Università degli Studi di Salerno
Dipartimento di Informatica

Tesi di Laurea Magistrale in
Informatica

Analisi e contrasto delle dinamiche di diffusione delle fake news. Integrazione euristica di modelli agent-based e deep reinforcement learning

Relatore
Prof. Rocco Zaccagnino
Correlatore
Prof.ssa Delfina Malandrino

Candidato
Ciro Perfetto
Matr. 0522501054

Anno Accademico 2022-2023

Abstract

La diffusione delle *fake news* sui social media è un tema particolarmente sentito negli ultimi anni. Vari studi hanno cercato di affrontare il fenomeno. Possiamo essenzialmente identificare due grandi filoni: la *fake news detection* (basata su metodi di intelligenza artificiale *data-driven* per analizzare il contenuto delle notizie ed i relativi metadati) e il *fake news modeling* (basato su approcci *model-driven* che mirano a studiare la diffusione delle fake news a partire da modelli simulativi ad agenti).

In questo lavoro di tesi, si tenta di studiare il fenomeno delle fake news e i possibili meccanismi di contrasto combinando gli approcci *model-driven* e *data-driven*. Da un lato, costruiamo una simulazione ad agenti per modellare la diffusione delle fake news (con annessa creazione delle cosiddette *echo chambers*), dall'altro utilizziamo un *super-agent* basato su *Deep Reinforcement Learning* per imparare dai dati della simulazione ad effettuare azioni, intese come decisioni atte al contenimento della diffusione del fenomeno. I risultati di una sperimentazione preliminare hanno dimostrato che la tecnica proposta si presta bene al contrasto della diffusione delle fake news (riducendo la "viralità" della stesse fino al 55%).

Riteniamo che lo strumento possa essere un valido supporto per i ricercatori impegnati nella ricerca al contrasto delle fake news e più in generale, dell'*information disorder*, in quanto il sistema così realizzato rappresenta una tela per successivi esperimenti e studi che coinvolgono approcci ibridi *model-driven* e *data-driven*. Per quanto ci risulta, questo lavoro rappresenta uno dei primi tentativi di studiare il fenomeno delle fake news attraverso il duplice approccio *model-driven*/*data-driven*.

Questa tesi è stata sviluppata in EDGE RESEARCH LAB

Indice

Capitolo 1

Introduzione

La popolarità dei social network, negli ultimi anni ha permesso di rivoluzionare la generazione e la distribuzione di informazioni. Al contempo, la facilità nell'accesso alle informazioni è un terreno fertile per la diffusione di disinformazione. La diffusione virale di fake news ha delle serie implicazioni sul comportamento e sulle opinioni del pubblico, rischiando di mettere in pericolo anche i processi democratici. Limitare quindi l'impatto negativo della disinformazione attraverso il riconoscimento preventivo e il controllo della diffusione è una delle sfide principali che oggi affrontano i ricercatori. Nonostante il fenomeno delle fake news non sia recente, può sorgere la domanda rispetto a cosa abbia suscitato l'interesse pubblico negli ultimi periodi. Il problema principale è che esse possono essere create e pubblicate online in modo più veloce ed economico rispetto ai media tradizionali, come televisione o giornali. La popolarità dei social media copre un ruolo essenziale in questa crescita nell'interesse dell'argomento, poiché la tendenza a reperire informazioni online ha come effetto quello di creare gruppi di persone che condividono la stessa opinione ("*echo chamber*") e che a loro volta consentono di rinforzare la diffusione di una notizia, vera o falsa che essa sia. Poiché le fake news tendono a diffondersi in maniera più rapida e ampia rispetto alle notizie vere, è possibile trovare delle correlazioni tra le *echo chamber* e la diffusione delle fake news. Tale collegamento risulta però difficile da dimostrare vista la complessità delle dinamiche sociali. Per analizzare questo fenomeno possono essere utilizzati diversi approcci: *(i) model-driven*, sistemi che simulano gli aspetti dei comportamenti sociali e che possono contribuire alla comprensione di questi ultimi [1, 2, 3]; *(ii) data-driven*, sistemi basati su intelligenza artificiale (machine learning, natural language processing e text mining) per l'individuazione sui social media di notizie false [4, 5, 6, 7, 8, 9].

