare i prezzi in termini relitivi. Si rappresenti i prezzi di un asset al

[11] Kwiatkowski–Phillips–Schmidt–Shin test (1992), *Journal of Econometrics*.

tempo t come p_t, il rendimento è espresso come:

$$return_t = \frac{p_t - p_{t-1}}{p_{t-1}}.$$

Poiché i prezzi sono una variabile non negativa e $return_t > -1$, il rendimento è esprimibile come:

$$return_t = \frac{p_t}{p_{t-1}} - 1.$$

Indicando con r_t il $log(\ return_t{+}1\ )$, è possibile effettuare una differenzazione in scala logaritmica:

$$r_t = \log\left(\frac{p_t}{p_{t-1}}\right) = \log(p_t) - \log(p_{t-1})$$

Nel capitolo finale, nel quale viene mostrata l'applicazione di tali modelli su dati finanziari, quest'ultimi saranno trasformati attraverso una differenzazione in scala logaritmica dei prezzi. Nei modelli autoregressivi integrati con media mobile ($ARIMA$) è fondamentale effettuare dei test diagnostici per evitare che ci siano problemi di mispecificazione in media e in varianza. I dati finanziari, diversamente da quelli economici, sono caratterizzati da un'alta frequenza che genera problemi di eteroschedasticità condizionale, risolvibili attraverso l'impiego dei modelli $ARCH$ e $GARCH$. Anche se $ARIMA$ e altri modelli lineari sono di solito più efficaci nella previsione a breve termine, questi modelli tendono a fallire nel prevedere con maggiore precisione il comportamento del mercato azionario nel tempo a causa della sua complessità. Una delle alternative per superare questa limitazione è stata l'implementazione di un tipo speciale di Reti Neurali Artificiali (ANN) chiamato Reti

Neurali Ricorrenti (*RNN*). Si noti però che le reti neurali sono meno interpretabili rispetto ai modelli classici.
Prima di procedere con l'analisi delle reti ricorrenti, è opportuno comprendere il funzionamento delle *ANN*.
Le reti neurali artificiali (Pang-Ning Tan et al., 2018) sono un modello di classificazione in grado di definire confini decisionali complessi e non lineari dai dati. L'idea base è l'imitazione del cervello umano, caratterizzato dai neuroni, collegati tra loro attraverso gli assoni. Ogni volta che un neurone viene stimolato, trasmette delle attivazioni nervose ad altri neuroni i quali recepiscono l'impulso attraverso i dendriti. La forza di contatto tra un dendrite e un assone è definita sinapsi, che indica il grado di connessione tra i neuroni. L'apprendimento nel cervello umano è dato dalla modifica della forza di connessione sinaptica tra i neuroni. Analogamente al cervello umano, le reti neurali artificiali sono caratterizzate dai *nodi*, connessi tra di loro attraverso dei legami. Il grado di connessione, espresso attraverso il *peso* del legame, rappresenta proprio la sinapsi tra neuroni.
Le reti neurali multi-strato sono in grado di definire bordi decisionali non lineari e sono caratterizzate da tre strati:

- *Input Layer*: ogni nodo rappresenta un attributo, numerico o binario. Nel caso in cui ci siano attributi categorici, allora ogni nodo andrà a rappresentare ogni categoria per quell'attributo.

- *Hidden Layer*: ogni nodo è un'unità di elaborazione, che opera sui segnali ricevuti dai nodi di input, o da quelli dello strato precedente, nel caso in cui ci siano più livelli nascosti, e produce un valore di attivazione da trasmettere poi allo strato successivo.

- *Output Layer*: lo strato finale elabora i valori di attivazione trasmessi dall'ultimo livello nascosto per realizzare le previsioni delle variabili. Nel caso di classificazione binaria, l'output avrà un solo nodo con la classe binaria predetta.

Il primo livello nascosto opera direttamente sugli attributi di input e cattura le caratteristiche più semplici, i successivi livelli nascosti li combina e ne determina altre più complesse. Ne segue una gerarchia di caratteristiche a diversi livelli di astrazione che vengono poi combinati con i nodi output per la fase previsionale. Questa architettura è tipica delle reti neurali *feed-forward*, ovvero l'informazione passa da uno strato a quello ad esso successivo senza tornare mai indietro. Gli strati nascosti permettono di realizzare confini decisionali complessi e non lineari. Nel caso in cui non ci fosse nessun livello nascosto, si ha un percettrone che può realizzare solo un iperpiano di separazione. Ogni nodo di uno strato nascosto può essere visto come un singolo percettrone.
Nella struttura più semplice, senza *hidden layer*, si hanno diversi nodi input x_i tutti connesi attraverso dei pesi w_i al nodo output, il quale è il risultato della funzione di attivazione applicata alla somma pesata dei vari input e di un termine di errore, definito *bias* (b). Indicando con x e w i vettori aventi per elementi rispettivamente i nodi input e i relativi pesi, nel caso dell'uso della *sign function* come funzione di attivazione, si ha che la previsione $\hat{y}$ è il risultato della seguente espressione:

$$\hat{y} = sign(w^T x + b) \tag{2.1}$$

Ne segue dunque che se l'espressione all'interno della *sign function* sarà positiva, il valore output sarà pari a 1, al contrario se sarà non positiva avremo -1 come risultato della classificazione. L'obiettivo della rete è aggiornare il valore dei pesi in modo da minimizzare l'errore di previsione espresso dalla differenza tra i valori osservati (y) e quelli predetti($\hat{y}$):

$$w_j^{l+1} = w_j^l + \lambda(y_i - \hat{y_i}^l)x_{ij} \tag{2.2}$$

Ne segue che il peso aggiornato (w_j^{l+1}), associato al j-esimo attributo, nell'iterazione successiva ($l+1$) è dato dal peso del livello precedente sommato al residuo tra il valore osservato e quello predetto nell'iterazione precedente ($y_i - \hat{y}_i^l$), moltiplicati per l'i-esimo input del j-esimo attributo (x_{ij}) e per il *learning rate*(λ). Quest'ultimo è un iper-parametro della rete neurale i cui valori sono $\lambda \in [0, 1]$, e indica quanto velocemente o lentamente la rete si aggiorna. Infatti se fosse esattamente pari al valore nullo, il "nuovo" peso sarebbe esattamente identico a quello dell'iterazione precedente e dunque non ci sarebbe un effettivo aggiornamento.

La Figura 2.1 presenta l'architettura base di un percettrone:

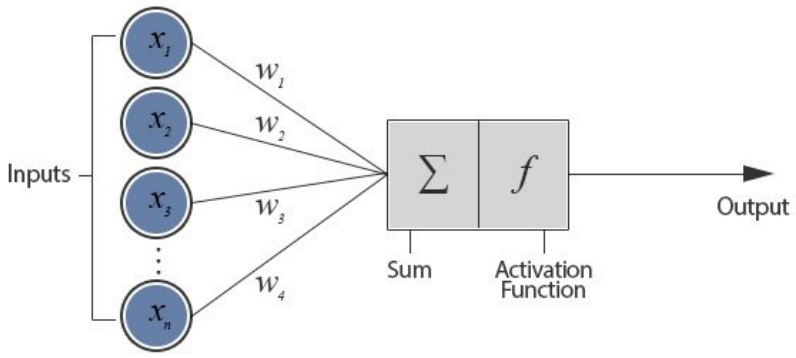

Figura 2.1: *Architettura base del percettrone* (*fonte:* *"Quora"*).

L'aggiornamento dei pesi avviene fino a quando la media dei residui è più piccola di un valore soglia, tendente al valore nullo. In generale, però, sono presenti anche i livelli nascosti, nella cui rete ogni singolo nodo di un livello è connesso con ogni nodo del livello successivo e la forza del legame è sempre espressa attraverso il peso w_{ij}. Nel caso di una rete con l = *0,1,...,L-1,L* livelli, dei quali quello contrassegnato con il valore nullo è l'*input layer* e quello contrassegnato da L è l'*output layer*, il valore di attivazione viene trasmesso e ricalcolato tra i vari livelli nascosti. In particolare, il valore di attivazione generato nell'i-esimo nodo allo strato l-esimo (a_i^l) è dato dalla funzione di attivazione applicata al relativo *linear predictor*(z_i^l), nel seguente modo:

$$a_i^l = f(z_i^l) = f(\sum_j w_{ij}^l a_j^{l-1} + b_i^l) \tag{2.3}$$

In tale espressione w_{ij}^l rappresenta il peso del legame tra l'i-esimo nodo del l-esimo livello con il nodo j-esimo del (l-*1*)-esimo livello. Inoltre, per definizione $a_j^0 = x_j$ e $a^L = \hat{y}$, così che nel primo livello nascosto (l=*1*), il valore di

attivazione è esattamente il risultato del percettrone, se la funzione di attivazione è la *sign function*. Esistono diverse funzioni di attivazione, tra quelle principali abbiamo:

Funzione di attivazione	Formula	Derivata
Funzione lineare	$f(z) = z$	$f'(z) = 1$
Funzione sigmoidea	$\sigma(z) = \frac{1}{1+e^{-z}}$	$\sigma'(z) = \sigma(z)(1-\sigma(z))$
Funzione tan-h	$tanh(z) = \frac{2}{1+e^{-2z}} - 1$	$tanh'(z) = 1 - tanh(z)^2$
Rectified Linear Unit (ReLU)	$f(z) = \begin{cases} 0 & \text{se } x < 0 \\ z & \text{se } x \geq 0 \end{cases}$	$f'(z) = \begin{cases} 0 & \text{se } x < 0 \\ 1 & \text{se } x \geq 0 \end{cases}$
Softplus	$f(z) = ln(1+e^z)$	$f'(z) = \sigma(z) = \frac{1}{1+e^{-z}}$

Tabella 2.1: Funzioni di attivazione

Nella Tabella 2.1 il valore z rappresenta il *linear predictor*. Nel caso della presenza di livelli nascosti, l'obiettivo consiste sempre nell'aggiornare i pesi e i *bias* per minimizzare una funzione di perdita, come ad esempio la *squared loss function*:

$$E(w,b) = \sum_{h=1}^{n} Loss(y_h, \hat{y_h}) = \sum_{h=1}^{n} (y_h - \hat{y_h})^2 \tag{2.4}$$

L'obiettivo è determinare i vettori w e b, rispettivamente dei pesi e dei *bias*, che minimizzino *E(w,b)*. Si ha però un problema, con l'impiego dei nodi nascosti caratterizzati da funzioni di attivazione non lineari, ne segue che *E(w,b)*

non è una funzione convessa, ma sarà alquanto complessa e caratterizzata da minimi locali, che rende difficile determinare un punto di ottimo globale. Per tale motivo è necessario impiegare il metodo di ottimizzazione denominato "*Gradient Descent*", il quale impiega l'iper-parametro *learning rate*(λ) per determinare la migliore soluzione locale. Per valori elevati di λ ne segue che potremmo individuare diversi minimi locali, con difficoltà di giungere rapidamente a convergenza, per valori bassi invece rischiamo l'effetto opposto, ovvero di rimanere nello stesso minimo locale senza trovarne altri migliori. Per risolvere tale problema spesso si utilizza una funzione decrescente per λ così da iniziare con valori elevati con la possibilità di arrivare a una zona con minimo globale, poi riducendo il valore di lambda si evita di individuare un altro minimo, che potrebbe essere locale. Di seguito è illustrata un'immagine per spiegarne il concetto:

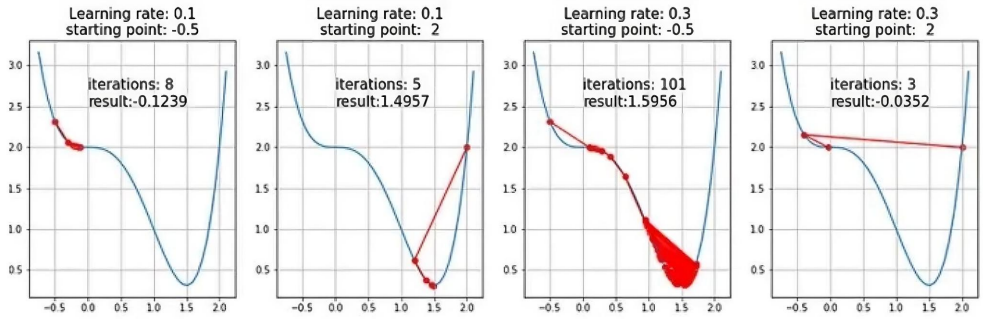

Figura 2.2: *Metodo del Gradient Descent (fonte: "Towards Data Science")*

Il valore in rosso indica il valore di λ che decresce per raggiungere il minimo locale, che potrebbe essere anche globale. Ovviamente tutto dipende dall'inizializzazione dei pesi, e dunque da quale "punto" partire. In particolare l'aggiornamento, ad esempio di un peso, avviene sottraendogli il valore della

derivata parziale della funzione di perdita rispetto al peso moltiplicato per il *learning parameter*:

$$w_{ij}^{l*} = w_{ij}^{l} - \lambda \frac{\delta E(w,b)}{\delta w_{ij}^{l}} \tag{2.5}$$

In una zona decrescente della *Loss Function* si avrebbe una derivata negativa ($\frac{\delta E(w,b)}{\delta w_{ij}^{l}} < 0$), λ è per definizione non negativo, e di conseguenza il nuovo peso aggiornato (w_{ij}^{l*}) sarà più grande rispetto al "vecchio" peso (w_{ij}^{l}). Ne consegue che un aumento del peso, in una zona decrescente della funzione di perdita, potrebbe farci individuare uno dei minimi e dunque un punto di ottimo, almeno locale (Figura 2.2). Diversamente se la derivata parziale è positiva, ci troviamo in una zona crescente della funzione e per avvicinarci al valore ottimale, il peso dovrà essere ridotto. Per il *bias* si ha la stessa procedura, solo che avremo la derivata parziale della funzione di perdita rispetto quel determinato *bias* (b_i^l).

Però c'è un ulteriore problema, nel caso di livelli nascosti viene propagato anche il valore di attivazione e ne segue che la previsione $\hat{y}$ corrisponde al valore di attivazione nello strato output (a^L). Dalla formula (2.3) si nota che la previsione dipenderà dai pesi e *bias* del livello output e dal valore di attivazione dell'ultimo livello nascosto, ma quest'ultimo dipende da altri pesi e *bias* attraverso una catena di valori di attivazione. É fondamentale impiegare la *backpropagation*, attraverso la quale l'aggiornamento dei pesi avviene in senso inverso rispetto a come le informazioni si propagano nella rete. Dunque si calcolano le derivate parziali procedendo in senso inverso, dall'output layer andando verso il livello input.

Bisogna precisare che quando vengono calcolate le derivate parziali rispetto ai pesi e ai *bias*, si calcola anche la derivata della funzione di perdita rispetto

al valore di attivazione di un determinato nodo i nel l-esimo livello:

$$\frac{\delta E(w,b)}{\delta w_{ij}^l} = \sum_{h=1}^{n} \frac{\delta Loss(y_h, \hat{y_h})}{\delta w_{ij}^l} = \sum_{h=1}^{n} \left(\frac{\delta Loss}{\delta a_i^l} \times \frac{\delta a_i^l}{\delta z_i^l} \times \frac{\delta z_i^l}{\delta w_{ij}^l} \right) \tag{2.6}$$

$$\frac{\delta z_i^l}{\delta w_{ij}^l} = \frac{\delta \sum_j (w_{ij}^l a_j^{l-1} + b_i^l)}{\delta w_{ij}^l} = a_j^{l-1} \tag{2.7}$$

$$\frac{\delta Loss}{\delta a_i^l} = \sum_r \left(\frac{\delta Loss}{\delta a_r^{l+1}} \times \frac{\delta a_r^{l+1}}{\delta a_i^l} \right) = \sum_r \left(\frac{\delta Loss}{\delta a_r^{l+1}} \times \frac{\delta a_r^{l+1}}{\delta z_r^{l+1}} \times \frac{\delta z_r^{l+1}}{\delta a_i^l} \right) \tag{2.8}$$

Nella (2.8), il termine r rappresenta un nodo del livello successivo (l+1).
Riassumendo il funzionamento di una rete neurale artificiale multistrato, innanzitutto vengono inizializzati i pesi e i termini *bias* in modo casuale, poi per ogni osservazione dei dati di allenamento vengono determinati i valori di attivazione (2.3), e attraverso la *back propagation* vengono calcolate le derivate parziali della funzione di perdita rispetto ai valori di attivazione (2.8), ai pesi e ai *bias*. Calcolate le derivate parziali delle funzioni di perdita, si calcolano poi le derivate parziali di E(w,b), sempre rispettivamente ai pesi e al *bias*. Attraverso il *Gradient Descent* si aggiornano i vari pesi e *bias* (2.5). Con i vari aggiornamenti, viene girata nuovamente la rete per ottenere nuove previsioni. La procedura si ripete fino a quando i pesi e i *bias* non si aggiornano, o meglio, fino a quando i nuovi valori sono molto simili a quelli precedenti, utilizzando un threshold.
In relazione alle serie storiche è possibile impiegare un'*Autoregressive Neural Network* (NNAR), che imita il processo autoregressivo (AR) attraverso una rete *feed-forward* (Hyndman R. et al., 2021). In particolare, il modello è ca-

ratterizzato da un solo livello nascosto e i nodi input rappresentano il numero di ritardi, o *lag*, della variabile dipendente. Per semplicità con il termine p si rappresentano i ritardi, come nel modello AR(p). Oltre al numero di nodi input, è necessario anche determinare il numero s di nodi dell'unico livello nascosto.