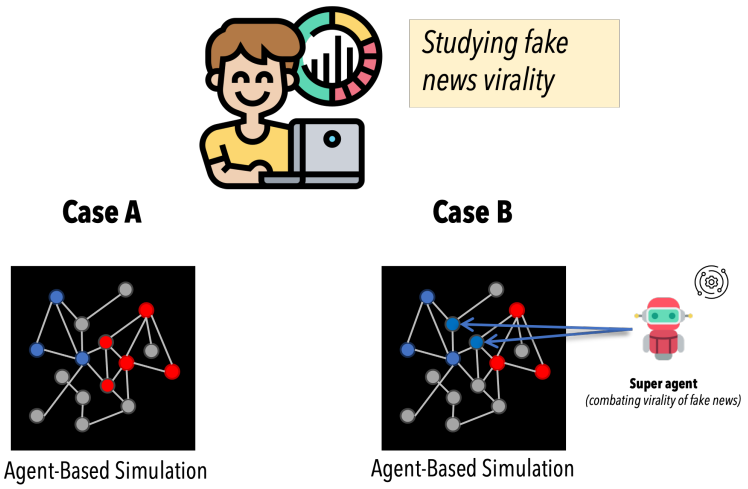


Figura 1.1: Visual abstract di questo lavoro di tesi. ***Case A****: studio della diffusione delle fake news in presenza di una echo chamber (approccio "model-driven");* ***Case B****: studio della diffusione delle fake news in presenza di una echo chamber (approccio "model-driven") e di un super agente basato su Deep Reinforcement Learning atto a limitarne la viralità (approccio "data-driven").*

In questo lavoro si ibridano gli approcci (vedi Figura 1.1). In una fase iniziale è stata creata una simulazione ad agenti al fine di analizzare le dinamiche di diffusione di una fake news all'interno di un social network. Nella simulazione è stata introdotta la presenza di una *echo chamber*, che permette il diffondersi di una fake news. In seguito, è stato studiato l'impatto di un *super agente* in grado di analizzare lo stato del network e di intervenire, durante l'avanzamento della simulazione, con azioni che permettono di contrastare la diffusione della disinformazione. Il super agente, che può rappresentare un amministratore di un social network o un ente pubblico che vuole prevenire il dilagarsi di notizie non veritiere, si basa su un meccanismo di intelligenza artificiale che tramite osservazioni e ricompense apprende quale sia la migliore azione di contenimento da compiere in un determinato momento sulla base dello stato del network. La tecnica utilizzata per addestrare il super agente è quella del Deep Reinforcement Learning (DRL).

Struttura della tesi. Il resto di questo lavoro è organizzato come segue:

- Capitolo 2: presentazione dell'oggetto di indagine e dei lavori in letteratura scientifica che hanno affrontato il fenomeno delle fake news;

- Capitolo 3: panoramica sui concetti di base per comprendere il lavoro svolto, in particolare sulla simulazione basata ad agenti e il Reinforcement Learning (RL);
- Capitolo 4: descrizione della strategia ibrida (*model-driven* e *data-driven*) utilizzata per studiare e contrastare il fenomeno delle fake news, con particolare attenzione al modello simulativo (alla base dello studio sulla diffusione) ed il modello di Reinforcement Learning (alla base dello studio sulle strategie di contrasto) utilizzato.
- Capitolo 5: dettagli sugli esperimenti condotti e analisi approfondita dei risultati ottenuti.
- Capitolo 6: osservazioni finali, limitazioni del lavoro e sviluppi futuri.

Capitolo 2

Il fenomeno delle fake news

In questo capitolo, verranno esposte le conoscenze utili a comprendere il lavoro di tesi. In particolare, verrà descritto il problema delle fake news da un punto di vista scientifico (Sez. 2.1), da un punto di vista sociale e psicologico (Sez. 2.2), lo stato dell'arte su come si tenta di contrastare la disinformazione (Sez. 2.3) e quali sono i limiti degli approcci attuali. (Sez. 2.4)

2.1 Le fake news: da problema scientifico a problema di policy

Con la crescita del web come mezzo principale d'informazione, non è strano notare una crescita nella disponibilità di notizie online. Il diffondersi delle notizie in questo modo ha dei riscontri sia positivi che negativi. L'accesso alle notizie è diventato più rapido ed economico. Questo flusso enorme di informazioni può portare a dei problemi. In primis, l'assenza di un controllo editoriale, per la qualità dei contenuti che circolano sulle piattaforme web, ha determinato l'assenza di controlli sull'originalità e qualità. Al contempo, la crescita del consumo di notizie online non è stata seguita parallelamente con quella dei media [10]. Negli ultimi anni, quindi, si è cercato di trovare un modo per distinguere le notizie vere dalle fake news.

Questo problema ha portato alla creazione di alcuni fenomeni online:

- *Echo Chamber* [11]: si creano quando più persone insieme condividono la stessa opinione e creano una sorta di rete che è in grado di aumentare la diffusione di una notizia. La causa di questo fenomeno può essere dovuta a bias cognitivi, come ad esempio, il *Confirmation bias*,

che spinge le persone a cercare l'affermazione della propria linea di pensiero [12].

- *Opinioni false non intenzionali*: chiunque può esprimere la propria opinione su internet e ci sono persone che possono tentare di massimizzare i propri follower adottando delle strategie di comunicazione. In questo caso comunicare delle opinioni non vere può causare disinformazione anche se in principio l'intenzione non era quella [10].

- *Opinioni false intenzionali*: visto che molti social non adottano filtri sui contenuti che può condividere una persona, si potrebbe cercare di manipolare l'opinione del pubblico con notizie false creando scompiglio. Questo perché anche se una notizia è falsa questa si diffonde più velocemente all'interno della rete.

- *Disinformazione politica*: le notizie possono essere usate per manipolare il pubblico anche in ambito politico e mirano a screditare avversari politici o a deviare il risultato delle elezioni [10].

Non tutti questi fenomeni però dovrebbero essere censurati portando ad un blocco del circolare delle notizie. La differenza delle opinioni è sempre esistita ed è alla base dei dibattiti politici. Opinioni diverse spesso sono anche il riflesso di preferenze personali o anche di diversi scenari culturali.

La proliferazione delle fake news ha portato allo sviluppo di una serie di policy, da quelle tecniche/tecnologiche, fino a quelle governative consistenti in nuove leggi che impongono multe alla condivisione di notizie false. La falla di questo approccio è che sorge il problema della libertà di espressione.

Filtrare tutte le opinioni che non si basano su fatti verificati, quindi, non è la soluzione al problema [13].

2.2 Le fake news come fenomeno sociale e psicologico

Ci sono studi scientifici che si concentrano sulle ragioni psicologiche alla base della diffusione delle fake news sui social media; infatti, ci sono molti bias cognitivi che di solito guidano il giudizio di una persona, ma che possono sviarla quando questa viene esposta a disinformazione [14]. La disinformazione lavora attraverso una serie di agganci cognitivi che rendono l'informazione più interessante, scatenando una reazione psicologica nel lettore. Contengono più novità rispetto alle notizie regolari e tendono a riferire le informazioni in maniera più semplice, cosa che può attrarre l'attenzione di una persona

che non presta molta attenzione [15]. Un altro fattore che determina il successo della disinformazione è la ripetuta esposizione alla stessa notizia, visto che la ripetizione incrementa la probabilità di essere d'accordo con essa. Inoltre, se l'algoritmo di visualizzazione delle notizie mette in evidenza il contenuto condiviso da una persona seguita da un utente, questo effetto può crescere ulteriormente, sfociando nel fenomeno delle echo chamber (vedere Sez. 2.1). In altre parole, *se un utente è esposto ripetutamente a un opinione, la probabilità di essere influenzati è alta.*

Altre ragioni dovute alla diffusione della disinformazione, possono essere i comportamenti sociali, quali ad esempio:

- *Bandwagon effect* [16]: tendenza ad allinearsi con l'opinione di quella che è percepita come una grande massa di individui; l'opinione del gruppo è considerata come l'opinione pubblica, mentre, le notizie discordanti sono percepite come degli attacchi ostili verso il pubblico.

- *Normative social influence theory* [17]: l'influenza dell'identità sociale ricopre un ruolo importante nella percezione delle notizie, portando le persone ad adeguarsi al comportamento e alle idee dei gruppi sociali a cui appartengono; queste percezioni sbagliate, sono sostenute da bias cognitivi ereditati dai comportamenti sociali.

- *Third-person effect* [18]: tendenza a credere che la disinformazione influisca più sugli altri gruppi rispetto a quello di cui fa parte l'individuo.

- *Naive realism* [19]: tendenza di un individuo a assumere che la realtà che percepiscono sia oggettiva e fattuale.

- *False consensus effect* [20]: tendenza degli individui a credere che le loro opinioni siano ampiamente accettate; sotto questo aspetto, le notizie discordanti possono essere classificate come tentativi minatori che vogliono mettere in discussione un fatto ampiamente diffuso.

Tutti questi comportamenti sociali, sono dei fattori chiave per la formazione dei cluster e echo chamber nei social network.

Infine, ci sono molti bias di un individuo che possono rafforzare il travisare delle informazioni [14]:

- *Confirmation bias*: un utente tende a concentrarsi su un'informazione che conferma le sue opinioni pregresse, invece di notizie che le contraddicono [21].

- *Attentional bias*: se una persona si convince di un'informazione scioccante può iniziare a notare più spesso quell'avvenimento [22].
- *Selective exposure*: le persone si affidano a notizie che sono in linea con le loro opinioni [23].
- *Congruence Bias*: le persone raramente cercano notizie che smentiscono le loro convinzioni [24].
- *Belief bias*: alcuni titoli possono risultare come più impattanti nonostante il fatto che gli argomenti siano scarni, se la conclusione è congruente con le aspettative che si sono create [25].
- *Emotional bias*: può distorcere la percezione di un'argomentazione, per esempio, può portare una persona a non voler accettare delle informazioni che possono creare nervosismo [26].