Ne segue che un modello NNAR(p, s) è una rete caratterizzata da p nodi input, e dunque ritardi della variabile dipendente $(y_{t-1}, y_{t-2}, ..., y_{t-p})$, e da s neuroni nell'*hidden layer*.

Diversamente dai modelli classici, nei quali devono sussistere alcune ipotesi quali quella di stazionarietà delle serie storiche, nel modello NNAR non vi sono tali vincoli, anzi ci sono diversi iper-parametri per gestire anche la stagionalità, come P per definire il numero di ritardi stagionali ("*seasonal lags*"). Se i valori di p e P non sono specificati, per serie non stagionali, in `default` verrà attribuito a p il valore ottimale di ritardi per un processo AR(p) considerando il criterio informativo "*AIC*" e P avrà un valore nullo. Per serie stagionali, P sarà pari a 1 e p sarà definito in base al miglior modello, in relazione all'*AIC*, adattato ai dati destagionalizzati. Nel caso in cui non venga stabilito il numero di nodi nascosti, allora per `default` sarà attribuito un valore pari a $s = \dfrac{(P + p + 1)}{2}$, approssimato a `integer`.

Uno dei problemi principali delle reti neurali è l'inizializzazione dei pesi, per la NNAR è possibile anche definire l'iper-parametro "*repeats*", per specificare il numero di reti da realizzare aventi gli stessi parametri (p, P, s), ma con diverse inizializzazioni casuali di pesi. Di questi, viene calcolata la media per inizializzare i pesi del modello di rete finale, utilizzato per determinare le previsioni.

Nella fase di predizione, la rete viene applicata in modo iterativo. Ad esempio, se i dati sono raccolti giornalmente e si volesse prevedere il valore del giorno successivo, allora saranno impiegati per l'analisi gli input storici disponibili. Nel caso in cui l'orizzonte temporale di previsione fosse più ampio, allora per prevedere il secondo giorno, sarà impiegata la previsione per il giorno successivo e i dati storici come input, e così via.
La rete NNAR permette anche di considerare dei regressori per la previsione, attraverso un ulteriore parametro opzionale: `xreg`. Bisogna inserire un dataset con regressori, solo numerici, avente lo stesso numero di righe della serie storica y_t.
Ovviamente uno dei problemi principali di questo modello è l'interpretabilità dei risultati e del contributo dato da ogni regressore nella pevisione finale. Il modello è, inoltre, limitato dall'utilizzo di un solo livello nascosto. Nel prossimo paragrafo tratteremo, invece, dell'impiego di un'altra tipologia di rete neurale diversa dalla *feed-forward*: reti neurali ricorrenti.

2.2 Reti neurali in finanza

La previsione dei dati finanziari è una delle sfide più impegnative nella previsione delle serie temporali a causa dell'influenza di fattori sociali, politici ed economici che contribuiscono a definire il comportamento del mercato azionario. I dati finanziari, quali i prezzi delle azioni, sono caratterizzati da un'alta frequenza, divenendo più soggetti a problemi di eteroschedasticità condizionale. Attraverso l'impiego dei modelli *ARCH*, *GARCH*, ecc., è possibile svolgere un'analisi sulla volatilità e realizzare delle previsioni più

accurate.

Se volessimo, però, utilizzare un modello di rete neurale per prevedere i prezzi delle azioni, allora si necessita dell'impiego di una rete che sia flessibile a diverse quantità di dati sequenziali.

In una rete neurale ricorrente (RNN) i neuroni sono collegati tra loro attraverso un ciclo *feedback* in modo tale da utilizzare i valori di input in modo sequenziale. Per comprenderne il funzionamento, di seguito viene mostrata l'architettura di una rete neurale ricorrente:

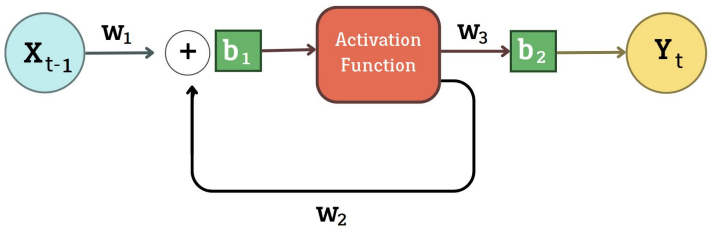

Figura 2.3: *Architettura della rete neurale ricorrente* (*fonte: StatQuest*)

Similmente alla rete neurale *feed-forward*, anche una generica *RNN* è caratterizzata da pesi, *bias* e una funzione di attivazione, la novità consiste nel ciclo *feedback* che permette l'impiego sequenziale degli input. Dalla Figura 2.3 sembra che una *RNN* richieda un solo valore input (x_{t-1}), ma attraverso il ciclo, è possibile impiegare input in modo sequenziale. Generalmente i valori vengono prima ridimensionati in un range $[0, 1]$, in modo tale da associare a valori nulli, le previsioni di prezzo "*basse*" e a valori unitari quelle alte. Una volta ridimensionati i dati, viene immesso il primo input, viene determinato il *linear predictor* ($z_t = x_{t-1}w_1 + b_1$) dal quale si ottiene il valore di attiva-

zione in base alla funzione scelta ($a_t = f(z_t)$). Tale valore viene impiegato in due percorsi:

- Il valore di attivazione viene impiegato per prevedere il valore al tempo t e ottenere così l'output y_t. Il problema è che non si è interessati alla previsione di oggi, perchè si dispone già del suo valore effettivo, l'obiettivo è determinare la previsione per il tempo *t+1*. Al momento, dunque, non consideriamo l'output y_t.

- Si utilizza il valore di attivazione nel ciclo *feedback*, che ci permette di aggiungere al *linear predictor* il valore di attivazione moltiplicato per il peso, basato sul valore di ieri, per determinare il nuovo *linear predictor* associato a *t+1*:

$$z_{t+1} = (x_{t-1}w_1 + b_1) + a_t w_2 = z_t + a_t w_2$$

Ne segue che il ciclo consente, in questo esempio, sia ai valori di ieri che a quelli di oggi di influire nella previsione per domani, attraverso il calcolo del nuovo valore di attivazione per *t+1*, rimoltiplicandolo poi per il relativo peso e aggiungendo il termine *bias*:

$$y_{t+1} = f(z_{t+1})w_3 + b_2 = a_{t+1}w_3 + b_2$$

Per chiarire il funzionamento, è possibile "*srotolare*" il ciclo *feedback*:

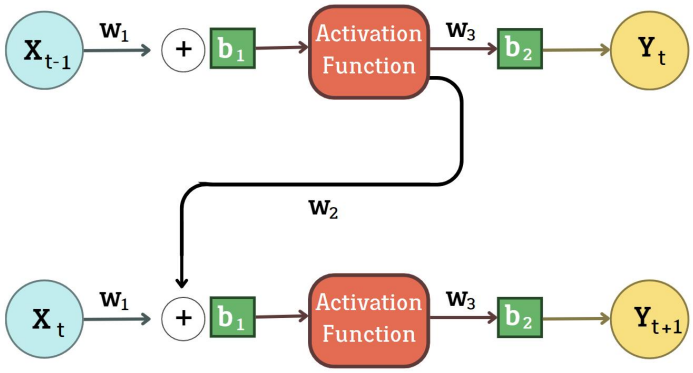

Figura 2.4: *Unrolled RNN* (*fonte: StatQuest*)

La Figura 2.4 mostra come il valore di attivazione determinato con l'input associato al tempo *t-1*, viene impiegato per poter poi prevedere il valore per l'istante *t+1*. Il modello, ovviamente, diviene più complesso in base all'ampiezza dell'orizzonte temporale di analisi e addestramento per la rete.
Bisogna precisare che indipendentemente dall'orizzonte temporale, vengono condivisi sempre gli stessi pesi e *bias*. Tale fenomeno genera però un problema (Alex Sherstinsky, 2020): il *"Vanishing/Exploding Gradient"*. L'addestramento della rete avviene attraverso la *backpropagation through time* (*BBTT*), che diversamente dal procedimento delle reti neurali classiche, sfrutta il fatto che per ogni rete ricorrente esiste una serie di reti *feed-forward*. É come se ci fossero una catena di reti aventi ognuna un comportamento identico alle altre (caratterizzate dagli stessi pesi e *bias*), solo che ognuna è associata a una sequenza temporale diversa. L'aggiornamento avviene, come nelle reti classiche, con lo scopo di ottimizzare una funzione obiettivo (minimizzare E(w,b)), impiegando anche il *Gradient Descent*. Il problema è che per ogni

passaggio temporale, si necessita di sommare tutte le contribuzioni di tutti gli istanti precedenti fino a quello corrente.

Consideriamo, ad esempio, solo w_2 nella Figura 2.4 e ipotizziamo di avere una serie storica con un orizzonte temporale di sessanta giorni. Ne segue che sarà necessario "*srotolatare*" la rete ricorrente per sessanta volte (da x_{t-60} a x_t). Da tenere presente che ogni volta, nella *backpropagation*, dobbiamo calcolare le derivate parziali per i vari parametri (pesi, *bias* e valori di attivazione), come è stato mostrato nelle formule (2.6), (2.7), (2.8).

Se $w_2 < 1$, il neurone di input verrà moltiplicato per w_2 per sessanta volte, per poter arrivare nell'ultimo strato dove andremo a calcolare la previsione per il giorno successivo. Ne segue che l'input verrà moltiplicato per un numero molto piccolo, tendende al valore nullo (*vanishing problem*). Nella fase di addestramento di una rete di questo tipo, nel calcolo delle derivate è presente questo valore estremamente piccolo, avendo come effetto un aggiornamento molto piccolo dei vari parametri, così da raggiungere molto lentamente il valore ottimale (Figura 2.2). Se $w_2 > 1$ si avrà un effetto opposto, ovvero che l'input sarà moltiplicato per un valore molto elevato (*exploding problem*). In questo caso, per l'addestramento, le derivate saranno caratterizzate da valori molto elevati e così nelle varie fasi di aggiornamento i parametri si sposteranno di continuo, senza raggiungere rapidamente il valore ottimale (Figura 2.2).

Per risolvere tale problematica, sono state progettate due reti neurali ricorrenti, ampiamente utilizzate nella pratica per la previsione di dati finanziari: *"Long Short Term Memory"* (LSTM) e *"Gated Recurrent Unit"* (GRU).

In primo luogo, viene presentato il funzionamento della rete *LSTM*. Nella seguente Figura 2.5 è possibile osservare l'architettura di una singola unità *LSTM*:

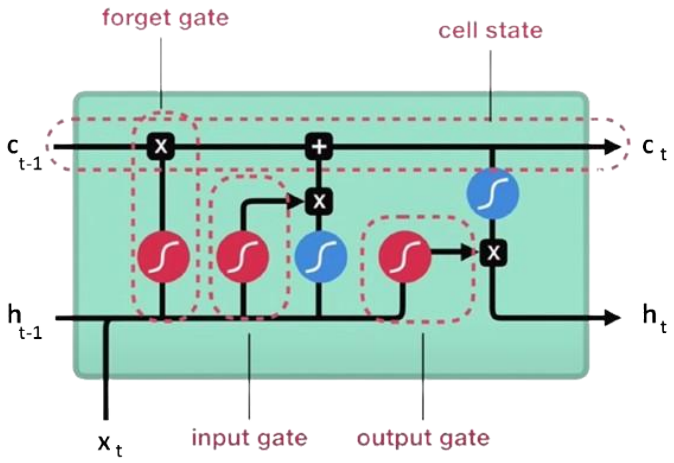

Figura 2.5: *Architettura di una singola unità LSTM* (*fonte: Towards Data Science*)

In una singola unità sono presenti due flussi (rappresentati nella Figura 2.5 dalle due frecce orizzontali che entrano ed escono dall'unità *LSTM*): la *cell state* (c_t), o *Long Term Memory* (*LTM*), e l'*hidden state* (h_t), noto anche come *Short Term Memory* (*STM*). Nella *LSTM* sono impiegate due funzioni di attivazione: la *sigmoid*, in rosso, e la *tanh*, in blu (*cfr. Tabella 2.1*).
La *LTM* non è caratterizzata da pesi e *bias*, impedendo così al gradiente di esplodere o svanire, mentre sono presenti nella *STM*. In particolare, un'unità *LSTM* è caratterizzata da tre fasi:

1. *Forget Gate*: In questa sezione viene determinata la percentuale di informazione da mantenere, o ricordare. Nel dettaglio, osservando la Figura 2.5 dal basso verso l'alto, si ha che l'informazione passata dal precedente *hidden state* (h_{t-1}) e quella dell'attuale input (x_t) influiscono, insieme ai rispettivi pesi, nella *sigmoid function* per determinare il risultato del *forget gate*, che denominiamo f_t:

$$f_t = \sigma(h_{t-1}w_1 + x_t w_2 + b_1)$$

Si ricorda che il risultato della funzione sigmoidea $f_t \in [0, 1]$, e nel caso in cui il valore tende a zero, si ha che quell'informazione è trascurabile. Per tale motivo, la prima fase dell'unità *LSTM* viene denominata *"forget gate"*. Il risultato (f_t) viene poi moltiplicato al precedente *cell state* (c_{t-1}).

Assumiamo che, per esempio, il precedente *hidden state* e l'attuale input sono entrambi pari a 1, e i rispettivi pesi nella fase di *forget gate* siano 2.60 e 1.80, mentre il *bias* sia 1.60, ne segue che $f_t = \sigma(1 \times 2.60 + 1 \times 1.80 + 1.60) = \sigma(6.00) = 0.99$. Di conseguenza il valore del *cell state* cambierà poco in questa fase, ad esempio $c_{t-1} = 3.00$ allora avremo $3.00 \times 0.99 = 2.97$, andando così a ricordare una buona percentuale della *LTM*. Nel caso cui invece f_t fosse vicino al valore nullo, ad esempio 0.01, si avrebbe come effetto che è necessario ricordare solo l'1% della *LTM* ($3.00 \times 0.01 = 0.03$).

2. *Input Gate*: Questa fase è più delicata perchè vengono impiegate due funzioni di attivazione, la sigmoidea per calcolare la percentuale di memoria potenziale da ricordare, la *tanh* per il potenziale della *LTM*.

Si analizza in primo luogo il potenziale della *LTM*, rappresentato come $\widetilde{c}_t$, il quale utilizza sempre il precedente *hidden state* e l'attuale input, però con diversi pesi e bias:

$$\widetilde{c}_t = tanh(h_{t-1}w_3 + x_t w_4 + b_2)$$

Poichè è risultato di una *tanh function*, si ha che $\widetilde{c}_t \in [-1, 1]$ e serve per regolare la rete e determinare il potenziale di memoria.
In secondo luogo, si ha la percentuale di memoria potenziale da ricordare che denominiamo con il termine i_t, anche qui influisce il precedente *hidden state* e l'attuale input, con i rispettivi pesi e bias:

$$i_t = \sigma(h_{t-1}w_5 + x_t w_6 + b_3)$$

Ne segue dunque che $i_t \in [0, 1]$ e serve per individuare quanta percentuale della memoria potenziale ricordare.
In terzo luogo si determina quale informazione è importante da mantenere e ricordare moltiplicando $\widetilde{c}_t$ con i_t. La fase di *input gate* termina con l'aggiornamento della *cell state* combinando il risultato del*forget gate* con il prodotto tra $\widetilde{c}_t$ e i_t:

$$c_t = f_t \times c_{t-1} + \widetilde{c}_t \times i_t$$

É stato aggiornato il *cell state* che sarà uno dei due risultati che escono da un'unità *LSTM*, infatti nella prossima unità verrà preso c_t e verrà aggiornato in c_{t+1} e così via.