I bias quindi, insieme alle strutture delle reti sociali, possono essere un ulteriore fattore che porta alla diffusione delle fake news. È importante tener conto di questi fenomeni psicologici, in quanto possono essere utili alla comprensione di alcune dinamiche che possono crearsi una rete.

2.3 Il problema fake news in letteratura: studio della diffusione e contrasto

In questa sezione si offre una panoramica sulla letteratura scientifica relativa alle fake news, ai modelli di diffusione e propagazione (Sez. 2.3.1), e al contrasto del fenomeno delle notizie false (Sez. 2.3.2).

2.3.1 Studio della diffusione delle fake news

Viene mostrato ora, nella Tabella 2.1, come in letteratura si è affrontato il problema delle fake news, cercando di analizzare pattern che possono emergere dallo studio delle diffusione delle notizie, utilizzando delle simulazioni ad agenti. Per semplicità, si fornisce un elenco puntato in cui ogni elemento riguarda un determinato lavoro di ricerca presente in letterature e, per ciascuno, il testo offre il confronto (evidenziando similitudini e differenze) con il presente elaborato:

- [27]: una differenza sostanziale che è presente in questo lavoro è che, nella simulazione ad agenti sviluppata, gli agenti prendono decisioni

in base ad una rete *Bayesiana*, la quale determina il cambio di opinione/stato. La rete *Bayesiana* è condizionata da due probabilità: *(a)* la probabilità che la fonte della notizia sia di un esperto e *(b)* la probabilità che la notizia provenga da una fonte affidabile. Entrambe queste probabilità sono però prese in considerazione dal punto di vista dell'utente e non corrispondono a una probabilità oggettiva. Un'altra caratteristica è quella che gli agenti partono tutti con le stesse possibilità comportamentali e cognitive. Si studiano quindi gli effetti di una echo chamber con degli agenti aventi queste caratteristiche.

- [28]: anche in questo lavoro gli agenti hanno una rete *Bayesiana* che li governa. In particolare, per esplorare il fenomeno delle echo chamber, viene osservato se gli agenti possono entrare a far parte della echo chamber dopo ripetute iterazioni con altri agenti nella rete. Sono presenti solo due opinioni, che sono una l'opposta dell'altra, ed è presente una verità globale obiettivo ($\mu_{true} = 0.5$). considerando che ogni agente parte con un valore diverso rispetto all'opinione globale e ha un grado di incertezza. A ogni passo della simulazione, gli agenti cercano altri agenti che hanno dei valori di opinione in prossimità dei loro. In base al grado di incertezza, ogni agente può scegliere se accettare o meno l'informazione. Accettando l'informazione questa fa variare l'opinione dell'agente rispetto alla verità globale in concordanza col modello *Bayesiano*. Più è alta la certezza su un'opinione più è difficile che un agente si lasci influenzare dall'opinione che riceve.

- [29]: il lavoro si focalizza su come taglie differenti del numero di media e i pattern di interazione del sistema di informazione, possano influenzare il dibattito collettivo e quindi la distribuzione di un'opinione. Viene costruita una simulazione ad agenti che studia le dinamiche delle opinioni, tenendo conto che esistono sia i media che i gossip come meccanismi separati di interazione. I media rispondono agli editori – e quindi tendono a formare gruppi polarizzati per diffondere le informazioni – mentre i chi vuole diffondere i gossip, interagisce con i propri vicini e con i media utilizzando un *bounded confidence model*, che indica la distanza tra l'opinione di un utente e la notizia che sta ricevendo, determinando i meccanismi di influenza.

Buona parte del lavoro di tesi si ispira al modello utilizzato nel paper intitolato *"Echo chambers and viral misinformation: Modeling fake news as complex contagion"* [12], pubblicato sul *journal Plos ONE*. Tale lavoro è stato scelto come punto di partenza per la sua riproducibilità. Sebbene l'autore