3. *Output Gate*: Nella terza e ultima fase viene determinato l'altro risultato che fuoriesce da un'unità *LSTM* (h_t, *cfr. Figura 2.5*) e consiste

nell'aggiornamento della memoria a breve termine (STM). In generale, l'*hidden state* contiene tutte le informazioni dei precedenti input che poi usiamo per la previsione. Dalla *cell state* aggiornata, determinata nella fase precedente (c_t), viene determinata la memoria a breve termine potenziale attraverso la *tanh function*:

$$\nu_t = tanh(c_t)$$

Attraverso la funzione sigmoidea viene calcolata la percentuale di memoria da ricordare (o_t), prendendo come argomenti il precedente *hidden state* e l'attuale input, con i rispettivi pesi:

$$o_t = \sigma(h_{t-1}w_7 + x_t w_8 + b_4)$$

Il nuovo *hidden state* viene calcolato moltiplicando i due risultati:

$$h_t = \nu_t \times o_t$$

Poichè in tale fase viene determinato il secondo risultato di una singola unnità $LSTM$, viene denominato *"output gate"*.

Si ha dunque che per ogni unità *LSTM* vengono determinati la *cell state* e l'*hidden state*, di conseguenza si osservi come funziona l'intero modello *LSTM*:

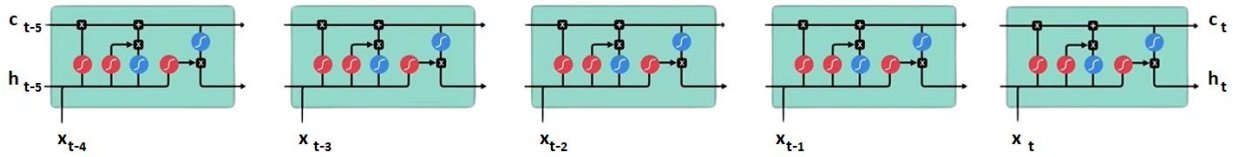

Figura 2.6: *Processo LSTM completo* (*fonte:Towards Data Science*)

Ogni unità serve per determinare quanta e quale informazione ricordare, o dimenticare, a ogni step temporale. Si avrà un'unità *LSTM* per ogni momento dell'orizzonte temporale. Se volessimo prevedere il giorno successivo e avessi un set informativo di cinquanta giorni per allenare il modello, avremmo cinquanta unità *LSTM*. All'inizio del processo, per $t = 0$, la *cell* e *hidden state* verranno inizializzate con valori casuali.
Un'altra alternativa è il *"Gated Recurrent Unit* (GRU), il quale, diversamente dalla *LSTM*, non è caratterizzato dal *cell state*, ma trasferisce le informazioni solo attraverso l'*hidden state*.

Similmente a una rete ricorrente, anche il GRU è caratterizzato da una serie di unità per ogni istante temporale:

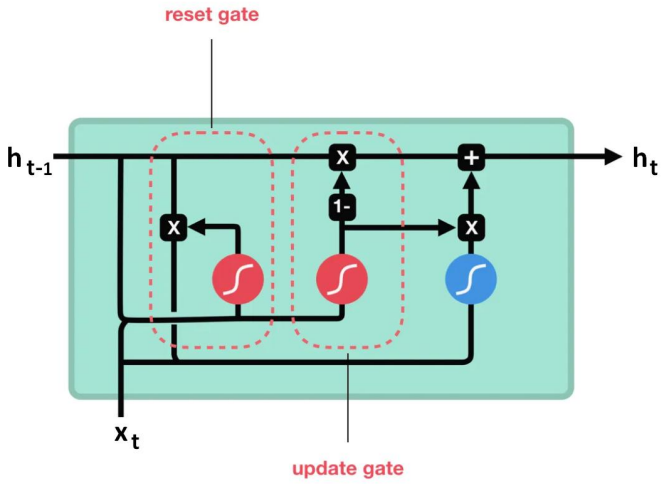

Figura 2.7: *Singola unità GRU* (*fonte: Towards Data Science*)

Sono presenti solamente due fasi:

1. *Reset Gate*: Determina quanta informazione passata deve essere ricordata. Nel dettaglio, viene calcolato il *reset value* per quel determinato step temporale (r_t) attraverso la funzione sigmoidea (*cfr. Tabella 2.1*):

$$r_t = \sigma(h_{t-1}w_1 + x_t w_2 + b_1)$$

   Successivamente, r_t viene moltiplicato con il precedente *hidden state* (h_{t-1}) per determinare quanta informazione passata debba essere mantenuta per quel particolare step temporale t.

2. *Update Gate*: Determina quali nuove informazioni debbano essere aggiunte.

In primo luogo si determina l'*hidden state* candidato ($\widetilde{h}_t$) attraverso la *tanh function*, la quale però avrà come argomenti l'attuale input e il prodotto tra il *reset* e il precedente *hidden state*:

$$\widetilde{h}_t = tanh((h_{t-1} \times r_t)w_3 + x_t w_4 + b_2)$$

Ne segue che, per determinare il candidato, viene considerata non tutta l'informazione del precedente istante temporale (*t-1*), ma solo ciò che deve essere ricordato, espresso attraverso "$h_{t-1} \times r_t$".
In secondo luogo, devono essere calcolate le informazioni da trasferire dal precedente *hidden state* attraverso nuovi pesi e bias:

$$z_t = \sigma(h_{t-1}w_5 + x_t w_6 + b_3)$$

Poiché $z_t \in [0, 1]$ si determina la percentuale di informazioni da mantenere. É necessario determinare, anche, le informazioni da trasferire dall'*hidden state* candidato.
Ne segue che dal corrente *hidden state* candidato manteniamo una percentuale pari a z_t, e per il resante 1-z_t manteniamo le informazioni del precedente *hidden state*. Ne segue che l'aggiornamento sarà determinato come:

$$h_t = (1 - z_t)h_{t-1} + z_t\widetilde{h}_t$$

Analizzato il funzionamento di una singola unità *GRU*, l'intero processo è caratterizzato da una struttura identica a quella illustrata nella Figura 2.6 per la *LSTM*, con la differenza che a ogni istante temporale viene solo aggiornato l'*hidden state*, fondamentale per la fase previsionale. Diversamente dalla

LSTM, il *GRU* impiega meno operazioni e risulta, così, più rapido nella fase previsionale. Nei casi pratici di analisi, vengono entrambe impiegate e confrontate per analizzare quale *RNN* sia più idonea per quello specifico fenomeno. Nel prossimo paragrafo verranno presentate varie modalità per selezionare il miglior modello e le relative problematiche.

2.3 Problemi di selezione dei modelli

Nel caso di modelli classici, quali gli *ARIMA*, per la selezione del miglior modello esistono diverse tecniche. In una prima analisi, attraverso dei test di specificazione, si determina quale modello non sia caratterizzato da problemi di autocorrelazione nei residui, e dunque un problema di mispecificazione in media. In seguito è possibile attuare anche dei test per verificare la presenza di problemi di eteroschedasticità condizionale, gestibile attraverso errori standard robusti oppure l'impiego del modello *GARCH*, e test sulla normalità dei residui. Una volta selezionati una serie di modelli, che non siano almeno mispecificati in media, vengono confrontati attraverso diverse metriche quali i criteri informativi (*C.I.*), l'errore quadratico medio (*RMSE*) e anche il *R-squared*.

Il problema (Giudici et al., 2020) è che tale sistema non è applicabile per modelli non probabilistici di *machine learning*, eccetto un confronto tramite *RMSE*. Quest'ultima metrica, però, soffre di un problema di dipendenza dell'unità di misura della variabile dipendente. In generale più l'*RMSE* tende al valore nullo, migliore è la predittività del modello. Assumiamo però di confrontare un modello *OLS* basato sui prezzi e un modello *LSTM* basato sui

dati scalati. Non sarà possibile confrontare i due modelli sul *RMSE*, poichè un modello basato sui prezzi potrebbe avere la metrica vicina ad esempio a 200, mentre per la *LSTM* pari a 0.60. Applicando la metrica sembrerebbe migliore la *LSTM* perchè più tendente al valore nullo, ma se i dati di addestramento della rete sono stati scalati in un intervallo [0,1], allora per costruzione l'*RMSE* non potrà avere valori molto distanti da quello nullo. In questi casi, infatti, le previsioni di una rete ricorrente vengono poi riportate nella loro forma originale e viene ricalcolato l'*RMSE*, confrontabile così con quello del modello *OLS*. Di conseguenza l'*RMSE* risulta non essere una metrica normalizzata, diversamente dai *C.I.* o dal *R-squared* utilizzabili però solo in alcuni modelli. Inoltre all'aumentare del numero di regressori, non necessariamente l'*RMSE* decresce.

É necessario definire una nuova metrica normalizzata applicabile anche in contesti di *machine learning*, quale quella definita da Giudici e Raffinetti (2020) sull'impiego dello zonoide di Lorenz.

In primo luogo è fondamentale chiarire la curva di Lorenz (1905), adottata in economia principalmente per analizzare la disuguaglianza di reddito tra vari paesi. Date n osservazioni di una variabile casuale Y, la curva di Lorenz (L_Y) viene determinata:

1. Riordinando i valori di Y in ordine non decrescente (y_i, con i = *1,...,n*).
2. Indicando con $\overline{y}$ il valore atteso di Y, si calcolano tutte le probabilità cumulate per definire l'insieme di punti $\left(i/n, \sum_{j=1}^{i} \frac{y_j}{n\overline{y}} \right)$.

In ambito economico, si indica con una bisettrice il caso di perfetta uguaglianza di reddito, mentre con la curva di Lorenz lo stato di gestione del

reddito di un determinato paese. Ne segue che più la curva si discosta dalla retta bisettrice, e dunque maggiore è l'area compresa tra le due curve, maggiore è il grado di disuguaglianza di reddito. Quest'area rappresenta il c.d. "*coefficiente di Gini*", il cui valore è $x \in [0, 1]$, perchè rappresenta la percentuale di disuguaglianza. Di conseguenza, se $x \to 0$, allora in quel paese ci sarà una ugual distribuzione di reddito tra la popolazione.
Oltre a L_Y, viene anche definita (Giudici et al., 2020) la *"dual Lorenz curve"* (L'_Y) determinabile con la stessa procedura, solo che i valori di Y sono ordinati in modo non crescente, ottenendo così una curva concava situata al di sopra della retta bisettrice.
Generalizzando la curva di Lorenz in uno spazio d-dimensionale, con l'aggiunta di altre variabili, si introduce il concetto di zonoide di Lorenz. Dunque se $d = 1$, lo zonoide andrà a coincidere con la curva di Lorenz, e di conseguenza con il coefficiente di *Gini*. Se si considera una variabile casuale Y, il relativo zonoide di Lorenz sarà:

$$LZ_{d=1}(Y) = \frac{2Cov(Y, r(Y))}{nE(Y)} \tag{2.9}$$

Il termine $r(Y)$, denominato *"rank score"*, rappresenta una variabile numerica costituita da interi con range $\in [1, n]$, dove n è il numero di osservazioni, che associa il valore unitario al minimo di Y e n al massimo. Dopo aver riordinato Y in modo non decrescente, il relativo *rank score* è una serie di interi da 1 a n e può essere definito, per costruzione, un nuovo termine:

$$q = \frac{1}{\overline{y}} \left[\frac{1}{n} \sum_{i=1}^{n} i y_i - \frac{n+1}{2} \overline{y} \right]. \tag{2.10}$$

Di conseguenza è dimostrabile (Giudici et al., 2020) che:

$$LZ_{d=1}(Y) = \frac{2q}{n}. \tag{2.11}$$

Definito lo zonoide di Lorenz, è necessario esaminare le sue proprietà:

- *Dipedenza lineare*: Considerate due variabili casuali X e Y, quest'ultima sarà linearmente dipendente ad X, se $LZ(X) \subset LZ(Y)$.

- *Proprietà di inclusione*: Se $LZ(X) \subset LZ(Y)$, allora lo zonoide relativo alla distribuzione di Y domina lo zonoide di X.
  Graficamente è come se la curva di Lorenz associata a Y fosse più lontana alla retta bisettrice, rispetto alla curva relativa a X. Di conseguenza la curva di X è situata tra la bisettrice e la curva di Y, e per tale motivo la proprietà viene definita *"di inclusione"*.

Nel caso univariato (*d=1*) in merito alla seconda proprietà, è valido un corollario per il quale la dominanza dello zonoide rispecchia l'ordinamento basato sulla variabilità, e viceversa:

$$Var(Y) > Var(X) \iff LZ(Y) \textit{ domina } LZ(X)$$

Se X fosse un unico regressore, Y la variabile dipendente e $\hat{Y}$ le previsioni del modello lineare *OLS*, allora la rappresentazione delle curve di Lorenz, e della *dual Lorenz*, relative a Y e $\hat{Y}$ rispetterebbero la relazione tra la varianza della variabile dipendente e quella spiegata dal modello ($Var(\hat{Y}) \leq Var(Y)$):

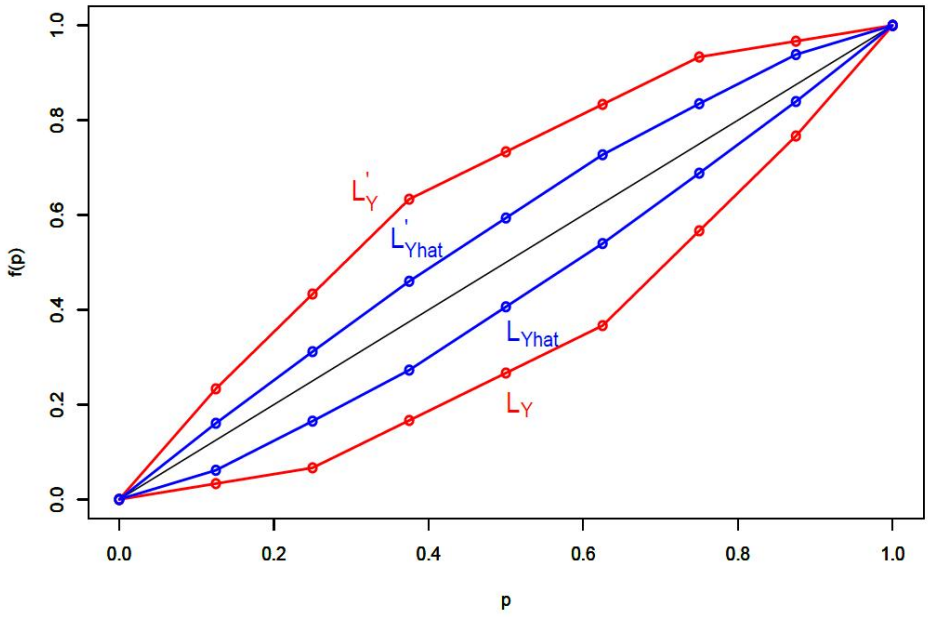

Figura 2.8: *Proprietà di inclusione dello zonoide di Lorenz* (*fonte: Lorenz model selection, Giudici et al., 2020*)

Nella Figura 2.8 sono rappresentate in rosso le curve di Lorenz e della dual Lorenz relative alla variabile dipendente, in blu quelle relative alla previsione del modello lineare impiegando il regressore X. É osservabile come le curve blu siano *incluse* all'interno dell'area, o dello zonoide, definita tra le due curve rosse.

Nella pratica si ha spesso una serie di regressori che possono migliorare o peggiorare la predizione di un determinato modello. É così possibile impiegare lo zonoide anche in relazione a diversi regressori $(X_1, X_2, ..., X_{k-1}, X_k)$ come una nuova metrica per la selezione dei modelli.

In particolare, è possibile ricavare due metrice (Giudici et al., 2020):

$$MGC_{(Y|X_h)} = \frac{LZ_{d=1}(\hat{Y}_{X_h})}{LZ_{d=1}(Y)} = \frac{Cov(\hat{Y}_{X_h}, r(\hat{Y}_{X_h}))}{Cov(Y, r(Y))} \tag{2.12}$$

$$PGC_{Y,X_{k+1}|X_1,...,X_k} = \frac{LZ_{d=1}(\hat{Y}_{X_1,...,X_{k+1}}) - LZ_{d=1}(\hat{Y}_{X_1,...,X_k})}{LZ_{d=1}(Y) - LZ(\hat{Y}_{X_1,...,X_k})} \tag{2.13}$$

Nella (2.12) è mostrata la contribuzione marginale di Gini (MGC) per una determinata variabile X_h con $h = 1, ..., k$. Tale metrica è utilizzabile per selezionare solo quei regressori che hanno il maggior contributo nella spiegazione della variabile dipendente. Nel dettaglio con $\hat{Y}_{X_h}$, si rappresenta la previsione di un determinato modello lineare con l'impiego del solo h-esimo regressore. Dunque si realizzano k modelli, ognuno dei quali è caratterizzato da uno solo dei k regressori, e poi vengono selezionate solo quelle variabili esplicative che sono caratterizzate da un elevato MGC.

Per analizzare, invece, l'effetto dato dall'introduzione di una nuova variabile X_{k+1} nel modello, si necessita di calcolare il contributo parziale di Gini (2.13). Il PGC è il rapporto tra il contributo dato dalla *k+1*-esima variabile e la percentuale dello zonoide relativo alla variabile dipendente non spiegata dallo zonoide della previsione del modello, con solo i k regressori ($\hat{Y}_{X_1,...,X_k}$). Quest'ultima metrica è sempre non negativa ed è caratterizzata da un range $\in [0, 1]$. Ne segue che se il $PGC \to 0$ allora quella *k+1*-esima variabile non andrà a dilatare di molto lo zonoide dato dal modello con i k regressori, e dunque non genererà un contributo significativo al modello.

Tali proprietà, però, valgono solo per variabili non negative a causa della costruzione dello zonoide di Lorenz (Giudici et al., 2020). Nel caso dell'uso di una rete neurale ricorrente, possiamo scalare i dati in un range $\in [0, 1]$,

realizzare i modelli con le varie combinazioni di regressori e utilizzare anche la tecnica con gli zonoidi di Lorenz.

Nel prossimo capitolo tratteremo dell'impiego di un sistema per l'interpretabilità dei modelli di *machine learning* e l'impiego degli zonoidi di Lorenz per la costruzione di ulteriori metriche.

Interpretabilità dei modelli di machine learning

Dopo aver compreso il funzionamento delle principali reti neurali impiegate in ambito finanziario, è fondamentale trattare di uno dei principali sistemi utilizzati per l'interpretabilità dei modelli di *machine learning*: i c.d. *Shapley-Value*.

3.1 Shapley Value

Nella teoria dei giochi sono stati introdotti gli *Shapley-Value* (SV) con l'obiettivo di dividere, in modo equo, il valore di un gioco tra i vari partecipanti. Si pensi, ad esempio, ad una competizione nella quale è possibile partecipare anche in gruppo e vincere dei premi per i primi tre classificati. Si assuma che il primo classificato vinca dieci mila dollari, il secondo sette mila e il terzo cinque mila e che se i due giocatori collaborano vincono il primo premio, se invece non collaborano, uno, giocatore A, vince il secondo premio

e l'altro, B, il terzo premio. Ne segue che ci sono tre possibili situazioni in base al gruppo, o coalizione, formata. Il premio che si vince rappresenta il valore della coalizione, quindi $val(C_{AB}) = 10.000\$$ poiché, nel caso in cui i due partecipanti collaborassero, riuscirebbero a vincere il primo premio. Di conseguenza $val(C_A) = 7.000\$$ e $val(C_B) = 5.000\$$. Ovviamente se nessuno partecipa non si vince alcun premio, di conseguenza è utile definire anche il $val(C_0) = 0$ per costruzione.

Il prossimo passo consiste nel definire i contributi marginali per ogni partecipante. Focalizziamoci sul concorrente A e definiamo con MC_A l'insieme di tutti i contributi marginali ad esso associati. In questo caso per A abbiamo solamente due contributi marginali, uno dato dal contributo di A rispetto alla coalizione di gruppo AB, ovvero $MC_{A_1} = val(C_{AB}) - val(C_B) = (10.000 - 5.000)\$ = 5.000\$$, ciò significa che il contributo di A alla colazione genererebbe un aumento del guadagno complessivo di 5.000\$. L'altro contributo marginale è il caso in cui A decida di partecipare, ovvero $MC_{A_2} = val(C_A) - val(C_0) = (7.000 - 0)\$ = 7.000\$$, ne segue che $MC_A = \{MC_{A_1}, MC_{A_2}\}$. Poiché lo *Shapley-Value* (SV) associato a un giocatore è la media pesata dei suoi contributi marginali, è necessario determinare i pesi da associare agli elementi di MC_A. Nel caso di soli due giocatori, ne segue che il peso da associare a $MC_{A_1} = \frac{1!1!}{2!} = 0.5$, mentre per $MC_{A_2} = \frac{1}{2} = 0.5$. Poiché $P(MC_A) = \{0.5, 0.5\}$, di conseguenza lo $SV(A) = 0.5 \times MC_{A_1} + 0.5 \times MC_{A_2} = 6.000\$$.

Applicando la stessa procedura per il giocatore B avremmo: $MC_B = \{3.000\$, 5.000\$\}$, $P(MC_B) = \{0.5, 0.5\}$, e dunque $SV(B) = (0.5 \times 3.000 + 0.5 \times 5.000)\$ = 4.000\$$. Da notare che se sommiamo i due *Shapley-Value* associati

al giocatore A e B, otteniamo il premio complessivo nel caso in cui i due partecipanti collaborassero:

$$SV(A) + SV(B) = (6.000 + 4.000)\$ = 10.000\$ = C_{AB}$$

All'aumentare del numero di giocatori, diviene più complesso calcolare i pesi e i contributi marginali. Ad esempio con tre giocatori A,B,C c'è anche la possibilità che collaborino (C_{ABC}). Considerando anche l'ordine si avranno in totale $3! = 6$ possibili combinazioni con tre partecipanti. Se calcoliamo i pesi relativi ai contributi marginali del giocatore A, si ha in primo luogo che è necessario calcolare $P(C_{ABC} - C_{BC})$. Per tale obiettivo si conti in quanti casi tale giocatore entra a fare parte del gruppo con B e C, e dunque quante tra le sei possibili triplette, il partecipante A è alla fine (ovvero le combinazioni {B,C,A} e {C,B,A}). Poiché ci sono due casi, su un totale di sei triplette, ne segue che $P(C_{ABC} - C_{BC}) = \frac{2!1!}{3!} = \frac{1}{3}$, che rappresenta la probabilità del giocatore A di dare un contributo marginale al gruppo {B,C}. Dopo aver esaminato il caso delle triplette, si necessita di considerare le varie coppie che il giocatore A può andare a costituire. Nel caso in esempio si hanno le coppie {B,A} e {C,A} su un totale di sempre $3! = 6$ coppie:

$$P(C_{AB} - C_B) = \frac{1!}{3!} = \frac{1}{6}$$
$$P(C_{AC} - C_C) = \frac{1!}{3!} = \frac{1}{6}$$

Infine il peso da associare al caso cui A non partecipi: $P(C_A - C_0) = \frac{1}{3}$. Come precedentemente, si calcola la media pesata dei contributi marginali per determinare lo *Shapley-Value* relativo al giocatore A.

É necessario generalizzare la procedura nel caso di p giocatori, attraverso la seguente formula (*A Data Odyssey, 2023*):

$$\phi_i = \sum_{S \subseteq \{1,...,p\}\{i\}} \frac{|S|!(p-|S|-1)!}{p!}[val(S \cup \{i\}) - val(S)] \tag{3.1}$$

Nella formula (3.1) ϕ_i rappresenta lo *Shapley-Value* associato all'i-esimo giocatore, con $i = 1, ..., p$, e si indica una generica coalizione o gruppo attraverso il simbolo S. Invece, $|S|$ corrisponde al numero di giocatori all'interno di un determinato gruppo, ad esempio se S indicasse il gruppo con giocatori A,B,C allora $|S| = 3$, poiché è costituito da tre giocatori. Per la definizione dei pesi è necessario considerare $(p-|S|-1)!$, ovvero il numero di combinazioni nelle quali i giocatori possono partecipare dopo che l'i-esimo giocatore è entrato nella coalizione S. Nella sezione a destra della formula (3.1) si ha invece il contributo marginale definito come la differenza tra i valori della coalizione formata con l'inclusione dell'i-esimo giocatore $(S \cup \{i\})$ e quello dello stesso gruppo senza l'i-esimo giocatore. Come analizzato precedentemente, lo *Shapley-Value* dell'i-esimo giocatore è dato dal valore atteso pesato, con pesi equi per tutte le possibili combinazioni, di tutti i suoi contributi marginali. Il termine alla base della sommatoria $(S \subseteq \{1, ..., p\}\{i\})$ rappresenta tutti i possibili sottoinsiemi senza l'i-esimo giocatore, che viene poi incluso per calcolare i vari contributi marginali.

Gli *Shapley-Value*, nella teoria dei giochi, sono caratterizzati da quattro importanti proprietà(*A Data Odyssey, 2023*):

1. *Efficienza*: vengono considerati tutti i valori di tutte le possibili coalizioni, infatti se vengono sommati tutti gli *Shapley-Value*, si otterrà il valore della coalizione con tutti i giocatori. Nell'esempio precedente con i giocatori A e B abbiamo dimostrato come $SV(A)+SV(B) = C_{AB}$

2. *Simmetria*: due giocatori saranno intercambiabili se daranno gli stessi contributi in tutte le coalizioni. In tal caso ai due giocatori si dovrà assegnare la stessa quota del valore totale del gioco.

3. *Null Player*: Se un giocatore non genera alcun contributo in nessuna coalizione, allora non riceverà nulla del valore totale, e dunque avrà $\phi_i = 0$.

4. *Additività*: Se si combinano i risultati di due giochi, allora il contributo totale di un giocatore sarà pari alla somma dei contributi di tutti i giochi presi individualmente.

Tali proprietà sono fondamentali, poiché rendono lo *Shapley-Value* un modo equo (*Fairness*) per dividere il gioco tra i vari partecipanti. Il prossimo obiettivo è estendere questa tecnica nei modelli di *machine learning*.
In tale contesto, lo *Shapley-Value* permette di dividere, in modo equo, la previsione di un modello tra i regressori che lo costituiscono. Ne segue che lo *Shapley-Value* non è una tecnica di selezione delle variabili, ma è una tecnica di analisi dell'impatto delle varie *feature* nella previsione del modello scelto, e dunque d'*interpretabilità* del modello.

La formula per determinare lo *Shapley-Value* relativo all'i-esimo tra i p regressori (ϕ_i) è la seguente:

$$\phi_i = \sum_{S \subseteq \{1,\dots,p\}\{i\}} \frac{|S|!(p-|S|-1)!}{p!}[f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)] \qquad (3.2)$$

Nella formula (3.2) S rappresenta il modello di *machine learning* scelto con una determinata combinazione di *feature*, e di conseguenza $|S|$ corrisponde al numero di regressori. Ad esempio se S indicasse il nostro modello con i regressori $\{A, B, C\}$ allora $|S| = 3$, poiché è costituito da tre variabili. Per la definizione dei pesi, il ragionamento è identico al calcolo degli *Shapley-Value* nella teoria dei giochi, solo che in questo caso non si considerano dei giocatori, ma delle variabili o regressori. Il contributo marginale è definito come la differenza tra la previsione del modello utilizzando anche l'i-esima variabile ($f_{S\cup\{i\}}(x_{S\cup\{i\}})$) e la previsione dello stesso modello senza la i-esima variabile ($f_S(x_S)$).
Le proprietà degli *Shapley-Lorenz* valgono anche nel *machine learning*, nel dettaglio (*A Data Odyssey, 2023*):

1. *Efficienza*: la previsione è divisa tra i vari regressori. É verificabile che la predizione del modello è pari alla somma tra gli *Shapley-Value*, relativi a ognuno dei p regressori, e il valore atteso della previsione, ovvero:

$$f(x) = \sum_{i=1}^{p} \phi_i + E_x[f(x)]$$

2. *Simmetria*: due regressori avranno lo stesso *Shapley-Value* se daranno gli stessi contributi in tutte le combinazioni di modello.

3. *Dummy*: Se una variabile non influirà nella predizione in nessuna combinazione, allora avrà $\phi_i = 0$. Ne segue che le variabili non impiegate nel modello selezionato, non possono generare alcun contributo, e dunque non possono avere uno *Shapley-Value*.

4. *Additività*: nei modelli *ensemble*, ad esempio la *RandomForest* (RF), lo *Shapley-Value* finale è la media pesata degli *Shapley-Value* di tutti i modelli che lo costituiscono.

5. *Consistenza*: tale proprietà è esclusiva dell'applicazione degli *Shapley-Value* nel *machine learning*. Nel caso cui si utilizzi un nuovo modello e cambiano i contributi marginali delle variabili, allora anche gli *Shapley-Value* cambieranno nella "stessa direzione". Dunque è possibile confrontare i vari ϕ_i di differenti modelli.

Gli *Shapley-Value* sembrano essere una buona tecnica per l'interpretabilità dei modelli di *machine learning*, ma ci sono delle problematiche da considerare:

- *Computazionalmente onerosi*: All'aumentare del valore di p, ovvero del numero di regressori, ci sono più combinazioni da considerare e più contributi marginali e pesi da determinare. Una possibile soluzione è approssimare gli *Shapley-Value* attraverso un campionamento casuale (metodo *MonteCarlo*[12]).

- *Non sono normalizzati*: Se si confrontassero due modelli uno basato sui dati in forma originaria e uno sui dati scalati, avremmo degli

[12]In Python esiste la libreria "*SHAP*" che realizza in automatico il metodo *MonteCarlo*, solo che non è applicabile ad alcuni modelli, come le reti ricorrenti, ma è ottimo per i principali modelli di *machine learning*, quali la *RandomForest*, *Support Vector Machine*.

Shapley-Value molto differenti in magnitudo, poiché dipendono dall'unità di misura della previsione. Una possibile soluzione è l'impiego degli *Shapley-Lorenz Value* (SLV), analizzati nel dettaglio nel prossimo paragrafo.

3.2 S.A.F.E. AI

L'innovazione introdotta da Giudici e Raffinetti (2021) consiste nell'impiegare nella formula (3.2) lo zonoide di Lorenz, invece che la previsione del modello:

$$SLV(f(x))_i = \sum_{S \subseteq \{1,\dots,p\}\{i\}} \frac{|S|!(p-|S|-1)!}{p!}[LZ(f(x_{S\cup\{i\}})) - LZ(f(x_S))] \tag{3.3}$$

Di conseguenza viene determinato il *pay-off* tra lo zonoide di Lorenz relativo alla previsione del modello, nel quale è stata impiegata anche la i-esima variabile, e quello relativo alla previsione senza l'i-esimo regressore.
Poiché lo zonoide di Lorenz è per costruzione normalizzato, ne segue che anche lo SLV sarà normalizzato, rispetto ai classici SV, come verrà poi mostrato nell'ultimo capitolo, inerente all'analisi e presentazione dei risultati.
Gli *Shapley-Value* sono determinabili per ogni i-esima variabile e j-esima osservazione, mentre lo *Shapley-Lorenz Value* è calcolabile solo per ogni i-esimo regressore.
Ne consegue che determinato un modello di *machine learning*, si potrebbe ottenere come misura di interpretabilità del modello un ranking basato sugli SV e SLV. I risultati ottenuti tramite quest'ultimi saranno, inoltre, confron-

tabili tra differenti modelli, indipendetemente dalle trasformazioni attuate nelle variabili esplicative, quali differenzazione per la risoluzione della stazionarietà oppure una standardizzazione dei dati, in modo tale da impiegarli in un modello di rete neurale.
Oltre alla possibilità di confrontare il ranking e le magnitudo degli SLV tra differenti modelli, è possibile calcolare diverse metriche di interpretabilità di un determinato modello, sulla base di quattro aspetti (Giudici et al., 2023):

1. *Sustainability*: un modello sarà considerato *sostenibile*, se i suoi risultati sono stabili nel caso cui i dati siano stati manipolati o ci siano scenari anomali, quali gli attacchi informatici. É stata definita una metrica (*Sust-score*) per misurare la sostenibilità di un modello. La procedura consiste nel riordinare le previsioni, sui dati del test-set, in modo non crescente in base alla loro accuratezza nella predizione. Ad esempio, si potrebbero riorganizzare le previsioni in ordine non decrescente rispetto al valore del residuo. Le previsioni riordinate vengono poi suddivise in G gruppi omogenei rispetto alla numerosità, ad esempio, in base ai decili della distribuzione. Di seguito, si costruisce un vettore $V_G^{SL^*}$ i cui elementi , $V_g^{SL^*}$ con $g = 1, ..., G$, rappresentano la somma degli SLV dei vari regressori per quel g-esimo gruppo, ovvero $V_g^{SL^*} = \sum_{i=1}^{K} SLV_{i_g}$ per K regressori. Calcolati tutti gli elementi del vettore $V_G^{SL^*}$, viene determinato il valore dello zonoide di Lorenz ad esso associato $LZ(V_G^{SL^*})$, ne segue che la metrica relativa alla sostenibilità del modello è calcolabile come:

$$Sust - score = 1 - LZ(V_G^{SL^*}) \tag{3.4}$$

2. *Accuracy*: attraverso l'RMSE è possibile determinare l'accuratezza del modello, però dipende dall'unità di misura. Per ovviare tale problematica, è possibile impiegare lo zonoide di Lorenz anche come misura di bontà della previsione, realizzando un rapporto tra lo zonoide sulle previsioni del modello nel test-set e lo zonoide calcolato sulla variabile di risposta nel test-set:

$$Ac - score = \frac{LZ(\hat{Y}_{X_1,...,X_k})}{LZ(Y_{X_1,...,X_k})}. \tag{3.5}$$

   Lo zonoide è normalizzato per costruzione, inoltre attraverso il rapporto si ha una misura $\in [0, 1]$. Più lo *score* tende al valore unitario, maggiore è il grado di accuratezza del modello.