Riferimento	Obiettivo	Metodologia	Punti Principali
[27]	Studiare l'emersione delle *echo chamber* in una rete sociale in un ambiente in cui ci sono opinioni contrastanti (una opinione può essere intesa come una fake news).	Simulazione Agent-based con grafi di varie dimensioni; test di robustezza con agenti bizantini; misurano il tasso di credenza ad una notizia.	*(i)* Uso di grafi random; *(ii)* gli archi del grafo non sono direzionati; *(iii)* gli agenti prendono decisioni binarie utilizzando reti Bayesiane analizzando la probabilità che una notizia sia riportata da un esperto e che sia vera; *(iv)* modellano un grado di apprendimento dell'agente.
[28]	Studiare il fenomeno delle *echo chamber*in una rete sociale costituita da agenti. In particolare si vuole evidenziare come la formazione delle *echo chamber* sia dovuta più alla struttura della network stessa piuttosto che dagli agenti che ne fanno parte.	Modello avente agenti Bayesiani che aggiornano le proprie "opinioni" senza bias; tutti gli agenti assumono che ciò che credono sia lo stato reale del mondo e tutti gli agenti riceventi hanno completa fiducia nella trasmissione dell'informazione; uso di vari parametri per differenziare le simulazioni; test di robustezza con agenti bizantini.	*(i)* Uso di reti random e scale-free; *(ii)* gli archi del grafo non sono direzionati; *(iii)* operazioni di *pruning* per stabilizzare l'opinione di un agente; *(iv)* Gli agenti prendono decisioni binarie utilizzando reti Bayesiane, aventi come parametro l'opinione iniziale dell'agente stesso.
[12]	Studiare la possibile relazione tra le *echo chamber* e la diffusione virale della disinformazione tramite la simulazione di una rete sociale; rappresentare un sistema complesso.	Simulazione agent-based; cluster di agenti formati tramite un parametro di polarizzazione della rete; diffusione delle notizie dovuta al superamento di una soglia contenuta sui nodi.	*(i)* Uso di reti Erdős-Rényi; *(ii)* gli archi del grafo non sono direzionati; *(iii)* calcolo del parametro di polarizzazione della rete e dell'opinione di una notizia per studiarne la viralità; *(iv)* gli agenti non utilizzano una rete bayesiana ma hanno un parametro di treshold per accettare o rifiutare una notizia ricevuta.
[29]	Creare un modello che analizza come la struttura dei social media può influenzare l'opinione degli utenti.	Modellare l'informazione su scenari differenti(Facebook e Twitter); utilizzo di due tipi di network che interagiscono: i media e i gossiper; introduzione della competizione(polarizzazione) per la diffusione dell'informazione da parte dei media; modello agent-based.	*(i)* Utilizzo di modelli matematici per le dinamiche delle opinioni; *(ii)* uso di reti con topologie scale-free e small-word; *(iii)* i gossiper e i media interagiscono in modo diverso: i primi selezionano in modo casuale a chi tramettere l'informazione, mentre i secondi selezionano i vicini con il maggior numero di follower; *(iv)* gli agenti non utilizzano una rete bayesiana ma hanno un parametro di treshold per accettare o rifiutare una notizia ricevuta.
	Creare una rete neurale che riesca a fermare il diffondersi delle fake news all'interno di una rete sociale; la rete neurale, attraverso il DRL, si addestra per riuscire a trovare l'azione giusta da fare dato uno stato della rete.	Simulazione basata ad agenti da cui estrarre i dati su cui deve essere addestrato la rete neurale; Vengono utilizzate tecniche di network analysis per identificare al meglio il punto della rete in cui agire per fermare l'andamento delle fake news.	*(i)* Presenza di un super-agent all'interno della rete; *(ii)* uso di reti random; *(iii)* utilizzo del DRL per addestrare il super-agent; *(iv)* i nodi hanno una soglia che permette di stabilire verso quale opinione sono più inclini (Θ); *(v)* i nodi possono avere un'opinione di tipo A o B, oppure rimangono neutrali.

Tabella 2.1: Sommario dei confronti con lo stato dell'arte sui modelli che analizzano il diffondersi delle notizie false all'interno di una rete

non avesse rilasciato nessun codice, il manoscritto riporta tutti i dettagli utili per altri ricercatori al fine di riprodurre l'ambiente, gli esperimenti e i risultati ottenuti. Qui di seguito, in particolare in Tabella 2.2, si analizzeranno le differenze tra il modello costruito nel lavoro di tesi e quello in [12].

Tabella 2.2: Tabella riassuntiva degli aspetti chiave del lavoro "Echo chambers and viral misinformation: Modeling fake news as complex contagion", Petter Törnberg 2018 [12] ed il lavoro di tesi, evidenziando le differenze.

[12]	**Questo lavoro di tesi**
Obiettivo / Domanda di ricerca Questo paper utilizza un modello di simulazione di (social) network per studiare una possibile relazione tra le echo chambers e la diffusione virale della disinformazione / fake news. Gli utenti della rete possono avere una tra due opinioni.	**Obiettivo / Domanda di ricerca** Utilizziamo un modello di simulazione ad agenti per studiare la diffusione virale delle fake news in un social network in presenza di una echo chamber e sfruttiamo un metodo di *DRL* per analizzare i dati di queste simulazioni e sviluppare strategie di contenimento / contrasto alla diffusione virale delle fake news. Gli utenti della rete possono avere una tra tre opinioni.
Dettagli degli agenti • Gli agenti hanno due opinioni, A (neutro / grigi) e B (pro-fake news / arancioni). • Gli agenti hanno una soglia di attivazione (*creduloneria*). La *creduloneria* è più alta all'interno della echo chamber.	**Dettagli degli agenti** • Gli agenti hanno tre opinioni, A (pro vera news / blu), B (pro-fake news / arancioni), C (neutro / grigi) • Gli agenti hanno una soglia di attivazione (*creduloneria*). La *creduloneria* è più alta all'interno della echo chamber. • Gli agenti hanno un margine di *creduloneria* (*opinion-metric*), cioè non si attivano al semplice superamento della soglia. • Esiste un *super-agent* esterno, che ha la capacità di osservare l'andamento del network e della diffusione delle notizie false ed effettuare delle operazioni di contrasto / contenimento individuate in letteratura.