3. *Fairness*: un sistema di intelligenza artificiale è necessario che non presenti dei pregiudizi tra diversi gruppi del campione. É possibile determinare una misura di equità o imparzialità (*Fair-score*), con una procedura simile a quella per la sostenibilità. Si considerino $k^* < K$ regressori, scelti attraverso un processo di selezione delle variabili. Ad esempio, attraverso un'analisi della significatività dei coefficienti in un modello OLS, oppure attraverso una selezione con lo zonoide di Lorenz. Scelto il sottoinsieme di k^* regressori, l'obiettivo è misurare la concentrazione delle variabili esplicative che potrebbero essere affette da un *bias* negli SLV. Si dividono le previsioni sul test-set in $m = 1, ..., M$ gruppi, e viene costruito un vettore $V_M^{SL^*}$ i cui elementi $v_m^{SL^*} = \sum_{i=1}^{k^*} SLV_{i_m}$.

Si determina la seguente metrica:

$$Fair - score = 1 - LZ(V_M^{SL^*}). \tag{3.6}$$

Se tale metrica tende al valore nullo, allora gli SLV sono simili nei vari gruppi di popolazione. Se tende al valore unitario, allora l'effetto delle variabili dipende dal gruppo di appartenenza e dunque si ha un *bias*.

4. *Explainability*: precedentemente è stato presentato, come metodo per misurare l'interpretabilità di un modello di *machine learning*, l'impiego degli SLV, i quali rappresentano il contributo fornito da ogni regressore nella previsione del modello. Si determina successivamente una metrica per determinare il contributo di tutti i regressori rispetto alla configurazione complessiva dello zonoide di Lorenz per la variabile di risposta:

$$Ex - score = \sum_{i=1}^{K} SLV_i. \tag{3.7}$$

Si noti che per tale metrica gli SLV sono calcolati su tutto il `dataframe` di analisi, senza suddividere l'analisi in train e test set. Se il risultato della formula (3.7) tende al valore unitario, ne segue che i contributi di tutti regressori sono sufficienti per spiegare e prevedere la variabile di risposta.

Le metriche presentate da Giudici e Raffinetti (2023) sono normalizzate e, di conseguenza, possono essere impiegate per confrontare differenti modelli di *machine* e *deep learning* non solo in relazione all'accuratezza, ma anche, ad esempio, per la sostenibilità. Per definire le metriche, è necessario determi-

nare gli *Shapley-Values* attraverso un processo computazionalmente oneroso, soprattutto nel caso dell'impiego di modelli complessi quali le reti ricorrenti. Un aumento del numero di regressori genera un notevole incremento delle combinazioni dalle quali determinare, in primo luogo, le previsioni e poi gli SV e SLV.

Nel prossimo, e ultimo, paragrafo verrà presentato un caso pratico di analisi, nel quale verrà presentato come sono stati determinati gli SLV nel caso specifico dell'impiego di modelli di *machine* e *deep learning*.

Applicazione: previsione del prezzo del Bitcoin

La finanza è uno dei principali settori nei quali l'intelligenza artificiale acquisisce un ruolo fondamentale per poter analizzare dati di serie storiche. I dati delle serie temporali, frequenti nei modelli finanziari, e solitamente analizzati da modelli econometrici, vengono recentemente analizzati anche da modelli di machine learning e, in particolare, da reti neurali ricorrenti. Ciò consente di effettuare previsioni sui prezzi ad alta frequenza modellando le serie temporali dei prezzi come input (Cao, 2021). Vengono utilizzate anche reti feed-forward meno complesse, il cui funzionamento e applicazione sono stati analizzati, ad esempio, nel lavoro di Triebe et al. (2019). Verrà presentata un'applicazione pratica dell'approccio S.A.F.E. con l'impiego di reti neurali autoregressive e ricorrenti.

4.1 Analisi descrittiva

Si riprenda l'analisi svolta dai ricercatori Giudici Paolo e Raffinetti Emanuela (2023), relativa al prezzo del Bitcoin in un orizzonte temporale di circa due anni (dal 18 maggio 2016 al 30 aprile 2018), con l'impiego di modelli di reti neurali ricorrenti. I dati sono stati raccolti, selezionando l'orizzonte temporale di interesse, presso il sito *"Investing.com"* e attraverso *"Yahoo finance"* (cfr. *Sitografia per la raccolta dei dati*). Per l'analisi sono state impiegate le seguenti variabili:

Variabile	**Descrizione**
Bitcoin Price	Il prezzo del Bitcoin, espresso in dollari ($), è la variabile di risposta da prevedere.
USD\EUR	Il tasso di cambio Dollaro-Euro ($/€).
USD\YUAN	Il tasso di cambio Dollaro-Yuan ($/¥).
SP500 Price	L'indice azionario americano *"Standard & Poor 500"* espresso in dollari ($).
Gold Price	Il prezzo dell'oro espresso in dollari ($).
Oil Price	Il prezzo del petrolio espresso in dollari ($).

Tabella 4.1: Descrizione delle variabili

Per la determinazione della base di dati, da utilizzare per l'analisi, è stato eseguito un `"left join"` in base alla data del prezzo del Bitcoin. Ne segue però che, a seguito di tale procedura, per alcune variabili, quali il prezzo dell'oro e lo SP500, si riscontra dei valori mancanti per i giorni festivi e i feriali, a seguito di un'assenza di quotazione nel mercato. É stato, di conseguenza, eseguito un processo di pulizia e sostituzione dei dati mancanti al tempo t, con il dato ad esso precedente (t-*1*). Ad esempio, se per il prezzo dell'oro al giorno t si riscontra un `"NA"`, viene sostituito con il prezzo dell'oro

del giorno precedente (*t-1*). Di seguito sono mostrati tutti i grafici di serie storica relativi alla variabile dipendente e ai vari regressori:

Figura 4.1: *Grafico di serie storica del prezzo del Bitcoin (fonte:Python)*

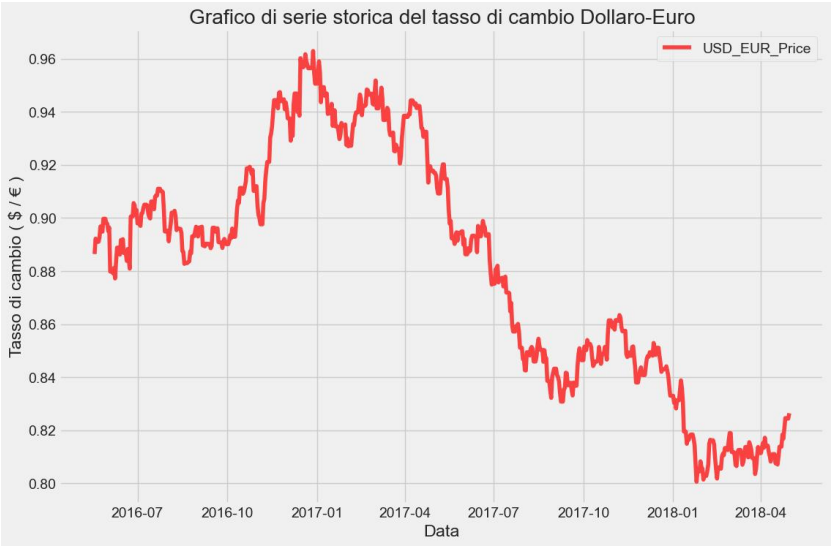

Figura 4.2: *Grafico di serie storica del Tasso di cambio Dollaro-Euro (fonte:Python)*

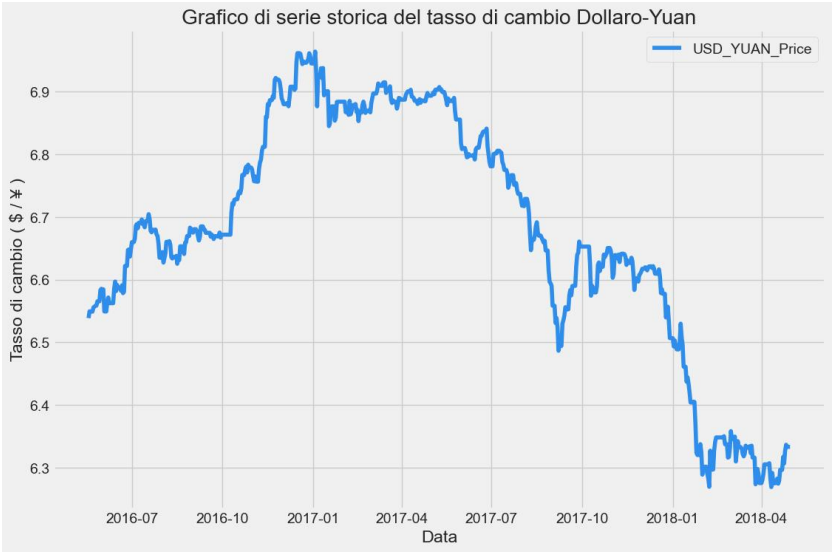

Figura 4.3: *Grafico di serie storica del Tasso di cambio Dollaro-Yuan (fonte:Python)*

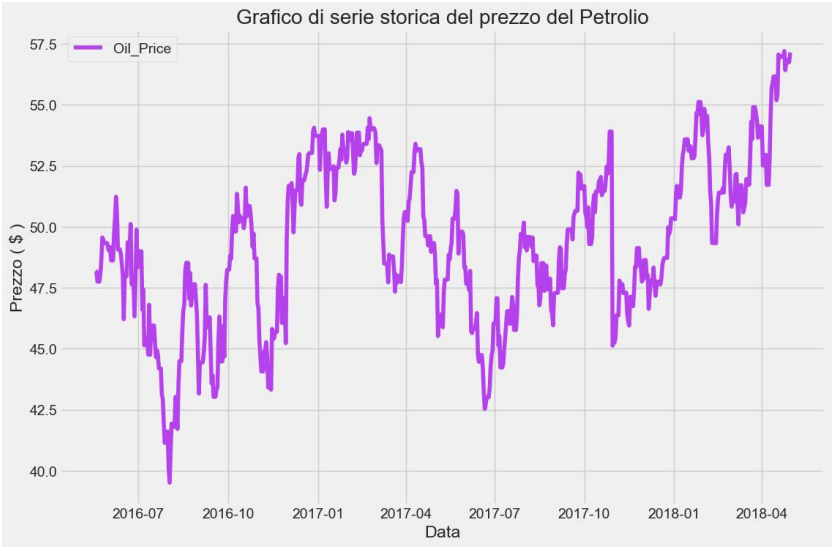

Figura 4.4: *Grafico di serie storica del Prezzo del Petrolio (fonte:Python)*

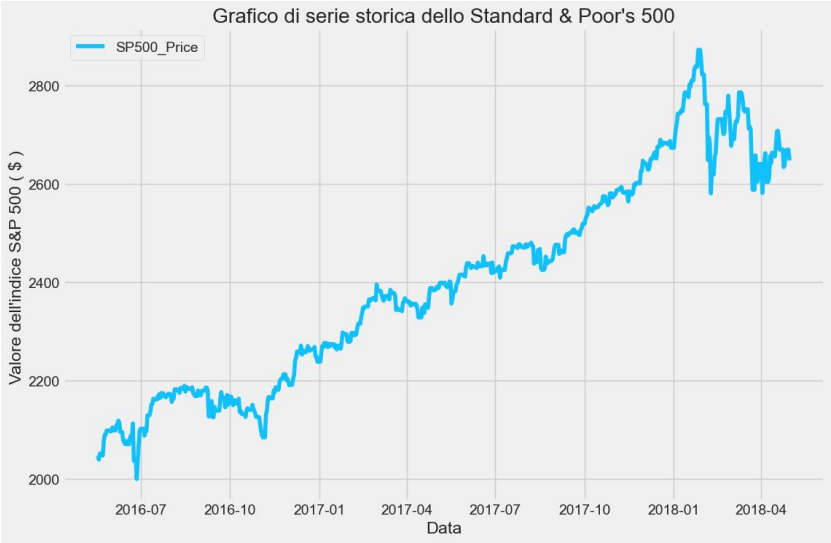

Figura 4.5: *Grafico di serie storica dello Standard & Poor's 500 (fonte:Python)*

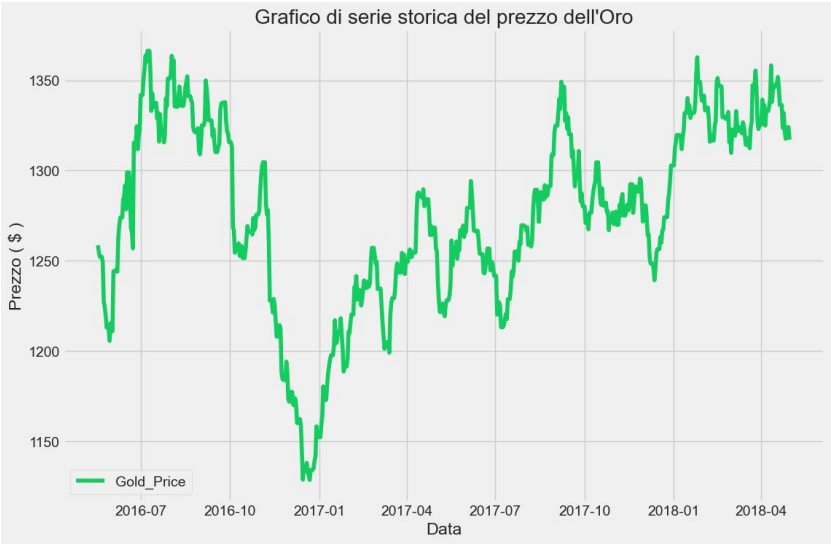

Figura 4.6: *Grafico di serie storica del prezzo dell'Oro (fonte:Python)*

Di seguito è presente una tabella per le statistiche descrittive di tutte le variabili:

Variabile	Minimo	Media	Mediana	Massimo	σ
BitcoinPrice	438.38	3919.05	1713.00	19 650.01	4318.98
USD_EUR	0.80	0.88	0.89	0.96	0.04
GoldPrice	1128.42	1275.57	1276.83	1366.38	52.34
SP500Price	2000.54	2399.17	2390.90	2872.87	212.31
USD_YUAN	6.27	6.68	6.67	6.96	0.19
OilPrice	39.51	49.36	49.30	57.20	3.37

Tabella 4.2: *Statistiche descrittive e deviazioni standard (σ) delle variabili*

La base di dati è caratterizzata da 713 osservazioni giornaliere ad alta frequenza. L'analisi verterà principalmente sui modelli di reti neurali ricorrenti, i quali sono in grado di gestire anche serie non stazionarie. Per tali modelli si necessita, però, che i dati siano scalati in un intervallo $[0, 1]$:

Variabile	Media	Mediana	σ
BitcoinPrice	0.18	0.07	0.22
USD_EUR	0.50	0.55	0.28
GoldPrice	0.62	0.62	0.22
SP500Price	0.46	0.45	0.24
USD_YUAN	0.58	0.58	0.27
OilPrice	0.56	0.55	0.17

Tabella 4.3: *Statistiche descrittive delle variabili scalate*

Nella Tabella 4.4 sono, inoltre, riportati i risultati dei principali test di stazionarietà sulle variabili:

	ADF			**PP**	**KPSS**
Variabile	**–c**	**–ct**	**–ctt**		
Bitcoin Price	×	×	×	×	×
USD/EUR	×	×	×	×	×
USD/YUAN	×	×	×	×	×
Gold Price	×	×	×	×	×
Oil Price	×	×	×	×	×
SP500 Price	×	✓	×	✓	×

Tabella 4.4: *Test di stazionarietà (fonte: Gretl)*

Per il test ADF sono stati considerati tre casi: presenza della costante (`--c`), del trend lineare (`--ct`) e del trend quadratico (`--ctt`). Il simbolo "×" rappresenta che, per quella determinata variabile, il test non rifiuta un'ipotesi di assenza di stazionarietà, con un livello di significatività del 5%. Solo per lo *"Standard & Poor 500"* due test, il caso di costante e trend lineare dell'ADF e il Phillips-Perron (PP), rifiutano l'ipotesi nulla e suggeriscono che la serie può essere considerata stazionaria. Si ricorda però che gli altri tre test suggeriscono l'opposto.