Continued on next page

Tabella 2.2 – ***continued from previous page***

[12]	**Questo lavoro di tesi**
Modello di diffusione • Un agente viene contagiato dalla fake news se la frazione dei nodi vicini è maggiore della propria soglia di attivazione (*creduloneria*). Modella il "mi hai convinto".	**Modello di diffusione** Un agente viene contagiato dalla fake news o dalla notizia vera se (entrambe le condizioni devono verificarsi): • la frazione dei nodi vicini (blu o arancioni rispettivamente) è maggiore della propria soglia di attivazione (creduloneria); • se ha superato il margine di credu-loneria per cui è possibile cambiare opinione / essere contagiati; • Se num(arancioni) == num(blu), allora non cambio opinione né mi convinco di qualcosa e "resto dove sono". • Modella il "mi hai convinto". Il margine di creduloneria di un agente viene modificato nella direzione della fake news o della vera news se la frazione dei nodi vicini (blu o arancioni rispettivamente) è maggiore della propria soglia di attivazione (creduloneria). Modella il "mi stai convincendo".

Continued on next page

Tabella 2.2 – ***continued from previous page***

[12]	**Questo lavoro di tesi**
Dettagli sull'ambiente • Il social network modellato ha la struttura *Erdős–Rényi* (media connessioni K). • Il network ha N nodi. • Una percentuale c di nodi nel network viene scelta per appartenere al cluster oggetto di analisi • Si scelgono una percentuale di archi che connettono un individuo nel cluster ed uno fuori; si eliminano tali link e si aggiunge uno stesso ammontare di link interni al cluster, aumentando quindi le connessioni interne e riducendo le esterne. • Inizialmente, si seleziona un nodo nel cluster e i suoi vicini; questi nodi diventano arancioni.	**Dettagli sull'ambiente** • Il social network modellato ha la struttura a scelta tra *Erdős–Rényi* (media connessioni K), con *Preferential Attachment* e con *Small World*. • Il network ha N nodi. • Una percentuale c di nodi nel network viene scelta per appartenere al cluster oggetto di analisi • Si scelgono una percentuale di archi che connettono un individuo nel cluster ed uno fuori; si eliminano tali link e si aggiunge uno stesso ammontare di link interni al cluster, aumentando quindi le connessioni interne e riducendo le esterne. • Inizialmente, si seleziona un nodo nel cluster e i suoi vicini; questi nodi diventano arancioni. Inoltre, si seleziona un altro individuo (a caso nel network) ed i suoi vicini; questi nodi diventano blu.
Dettagli sulla echo chamber Nel network creato, si intende una echo chamber come un insieme di utenti caratterizzati da due proprietà: opinione e polarizzazione della rete. • Polarizzazione delle opinioni significa che, in relazione a una data domanda, sono più inclini a condividere opinioni simili. • La polarizzazione della rete significa che sono più densamente connesse tra loro che con la rete esterna.	**Dettagli sulla echo chamber** Nel network creato, si intende una echo chamber come un insieme di utenti caratterizzati da due proprietà: opinione e polarizzazione della rete. • Polarizzazione delle opinioni significa che, in relazione a una data domanda, sono più inclini a condividere opinioni simili. • La polarizzazione della rete significa che sono più densamente connesse tra loro che con la rete esterna.

Continued on next page

Tabella 2.2 – ***continued from previous page***

[12]	**Questo lavoro di tesi**
	Ulteriori dettagli l'influenzabilità / margine di creduloneria di un agente blu è basso, cioè serviranno più "sforzi" per convincerlo a cambiare idea.