Ne segue che per i modelli classici è necessario impiegare la serie dei rendimenti, diversamente da quella dei prezzi, poichè caratterizzati da un problema di stazionarietà nella quasi totalità delle variabili d'interesse.

4.2 Definizione delle funzioni su R e Python

Per la rete *feed-forward* è stato impiegato come software di analisi *R-Studio*, mentre per le due reti ricorrenti è stato utilizzato *Python*. Per entrambi i software sono state ricostruite delle funzioni per calcolare gli *Shapley-Value* (SV) e gli *Shapley-Lorenz Value* (SLV).
Le principali funzioni costruite su *R-Studio* sono le seguenti:

1. `lz(y,y_prev)`: per calcolare il Lorenz Zonoid di una previsione `y_prev`($\hat{y}$).

2. `model_comb(data)`: per creare una lista di tutte le combinazioni di regressori, nelle quali viene considerata la prima colonna di `data` come variabile dipendente (y). Da notare che in `data` non deve esserci la variabile *"Date"*, oppure bisogna impostarla come indice.

3. `preds_lm(train, test, lista_formule)`: per creare un dataset con tutte le previsioni di ogni modello OLS che impiega come regressori quelli presenti nella lista delle formule (output di `model_comb`).

4. `preds_nnetar(train, test, lista_formule,P=2, size=4,repeats=1, seed=87)`: Esattamente come `preds_lm()` solo che in questo caso viene impiegato un modello NNAR, invece che quello OLS. Gli iper-parametri definiti per default sono stati scelti precedentemente dividendo il dataset tra training, validation e test e minimizzando l'*RMSE* del validation set. Per quanto riguarda il `seed`, serve per avere la certezza di impiegare sempre la rete con la stessa inizializzazione di pesi.

5. `sv_weights(data, lista_formule)`: per la creazione dei pesi per gli SV e SLV.

6. `slv_plot(test, all_prev, lista_pesi)`: per calcolare gli SLV per tutti i regressori presenti nel test-set e salvarli in un dataframe. L'argomento `all_prev` è il risultato della funzione `preds_lm()` o `preds_nnetar()`, mentre `lista_pesi` è il risultato della funzione `sv_weights()`. Attraverso il comando "`$Plot`" è possibile visualizzare un ranking con un bar plot.

7. `sv_plot(test, all_prev, lista_pesi)`: Esattamente come `slv_plot()`, solo che in questo caso si ha come unico output il dataframe, non è presente l'operazione `$Plot`.

Invece su Python sono state create varie funzioni solo per realizzare il c.d. "`all_prev`" su reti ricorrenti, da importare poi su *R-Studio* per calcolare gli SV o SLV:

1. `model_comb2(data)`

2. `data_from_formula(data, formula)`: Questa funzione è per definire una lista di DataFrame aventi come regressori quelli della i-esima formula considerata. Inoltre `formula` è l'output di `model_comb2()`.

3. `generic_lstm(data,time, formula,seed=13, lag=60, epochs=5, batch_size=1, units=50, learning_rate=0.001)`: per realizzare un modello *LSTM* usando come train data l'intero dataset fino alla data espressa da "`time`", come test set i dati dopo la data "`time`". La funzione restituisce la serie delle previsioni.

4. `generic_gru(data,time, formula,seed=53, lag=60, epochs=5, batch_size=1, units=50, learning_rate=0.001)`.

5. `preds_lstm(data,time,seed=13, lag=60, epochs=5, batch_size=1, units=50, learning_rate=0.001)`: genera il dataset con tutte le previsioni per ogni combinazione di regressori (creazione di `all_prev`).

6. `preds_gru(data,time,seed=53, lag=60, epochs=5, batch_size=1, units=50, learning_rate=0.001)`.

Tutte le funzioni realizzate sono state poi verificate con l'analisi svolta da Giudici e Raffinetti in *"Lorenz Model Selection"* (2020) e *"Shapley-Lorenz eXplainable Artificial Intelligence"*(2021), riottenendo gli stessi risultati per il calcolo degli *Shapley-Lorenz Value* (SLV).

4.3 Applicazione

Nel seguente paragrafo saranno riportati i risultati ottenuti dall'approccio S.A.F.E. nelle reti neurali. Nei vari modelli di *machine* e *deep learning* saranno impiegate le osservazioni fino al 31 dicembre 2017 come dati di addestramento (*train set*), le restanti come *test-set*. Per la definizione degli iperparametri, sarà necessario suddividere i dati di addestramento, in prossimità del 30 settembre 2017, per determinare un *validation set* rappresentativo di circa il 30% del *train set*.

4.3.1 Rete neurale autoregressiva: NNAR

Il modello NNAR necessita che i dati vengano prima posti in un intervallo $[0, 1]$. Per la scelta degli iper-parametri è stata condotta un'analisi con l'obiettivo di minimizzare l'errore quadratico medio (*RMSE*) del *validation set*, sono riportate di seguito le migliori e peggiori combinazioni:

P	size	repeats	Train RMSE	Validation RMSE
2	4	1	0.1307	0.2801
6	7	1	0.0103	0.2820
1	2	1	0.0240	0.2851
⋮	⋮	⋮	⋮	⋮
4	7	10	0.0125	0.3573
6	7	5	0.0097	0.3644
5	7	1	0.0087	0.4477

Tabella 4.5: *Determinazione degli iper-parametri del modello NNAR*

In riferimento ai risultati della Tabella 4.5, la rete migliore sembra essere quella caratterizzata da quattro neuroni nello strato nascosto e da due ritardi stagionali come input. É interessante notare che all'aumentare del numero di neuroni dello strato nascosto, il *RMSE* relativo ai dati di addestramento della rete diminuisce (*Train RMSE*), mentre aumenta quello associato al set di validazione (*Validation RMSE*). Risulta che un incremento dell'iper-parametro "`size`" possa generare un problema di *overfitting*, ovvero ottime perfomance nei dati di addestramento e risultati non ottimali con nuovi dati. Nel momento cui la rete viene applicata ai dati disponibili di addestramento (*train* e *validation set*), il modello genera le previsioni nel *test set* rappresentate nella Figura 4.7:

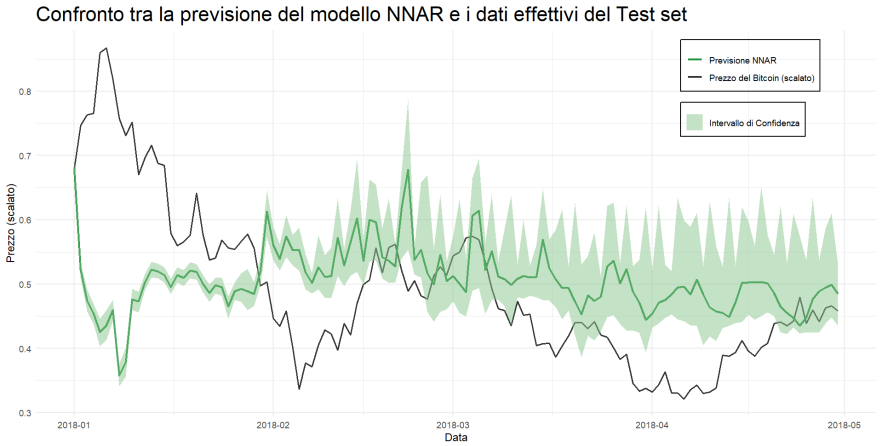

Figura 4.7: *Previsioni del modello NNAR sul test set (fonte:R-Studio)*

La libreria `nnet`, utile per impiegare il modello NNAR su *R-Studio*, permette di ricavare anche gli intervalli di confidenza della previsione, settando il parametro `PI` come `TRUE`. Per maggiori dettagli riportiamo le prime e ultime previsioni in forma originaria, ovvero il prezzo del Bitcoin, nella seguente tabella:

Data	Bitcoin ($)	Previsione ($)	LWR ($)	UPR ($)
2018-01-01	13480.01	13518.718	13284.962	13750.607
2018-01-02	14781.51	10458.047	10218.373	10700.503
2018-01-03	15098.14	9558.849	9239.164	9887.196
2018-01-04	15144.99	9166.688	8847.983	9465.038
⋮	⋮	⋮	⋮	⋮
2018-04-25	8865.98	9047.116	8605.105	10748.75
2018-04-26	9272.12	9592.944	8592.547	12619.88
2018-04-27	8922.55	9831.291	8603.605	10756.65
2018-04-28	9329.99	9944.927	8873.944	11716.88

Tabella 4.6: *Previsioni ($), Estremo Inferiore (LWR) e Superiore (UPR) del modello NNAR sul test set*

Dalle previsioni, espresse però nell'intervallo $[0, 1]$, sono stati ricavati gli *Shapley value*, rappresentati di seguito:

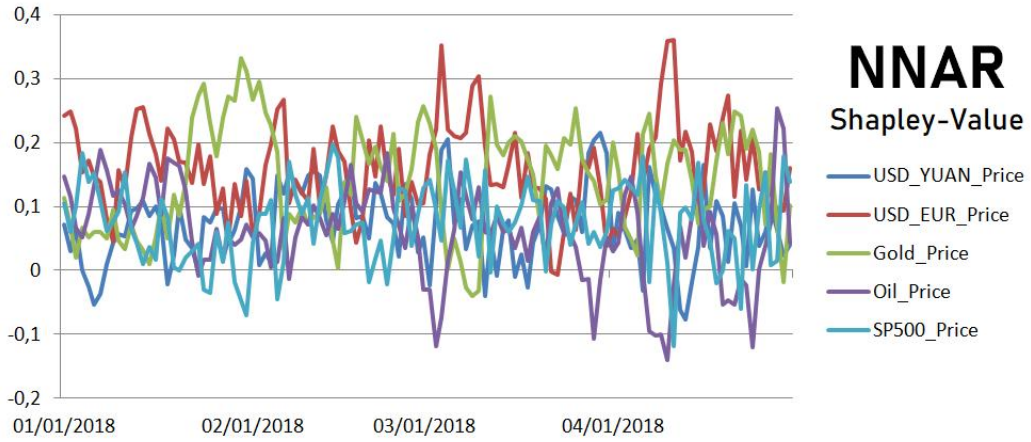

Figura 4.8: *Shapley-Values del modello NNAR nel test set (fonte:R-Studio)*

Una volta ottenuto il set di dati con tutti gli SV per ogni regressore e per ogni giorno del test set, è possibile determinare diverse metriche, come la loro somma, il loro valore atteso e il loro valore atteso assoluto, come nella Tabella 4.7 sotto, con l'obiettivo di fornire un *ranking* del contributo di ogni variabile alla previsione.

Variabile	Somma(SV)	Media(SV)	Media Assoluta(SV)
USD_EUR	19.2306	0.1603	0.1604
GOLD	16.9857	0.1415	0.1435
SP500	8.7799	0.0732	0.0816
USD_YUAN	8.6867	0.0724	0.0797
OIL	7.1530	0.0596	0.0796

Tabella 4.7: *Shapley-Value Ranking (modello NNAR)*

Dalla Tabella 4.7 si nota che il tasso di cambio Dollaro-Euro e il prezzo dell'oro sembrano generare, in magnitudo, un contributo medio due volte, circa, superiore rispetto al tasso di cambio Dollaro-Yuan e agli altri regressori.

La Tabella 4.8 riporta i valori degli *Shapley-Lorenz* per lo stesso modello e gli stessi dati.

Variable	Shapley-Lorenz Value (SLV)
USD_EUR	0.1212
GOLD	0.0238
SP500	0.0163
OIL	−0.0086
USD_YUAN	−0.0706

Tabella 4.8: Shapley-Lorenz Value Ranking (modello NNAR)

Dalla Tabella 4.8 si nota che la classifica degli SLV sembra riflettere la classifica fornita dal valore assoluto medio degli SV. Nel prossimo paragrafo analizziamo il caso più complesso delle reti neurali ricorrenti.

4.3.2 Reti neurali ricorrenti: LSTM & GRU

Per la determinazione degli iper parametri, anche per le reti ricorrenti, è stato impiegato il *validation set*. I risultati, quali il RMSE, sono stati poi riportati in forma di prezzo ($):

Batch Size	Epochs	Units	Learning Rate	RMSE Train	RMSE Validation
5	10	100	0.01	134.16	930.96
5	10	50	0.01	128.23	1150.93
5	10	100	0.001	134.16	1342.86
5	5	50	0.01	206.64	1512.21
⋮	⋮	⋮	⋮	⋮	⋮
1	10	50	0.1	164.61	8417.29
1	10	100	0.1	212.04	9004.83
1	10	50	0.01	164.61	9300.85
1	5	100	0.1	196.37	9364.49

Tabella 4.9: *Risultati LSTM ordinati per il Validation RMSE*

I valori rappresentati nella Tabella 4.9 sono stati ordinati in modo non crescente rispetto al RMSE nel *validation set*. Si nota che la migliore rete è quella caratterizzata da un *learning rate* pari a 0.01, e nonostante un valore elevato di unità, epoche e `batch`, non dovrebbe esserci un problema di *overfitting*, poichè si hanno ottimi risultati sia nei dati di addestramento che in quelli di validazione.

La previsione, ottenuta impiegando il modello *LSTM* per il *test set* scalato, è rappresentata nella figura sottostante, in cui la linea rossa rappresenta il *test set* originale, quella blu il *train set*.

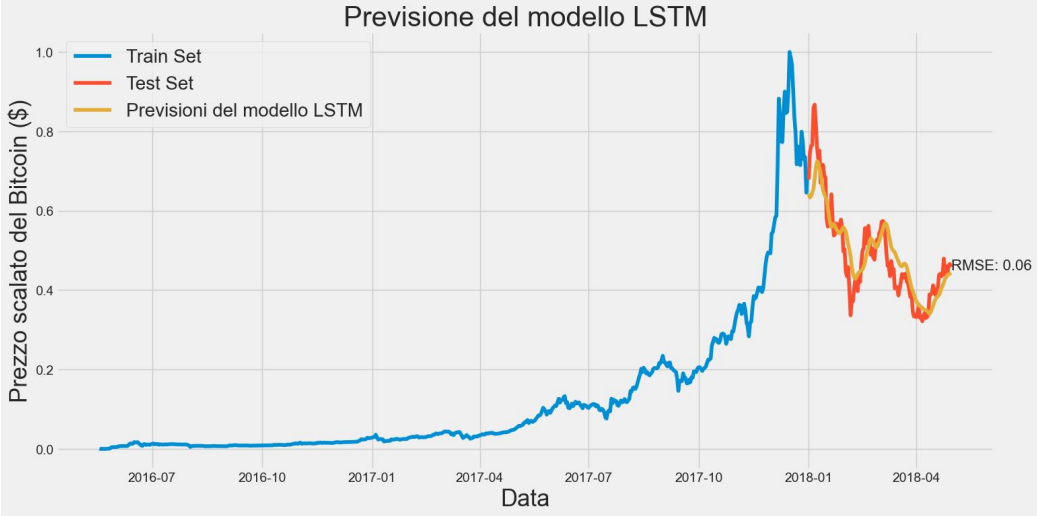

Figura 4.9: *Previsioni del modello LSTM (fonte:Python)*

Dalla Figura 4.9 si nota che le previsioni, rappresentate in giallo, sembrano essere più accurate e precise rispetto al modello NNAR. Successivamente le previsioni sono state riportate in forma di prezzo e rappresentate con il relativo intervallo di confidenza, ottenendo il seguente grafico di serie storiche:

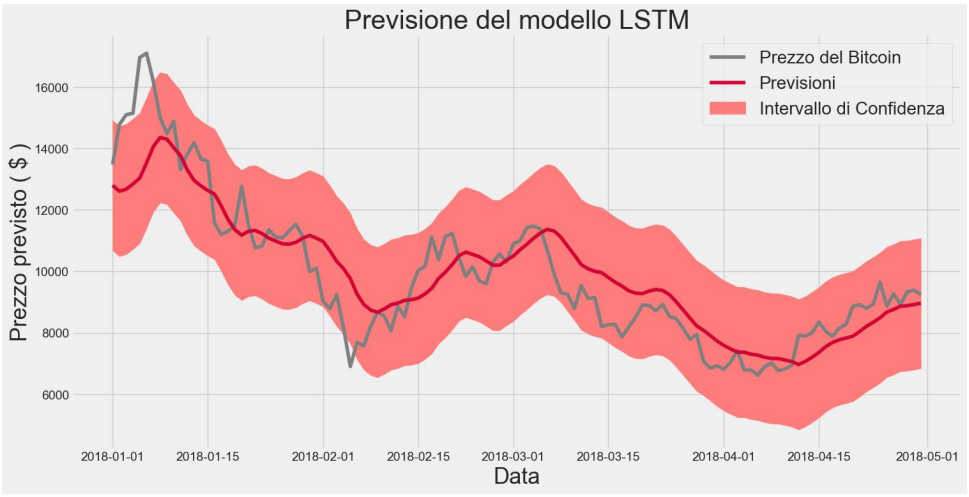

Figura 4.10: *Previsioni e Intervallo di confidenza LSTM (fonte:Python)*

Un'ulteriore figura rappresenta gli SV ottenuti dalle previsioni del prezzo del Bitcoin, impiegando il modello LSTM:

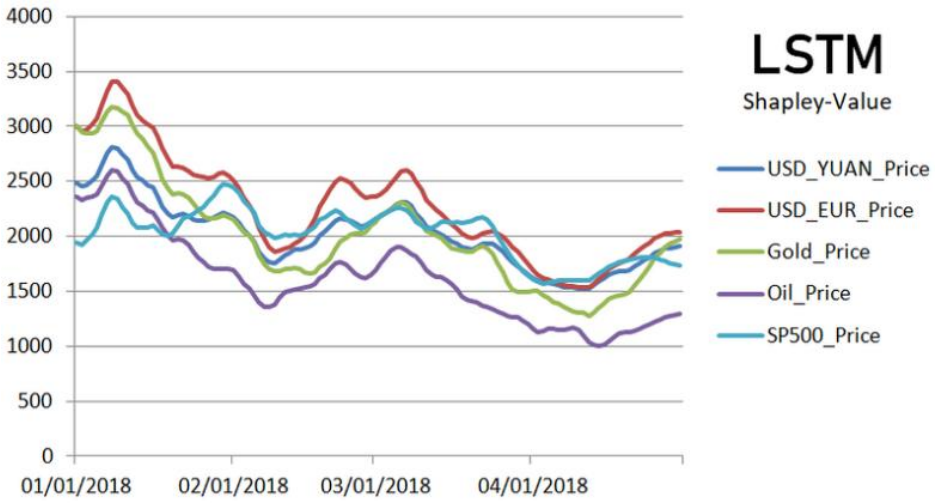

Figura 4.11: *Shapley-Values del modello LSTM nel test set (fonte:R-Studio)*

Per analizzare nel dettaglio il *ranking* dei regressori che generano un maggiore contributo nella previsione del modello LSTM, sono state calcolate alcune metriche. Si noti che, poiché gli *Shapley-Value* risultano essere tutti positivi (Figura 4.11), non è necessario calcolare il valore assoluto.

Variabile	Somma(SV)	Media(SV)
USD_EUR	271 552.1839	2262.9349
USD_YUAN	242 424.0912	2020.2008
SP500	242 122.1858	2017.6849
GOLD	240 633.0369	2005.2753
OIL	192 974.4335	1608.1203

Tabella 4.10: *Shapley-Value Ranking (modello LSTM)*

Di seguito sono, invece, presentati gli *Shapley-Lorenz Values* associati:

Variabile	Shapley-Lorenz Value (SLV)
USD_EUR	0.0574
GOLD	0.0305
USD_YUAN	0.0229
SP500	0.0156
OIL	−0.0143

Tabella 4.11: *Shapley-Lorenz Value Ranking (modello LSTM)*

Si noti che gli SV e SLV generano un *ranking* alquanto simile, ad eccezione della variabile "*GOLD*" che acquisisce un'importanza diversa. Nel complesso, tutte le variabili sembrano dare un piccolo contributo, inferiore rispetto al modello NNAR. Ciò può essere dovuto al fatto che la rete LSTM risulta essere più accurata e precisa rispetto al modello NNAR. È interessante notare che con un modello con sole variabili non negative, come i prezzi, gli SLV ci consentono di identificare eventuali contributi non positivi alle previsioni del modello.

L'ultima rete ricorrente da analizzare è la GRU, per la quale sono stati determinati gli iper-parametri attraverso lo stesso *validation set*. In questo caso la migliore combinazione di parametri è caratterizzata da un *learning rate* pari a 0.001, un valore elevato di 100 unità, 10 epoche e un solo `batch`. Nonostante non sia il caso con più basso RMSE sui dati di addestramento, si ha un ottimo risultato nel RMSE del *validation set*, suggerendo che non dovrebbe esserci un problema di *overfitting*.

Batch Size	Epochs	Units	Learning Rate	RMSE Train	RMSE Validation
1	10	100	0.001	230.09	905.45
5	10	100	0.01	110.16	1265.65
1	5	50	0.01	144.04	1403.96
1	10	50	0.001	177.15	1435.28
⋮	⋮	⋮	⋮	⋮	⋮
5	10	100	0.1	110.16	8443.14
1	5	50	0.1	144.04	8582.64
1	10	100	0.1	230.09	8712.70
5	5	100	0.1	109.70	9677.17

Tabella 4.12: *Risultati GRU ordinati per il Validation RMSE*

Nel *test set* è stata ottenuta la seguente previsione sui dati scalati:

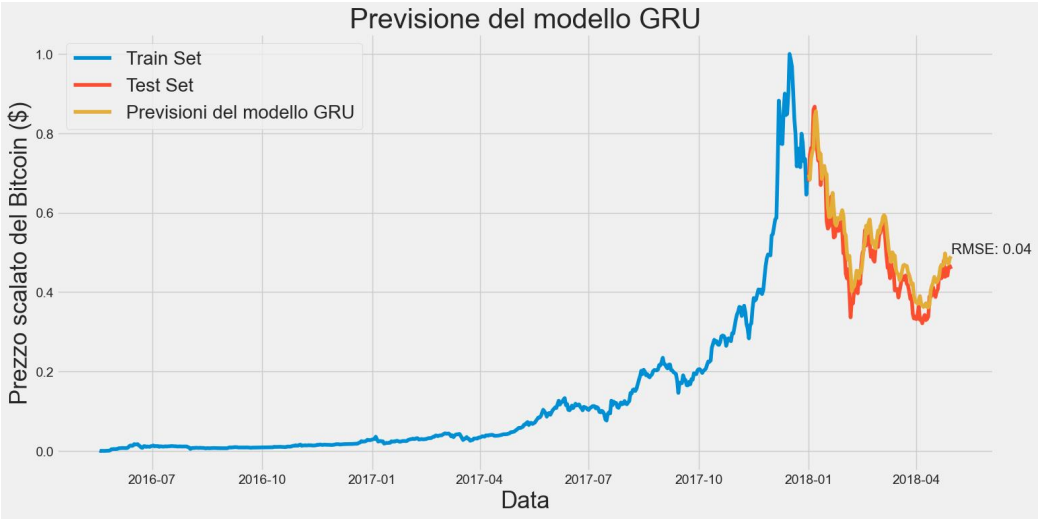

Figura 4.12: *Previsioni del modello GRU (fonte:Python)*

Il modello GRU risulta essere più accurato, rispetto alla rete LSTM, poiché caratterizzato da un RMSE inferiore, pari a circa 0.0439. Si osservi nel dettaglio anche l'intervallo di confidenza della GRU:

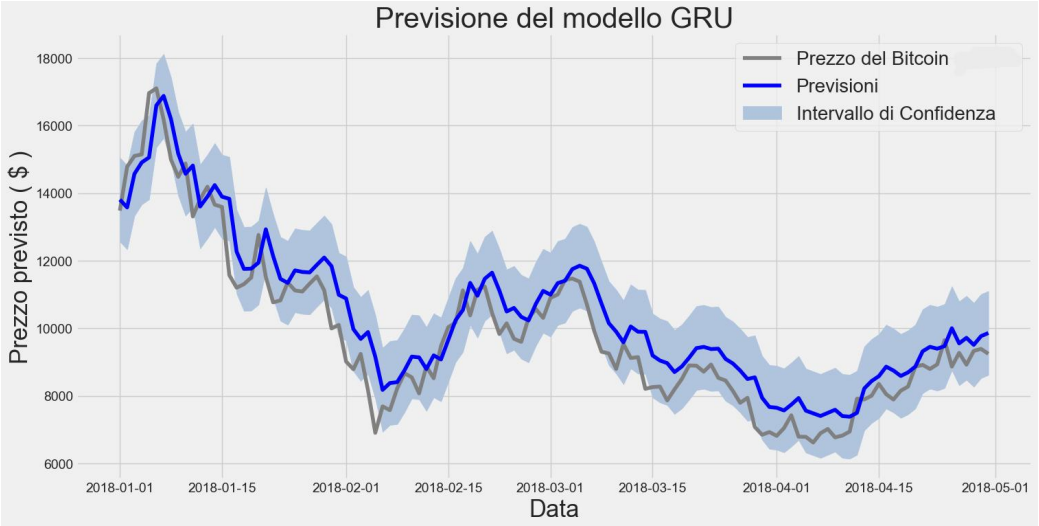

Figura 4.13: *Previsione e Intervallo di Confidenza GRU (fonte:Python)*

Dalla Figura 4.13 è osservabile come l'intervallo di confidenza della GRU è meno esteso rispetto a quello della LSTM (Figura 4.10). Di seguito riportiamo una tabella delle previsioni e degli intervalli di confidenza:

Data	Bitcoin ($)	Previsione ($)	LWR ($)	UPR ($)
2018-01-01	13480.01	13807.20	12565.17	15049.23
2018-01-02	14781.51	13575.91	12333.88	14817.95
2018-01-03	15098.14	14569.58	13327.54	15811.61
⋮	⋮	⋮	⋮	⋮
2018-04-28	9329.99	9508.49	8266.45	10750.52
2018-04-29	9389.01	9766.39	8524.35	11008.42
2018-04-30	9243.83	9859.48	8617.44	11101.51

Tabella 4.13: *Previsioni GRU ($), Estremo Inferiore (LWR) e Superiore (UPR)*

Si analizzino gli Shapley-Value delle previsioni della rete GRU:

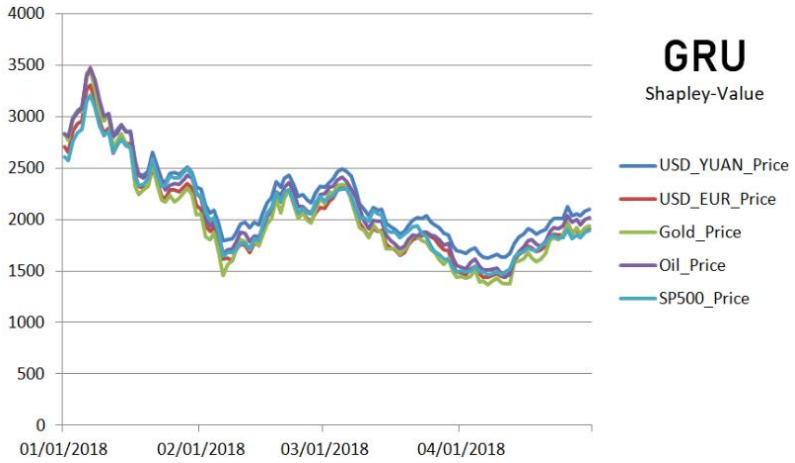

Figura 4.14: *Shapley-Value del modello GRU (fonte:R-Studio)*

Per la robustezza rispetto alla scalabilità dei dati, applichiamo lo stesso modello ai dati scalati. Gli Shapley-Value sono i seguenti:

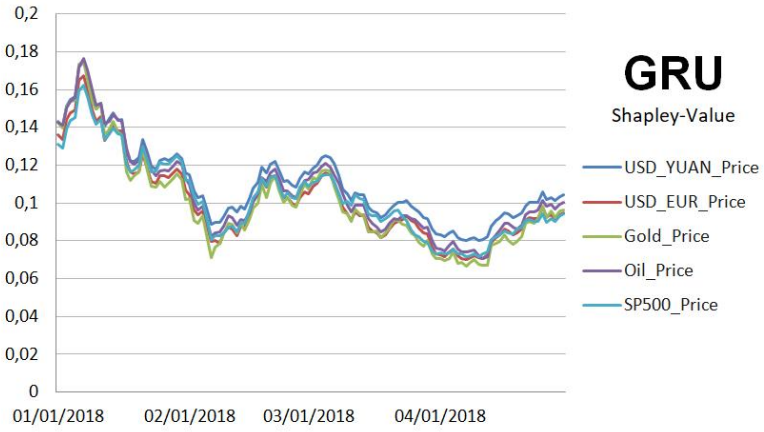

Figura 4.15: *Shapley-Value del modello GRU con dati scalati (fonte:R-Studio)*

Gli Shapley Value rimangono quasi gli stessi, indipendentemente dalla diversa scala. Si confronti ora le metriche relative agli SV e SLV per le due situazioni, dati in forma originale e scalati:

Dati Originali (GRU)		
Variabile	Somma(SV)	Media(SV)
USD_YUAN	264014.9535	2200.125
OIL	253360.5201	2111.3377
SP500	247078.6633	2058.9889
USD_EUR	243644.1917	2030.3683
GOLD	241401.6594	2011.6804

Dati scalati (GRU)		
Variabile	Somma(SV)	Media(SV)
USD_YUAN	13.1948	0.1100
OIL	12.6402	0.1053
SP500	12.3132	0.1026
USD_EUR	12.1345	0.1011
GOLD	12.0177	0.1001

Tabella 4.14: *Confronto SV nel modello GRU tra dati originali e scalati*

Original Data (GRU)	
Variable	Shapley-Lorenz Value (SLV)
USD_YUAN	0.0463
OIL	0.0396
SP500	0.0322
GOLD	0.0230
USD_EUR	0.0298

Scaled Data (GRU)	
Variable	Shapley-Lorenz Value (SLV)
USD_YUAN	0.0484
OIL	0.0414
SP500	0.0337
GOLD	0.0314
USD_EUR	0.0312

Tabella 4.15: *Confronto SLV nel modello GRU tra dati originali e scalati*

Dalle tabelle di cui sopra si nota che la classifica degli SV e SLV non cambia in seguito a un *rescaling*, quindi il sistema sembra essere robusto rispetto alla scalabilità dei dati. Attraverso gli *Shapley-Value* è possibile analizzare il contributo di quel determinato predittore alla previsione del modello. Ad esempio, il tasso di cambio Dollaro-Yuan genera un aumento in media pari a circa 2200$ al valore medio della previsione del prezzo del Bitcoin. Generalmente tutti i predittori generano un contributo medio alquanto simile. Ovviamente l'interpretazione cambia se si impiega un modello i cui risultati sono scalati in un intervallo $[0, 1]$. Il tasso di cambio Dollaro-Yuan, nel caso scalato, genera un contributo medio positivo pari a circa 0.11. Poichè i dati sono scalati, è possibile affermare che si ha un aumento dell'11% rispetto al valore medio della previsione. Interessante è l'impiego degli SLV, in questo caso si dimostra che, oltre a mantenere una classificazione alquanto simile rispetto a quella proposta dagli SV, è direttamente confrontabile. Il tasso di cambio Dollaro-Yuan genera, nelle previsioni espresse come prezzi, un aumento di circa il 4.6% nello zonoide della previsione. Nel caso scalato si ha un aumento del 4.8%, molto simile rispetto ai risultati sui prezzi. Tale fenomeno si ripresenta per anche gli altri regressori. É possibile, dunque, realizzare un confronto tra differenti modelli, impiegando direttamente gli SLV, senza preoccuparsi del problema della scalabilità dei risultati.

4.4 Risultati e confronto delle metodologie

In questa sezione si propone un confronto di tutti i risultati ottenuti, per analizzare se si riscontrano fenomeni simili rispetto alle diverse reti impiegate.

I modelli NNAR e GRU non presentano differenze significative tra la classifica ottenuta con il classico Shapley-Value e quella ottenuta impiegando lo zonoide di Lorenz, mentre nel modello LSTM, il prezzo dell'oro (*GOLD*) sembra acquisire una diversa importanza. Se si confrontano le classifiche ottenute tra i vari modelli, notiamo delle disposizioni diverse in relazione alla rete impiegata. É, però, utile ricordare che le magnitudo, negli SV e negli SLV, sono alquanto basse, soprattutto nei modelli LSTM e GRU, le cui previsioni sono più accurate rispetto al modello NNAR.