2.3.2 Contrasto alle fake news

Parte della letteratura di riferimento, si è concentrata sul contrasto alle fake news sfruttando metodi di detection *data-driven*, e quindi tecniche basate sull'Intelligenza Artificiale. Tali metodi sono in grado di riconoscere, tramite tecniche di Natural Language Processing (NLP) e Machine o Deep Learning, qual è la probabilità che una notizia, presa da un giornale sul Web o da Social Media, sia una fake news, ovvero una notizia che non riporta la verità dei fatti realmente accaduti. In questo genere di lavori, si parte dall'assunto che sia possibile distinguere una notizia falsa dall'analisi di *(a)* il testo della notizia, il titolo ecc, *(b)* metadati associati ad essa (autore, URL, ecc.). In questa prospettiva si sono mossi un gran numero di ricercatori in tutto il mondo che hanno dato alla luce una serie di meccanismi più o meno sofisticati, basati principalmente su NLP e *Deep Learning*. In questo lavoro, si analizzano alcuni dei contributi più recenti e interessanti. I ricercatori della *Siracuse University* in [30] hanno effettuato una rassegna dei lavori fino al 2020 che si sono concentrati sul contrasto alle fake news in linea con il paradigma precedentemente elucidato. Dalla loro analisi, il riconoscimento delle fake news si basa su quattro metodologie:

1. *Metodi basati su una base di conoscenza*: i quali riconoscono le fake news verificando se i fatti contenuti nelle notizie, cioè il testo, sono coerenti con la conoscenza generale, la verità.

2. *Metodi basati sullo stile*: che riguardano come le fake news sono scritte, ad esempio con emozioni portate agli estremi.

3. *Metodi basati sulla propagazione*: che individuano le fake news in riferimento a come si diffondono le notizie online.

4. *Metodi basati sulla fonte*: che individuano le fake news investigando sulla credibilità delle fonti delle notizie a vari stadi (creazione, pubblicazione, diffusione sui Social Media).

I *metodi basati su una base di conoscenza* si dividono in metodi per controllare le notizie *manualmente* e *automaticamente*. Quelli manuali a loro volta si suddividono in:

- Notizie *verificate da esperti*: che appunto si basano sulla verifica di esperti per determinare se le notizie sono veritiere. Alcuni siti di notizie, per esempio, integrano dei gruppi di esperti che verificano le notizie e provvedono quindi a definire una conoscenza di base per il riconoscimento delle fake news.

- Notizie *verificate dalla massa*: basata dalle notizie prese da alcuni *marketplace* basati sulla popolazione comune come *Amazon Mechanical Turk*. Questa tecnica a differenze di quella basata sugli esperti è molto più scalabile ma non è altrettanto affidabile.

Quindi in generale la verifica manuale delle notizie non scala e per risolvere questo problema sono state introdotte tecniche di NLP e *Machine Learning* (ML).

La verifica automatica può essere suddivisa in due stadi: l'estrazione delle notizie, quindi la creazione della *Knowledge Base* (KB), e la verifica delle notizie. Un fatto può essere identificato come una tripla di soggetto, predicato e oggetto (SPO). Tramite le SPO si possono costruire dei *knowledge graph* che hanno come nodi i soggetti o gli oggetti e come archi le relazioni, ovvero i predicati. Quindi in generale per verificare le notizie, abbiamo bisogno di confrontare la conoscenza estratta dalle news, per esempio le *SPO*, con i fatti. Successivamente, la strategia prevede di calcolare la possibilità che l'arco del predicato esista dal nodo soggetto al nodo oggetto nel grafo. Un esempio, può essere un nodo soggetto con "*Trump*", un nodo oggetto con "*presidente*" e un arco che li collega con predicato "*professione*".

I *metodi basati sullo stile* si basano sull'intuizione e l'assunzione che chi scrive fake news, lo fa in un modo tale da far credere a chi legge che ciò che è scritto è reale. Quindi tramite un set di caratteristiche quantificabili, ad esempio tramite ML, si possono distinguere le fake news dalle vere. Queste *feature* possono essere raggruppate in *feature* testuali e *feature* visive rappresentati da testo e immagini rispettivamente. In generale per il testo viene riconosciuta la semantica, il lessico, il discorso e la sintassi mentre per le immagini ci sono studi ancora in corso per distinguere se un immagine è stata creata appositamente, utilizzate reti neurali per lo scopo. Per il testo sono stati riconosciuti dei pattern nelle fake news, ad esempio l'informalità, la diversità dei verbi utilizzati, la soggettività e l'espressione di emozioni,

per le immagini, invece, è stato rilevato che spesso, per attrarre il pubblico vengono, usate immagini invitanti ma irrilevanti all'avvenimento in sé per sé.