Shapley-Value		
NNAR	LSTM	GRU
USD_EUR	*USD_EUR*	*USD_YUAN*
GOLD	*USD_YUAN*	*OIL*
SP500	*SP500*	*SP500*
USD_YUAN	*GOLD*	*USD_EUR*
OIL	*OIL*	*GOLD*

Shapley-Lorenz Value		
NNAR	LSTM	GRU
USD_EUR	*USD_EUR*	*USD_YUAN*
GOLD	*GOLD*	*OIL*
SP500	*USD_YUAN*	*SP500*
OIL	*SP500*	*USD_EUR*
USD_YUAN	*OIL*	*GOLD*

Tabella 4.16: *Confronto del ranking ottenuto nelle tre reti neurali*

Per analizzare anche le perfomance dei vari modelli, nella Tabella 4.17, sono riportate le principali metriche S.A.F.E. e gli RMSE, espresso in termini di prezzo e in forma scalata:

Model	**Su-Score**	**Ac-Score**	**Ex-score**	**RMSE ($)**	**RMSE (scaled)**
NNAR	0.7157	0.3718	0.6326	2608.75 $	0.1358
LSTM	0.9607	0.8186	0.5693	1078.07 $	0.0561
GRU	0.9244	0.8865	0.6282	844.20 $	0.0439

Tabella 4.17: *Confronto delle metriche S.A.F.E. nei modelli*

Nella Tabella 4.17 si osserva la difficoltà nel confrontare le reti neurali in relazione al RMSE, poichè dipende dall'unità di misura. Ipotizziamo che le reti ricorrenti siano state esaminate solo in termini di prezzi, mentre la NNAR solo con dati scalati, ne segue che l'RMSE non sarà più una metrica affidabile per il confronto. Si ripropone lo stesso problema inerente al confronto degli SV relativi alle previsioni in forma originale e scalata. In tale contesto, per il modello NNAR si avrebbe un RMSE approssimativamente uguale a 0.14, mentre nelle due reti ricorrenti è più alto, 1078.07 (LSTM) e 844.20 (GRU), ma solo perché in quest'ultime le previsioni sono espresse in termini di prezzi ($).

Nella Tabella 4.17 non si presenta tale problematica, perchè sono stati riportati gli RMSE per entrambe le situazioni. Invece le metriche S.A.F.E. risultano sempre direttamente confrontabili, indipendentemente dall'unità di misura delle previsioni e della variabile di risposta. Se confrontiamo le accuratezze dei modelli, le reti ricorrenti risultano avere delle perfomance migliori nel *test set*, diversamente dalla rete *feed-forward* autoregressiva, la quale riesce a spiegare solo il 37% circa dello zonoide della variabile di risposta.

Per quanto riguarda la sostenibilità, le reti ricorrenti risultano essere le migliori, in particolare la LSTM. Di conseguenza si ha che ordinando le previsioni per la loro accuratezza e dividendole in dieci gruppi, non si riscontrano differenze significative nelle magnitudo per ogni regressore. La rete NNAR invece riscontra qualche differenza in base all'accuratezza del gruppo, tale da ridurre la metrica relativa alla sicurezza, o sostenibilità, del modello.
La metrica relativa alla spiegabilità (*Ex-score*) è più delicata da analizzare. Il modello NNAR sembra essere quello caratterizzato da una maggiore stabilità, ma è anche il meno accurato. Se un modello è caratterizzato da una buona accuratezza, lo SLV ci suggerisce quali regressori hanno maggiormente contribuito ad ottenere i buoni risultati rappresentati dalla previsione. In caso contrario, una previsione non accurata, lo SLV suggerisce sempre quali sono i contributi dati da ogni regressore però per una previsione non ottimale. Ne segue dunque che la metrica *Ex-Score* deve essere sempre confrontata in relazione all'accuratezza del modello. Nelle reti ricorrenti più accurate, l'*Ex-score* è comunque alquanto elevato. Il problema consiste dunque nel fatto che la metrica della spiegabilità considera solo la previsione, non la variabile di risposta, diversamente dall'accuratezza.
Successivamente si confrontano gli *Shapley-Value* di ogni osservazione in relazione al valore del relativo regressore, attraverso degli *swarm plot* per ogni modello analizzato. Si ricorda che i risultati delle reti ricorrenti sono espressi in prezzo, mentre quelli della rete autoregressiva sono in forma scalata.

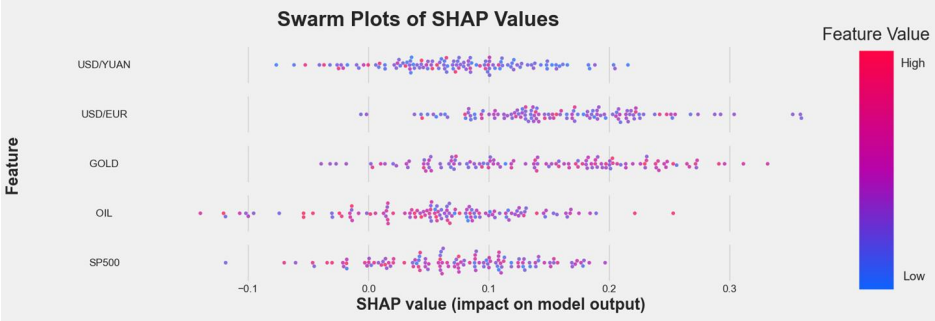

Figura 4.16: *Swarm Plot degli Shapley-Value (NNAR) (fonte:Python)*

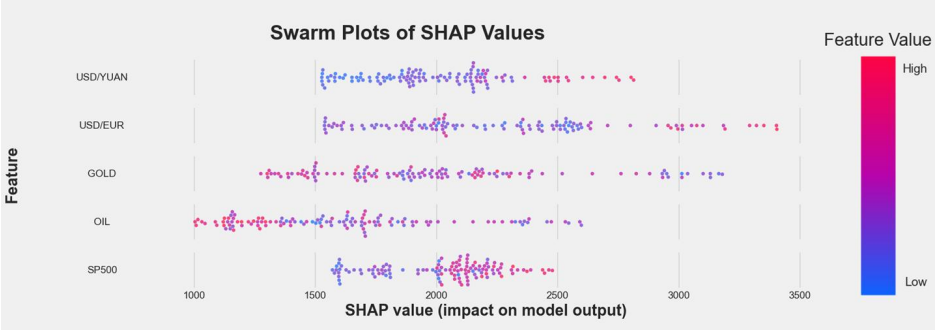

Figura 4.17: *Swarm Plot degli Shapley-Value (LSTM) (fonte:Python)*

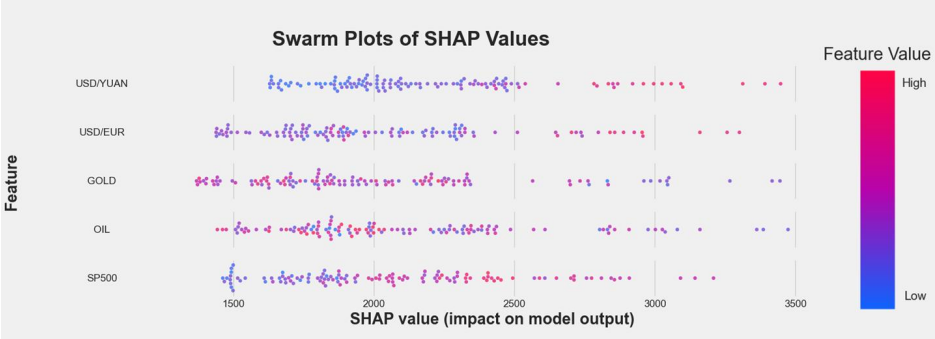

Figura 4.18: *Swarm Plot degli Shapley-Value (GRU) (fonte:Python)*

Nelle reti ricorrenti (Figura 4.17 e 4.18) è osservabile un legame tra la disposizione degli *Shapley-Value* e il colore, associato al valore della variabile (*Feature Value*), per il tasso di cambio Dollaro-Yuan. Di conseguenza, per valori elevati di tale regressore si riscontra un maggiore contributo, e dunque uno *Shapley* più elevato, nella previsione. Un comportamento simile si verifica anche per lo *Standard & Poor's* (*SP500*). Per il prezzo dell'oro e del petrolio, sempre nelle reti ricorrenti, si evidenza un leggero legame tra colore e intensità dello *Shapley-Value*. In particolare, si mostra una concentrazione di colori tendente al rosso per zone con *Shapley-Value* meno intensi. Risulta che, nelle reti ricorrenti impiegate, un aumento del valore del prezzo dell'oro e del petrolio possa apportare un contributo minore in magnitudo.
Per la rete autoregressiva (Figura 4.16) non si evidenziano molti legami tra colore e contributo apportato dal regressore nel modello. Per avere, però, una visione più chiara, sono state calcolate le correlazioni tra lo *Shapley-Value* e il valore della variabile di interesse (*Feature Value*):

Modello	**USD/YUAN**	**USD/EUR**	**GOLD**	**OIL**	**SP500**
NNAR	- 0.26	0.0012	0.33	- 0.20	- 0.24
LSTM	0.86	0.47	- 0.35	- 0.43	0.68
GRU	0.89	0.54	- 0.27	- 0.23	0.66

Tabella 4.18: *Confronto delle correlazioni tra Shapley e Valore della variabile*

La Tabella 4.18 conferma le osservazioni sugli *swarm plot*. In particolare nel modello NNAR non risulta esserci, in generale, un legame tra contributi nella previsione e valore del regressore. Nelle reti ricorrenti si riscontra una forte correlazione positiva per il tasso di cambio Dollaro-Yuan e *SP500*, alquanto elevata anche per l'altro tasso di cambio (Dollaro-Euro). Molto interessante

è anche la concordanza tra i segni delle correlazioni tra le due reti ricorrenti. Sembra che le reti ricorrenti risultino essere più performanti per l'individuazione di pattern di dipendenza, probabilmente a causa della loro elevata accuratezza.

Conclusioni

L'impiego degli *Shapley-Value* (SV) è un ottimo metodo di interpretabilità di qualsiasi modello di *machine* e *deep learning*. Permette di individuare quelle variabili che apportano un maggior contributo nella nostra previsione. Il metodo risulterà utile solo se prima viene effettuata un'adeguata analisi per la definizione del modello, le cui performance possono essere analizzate attraverso l'impiego delle metriche S.A.F.E. e dello zonoide di Lorenz. Attraverso gli *Shapley-Lorenz value* (SLV) è poi possibile effettuare dei confronti diretti tra i risultati di diversi modelli, senza la necessità di riportare i risultati nella stessa scala.

Il problema è che il processo è alquanto pesante computazionalmente. Ad esempio con cinque regressori è stato necessario calcolare trentuno previsioni, in relazione a tutte le possibili combinazioni di variabili esplicative, per ogni tipologia di modello.

Nonostante questa problematica, il metodo è utile in differenti contesti e facilmente interpretabile con l'impiego degli *swarm plot*. Si ipotizzi il caso di una banca che impiega un modello di rete neurale per classificare se il cliente possa acquisire un mutuo in base a una serie di variabili. Attraverso gli SLV è possibile individuare quali regressori generino un contributo più forte nella

decisione finale della rete. Attraverso gli SV possiamo, invece, analizzare il caso specifico per quell'i-esimo cliente e spiegargli quale aspetto abbia maggiormente contribuito per avergli concesso, o no, il mutuo.
Stessa situazione anche per contesti non economico-finanziari. Nel settore medico si possono analizzare, ad esempio, quali siano i fattori principali (*età, sesso, patologie specifiche, ecc...*) per i quali un virus possa risultare mortale per un individuo.

Bibliografia

Aldasoro, I., Gambacorta L., Giudici, P., Leach, T. (2022): *"The drivers of cyber risk"*, Journal of Financial stability.

Alex Sherstinsky (2020), *"Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network"*, *"Physica D: Nonlinear Phenomena"*, Elsevier journal , Volume 404.

Cao, L.: *"AI in finance: Challenges, techniques and opportunities"*. Technical report, University of Sidney (2021).

Comitato di Basilea (2021), *"Calls for improved cyber resilience, reviews climate-related financial risks and discusses impact of digitalisation"*.

Commissione europea (2019), *"Ethics guidelines for trustworthy AI"*.

Commissione europea (2020), *Libro bianco sull'intelligenza artificiale - Un approccio europeo all'eccellenza e alla fiducia*.

Commissione europea (2021), *"Regulatory Framework on Artificial Intelligence"*.

Comunicazione della Commissione al Parlamento Europeo (2018), al Consiglio, al Comitato Economico e Sociale Europeo e al Comitato delle Regioni: Piano coordinato sull'intelligenza artificiale.

Euro Tech Conseil (2022), *"Fintech vs Banks: What's the Difference?"*.

Giovanni Pascuzzi (2020), *"Il diritto dell'era digitale"*, quinta edizione Pandora Campus, ilMulino.

Giudici P., Raffinetti E. (2020): *"Lorenz Model Selection, Journal of Classification"*, 37(3), 754-768.

Giudici P., Raffinetti E. (2021): *"Shapley-Lorenz eXplainable Artificial Intelligence"*, Expert Systems With Applications, 167, 114104.

Giudici P., Raffinetti E.(2023): *"SAFE Artificial Intelligence in Finance"*, Finance Research Letters, 104088.

Jeff Orlowski (2020), docufilm *"The Social Dilemma"*, Netflix.

Joshua Starmer (2022), *"Recurrent Neural Networks (RNNs)"*, YouTube, StatQuest.

Joshua Starmer (2022), *"Long Short-Term Memory (LSTM)"*, YouTube, StatQuest.

Michael Phi (2018), *"Illustrated Guide to LSTM's and GRU's: A step by step explanation"*, Towards Data Science.

Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar (2018), *"Introduction to Data Mining"*, seconda edizione, *Global Edition*.

Rob J Hyndman, George Athanasopoulos (2021), *"Forecasting: Principles and Practice"*, terza edizione, *Monash University*, Australia.

Ruey S. Tsay (2005), *"Analysis of Financial Time Series"*, seconda edizione, *Wiley-Interscience*, *University of Chicago*.

Shoshana Zuboff (2019), *"The Age of Surveillance Capitalism: the fight for a human future at the new frontier of power"*, in *"PublicAffairs"*, New York.

Triebe, O., Laptev, N., Rajagopal, R.Z., Fishwick, P.: *"Ar-net: A simple auto- regressive neural network for time-series"*. Technical report, Stanford University.

Sitografia per la raccolta dei dati

FRED, Economic Data, *"Coinbase Bitcoin (CBBTCUSD)"*, Economic Research, Units: U.S. Dollars, frequenza giornaliera.

Investing.com, *"S&P 500 (SPX)"*,S&P 500 Historical Data, Price, frequenza giornaliera.

Investing.com, *"USD/CNY - US Dollar Chinese Yuan"*, USD/CNY Historical Data, Price, frequenza giornaliera.

Investing.com, *"USD/EUR - US Dollar Euro"*, USD/EUR Historical Data, Price, frequenza giornaliera.

Yahoo Finance, *"Crude Oil Mar 24 (CL=F)"*, NY Mercantile - NY Mercantile Delayed Price. Currency in USD, Adj. Close Price, frequenza giornaliera.

Nasdaq, *"Gold Apr 2024 (GC:CMX)"*, GC:CMX Historical Data, Close Price, frequenza giornaliera.

Ringraziamenti

Desidero ringraziare:

La mia relatrice Maria Cristina Recchioni, per i suoi consigli, per la sua disponibilità e per avermi concesso l'occasione di conoscere i Professori Giudici Paolo e Raffinetti Emanuela.

Il Professor Giulio Palomba per avermi fatto appassionare al campo dell'econometria durante il mio percorso alla triennale e poi alla magistrale in relazione alle serie storiche. Grazie ai suoi consigli decisi di intraprendere il percorso in Data Science.

La mia famiglia per essermi stata vicina in tutti i momenti duri della nostra vita. Senza il loro sostegno non avrei mai potuto raggiungere questo importante traguardo!

I miei amici del liceo e dell'università per aver condiviso insieme tanti traguardi, momenti di svago e di aver sempre creduto in me.

Un particolare ringraziamento a mia sorella che mi ha sempre saputo consigliare in tutti i miei momenti di indecisione.

Ringrazio mio zio Venanzo per le mille chiamate al telefono per sapere come fosse andato ogni singolo esame che ho dovuto sostenere nel mio percorso di studi, per avermi sempre dato la forza di non mollare mai, per essere stato da sempre al mio fianco. So che sei il primo a festeggiare da lassù, un forte Grazie e abbraccio zio.