I *metodi basati sulla propagazione* utilizzano delle strutture chiamate *News Cascade* (notizie a cascata). Sono delle strutture ad albero che catturano direttamente la propagazione di alcuni articoli di giornale sui social network. Il nodo radice rappresenta l'utente che per primo ha diffuso la notizia e gli altri nodi rappresentano gli utenti che hanno a loro volta condiviso l'articolo dopo che è stato pubblicato dal loro nodo padre. Una cascata di notizie può essere rappresentata in termini numerici, ad esempio, i passi, quindi gli *hop*, che la notizia ha effettuato oppure il numero di volte che è stata condivisa, basata sul tempo. Inoltre, nel modello a cascata possono essere presenti altre informazioni sugli utenti, come ad esempio, se supportano o si oppongono alla fake news, il loro profilo di informazione, post precedenti e così via. Quindi si arriva a dover classificare la cascata di una fake news come vera o falsa. Per farlo si può usare il ML tradizionale oppure reti neurali. Con un modello di *Machine Learning* tradizionale, le caratteristiche di un modello a cascata possono essere ispirati ai pattern di propagazione delle fake news osservati da studi empirici. Gli studi dimostrano che le fake news, si diffondono più velocemente e più ampiamente, rendendole più virali rispetto alle notizie normali. Mentre, le notizie reali impiegano tempo per diffondersi in profondità. La strategia con il ML si basa quindi, sul calcolo di similarità attraverso dei grafi, confrontando varie cascate di notizie. Per i modelli basati su reti neurali vengono usati dei classificatori e dei modelli appositamente addestrati tramite funzioni di costo. Le network, su cui poi vengono costruiti i grafi di propagazione, possono essere di tre tipi: omogenei, eterogenei e gerarchici:

- Network Omogenee: le reti omogenee contengono un solo tipo di nodo e un solo tipo di arco.
- Network Eterogenee: le reti eterogenee hanno più tipi di nodi o archi.
- Network gerarchiche: le reti gerarchiche, vari tipo di nodi e archi formano delle relazioni di tipo insieme/sotto-insieme, ad esempio una gerarchia.

Tramite i *metodi basati sulle fonti*, si possono determinare le fake news valutandone la credibilità della loro fonti, dove la credibilità è definita nel senso di qualità e affidabilità. Ci sono tre stadi in cui è suddiviso il ciclo di vita di una fake news: la creazione, la pubblicazione online e la propagazione sui social media. Le sorgenti includono:

1. la fonte che crea la notizia;
2. la fonte che pubblica la notizia;
3. le fonti che pubblicano le notizie sui social media;
4. le fonti che diffondono le notizie.

Si analizza ora la qualità e l'affidabilità di questi quattro punti, assumendo che i punti 1 e 2 siano combinati. Questi ultimi vengono uniti perché sono anche un parametro per quanto riguarda la valutazione indiretta di una fake news, per esempio, uno può considerare le notizie da fonti inaffidabili come fake news, sebbene queste possano comunque pubblicare notizie veritiere.

- Sono stati identificati dei pattern comuni di autori di notizie poco affidabili, che possono aiutare a valutare la credibilità di autori sconosciuti, esplorando le loro relazioni con altri autori o editori. La ricerca ha mostrato che le notizie pubblicate hanno un alto tasso di omogeneità nelle reti che formano.
- Chi pubblica notizie di solito lo fa sui propri siti web. Determinare quindi chi pubblica notizie inaffidabili, può essere ridotto a individuare siti non credibili. Per verificare la credibilità di un sito sono state sviluppate alcune tecniche, come ad esempio algoritmi di ranking dei siti web, che analizzano sia le caratteristiche dei contenuti dei siti e sia i collegamenti, utilizzando dei grafi per individuare lo spam.
- Un'ulteriore risorsa per determinare l'affidabilità di un sito, è la capacità di ottenere delle informazioni sulla credibilità o sul pensiero politico di chi pubblica notizie. Una risorsa è il sito *Media Bias/Fact Check*, che ha una lista degli editori e dei loro orientamenti politici. Un'altra risorsa è *NewsGuard*, che si affida a valutazioni di esperti che classificano una notizia sulla base di alcuni criteri.

Un'altra rassegna interessante è stata condotta dai ricercatori della *Tampere University* in collaborazione con l'*University of Denmkark* [31]. Lo scopo principale di questo lavoro, è quello di fornire una panoramica sui metodi già usati nel rilevamento delle fake news allo stato iniziale tramite ML, e quindi a mostrare lo stato dell'arte a cui si è giunti attualmente per migliorare le ricerche in quest'ambito. In particolare, oltre ad elencare una serie di strumenti utili al contrasto delle fake news, il lavoro si concentra su un aspetto degno di nota del mondo delle fake news: il *fact checking*. La

rapida diffusione di fake news ha portato i ricercatori ad automatizzare il processo di riconoscimento usando tecniche di ML e *Deep Neural Network*.

Sono state, inoltre, esaminate strategie per analizzare le informazioni in modo automatico. Per fare ciò sono state analizzate migliaia di affermazioni in relazione a cultura, storia, geografia e così via, utilizzando un grafo di conoscenza estratto da *wikipedia*. É stato trovato che le affermazioni vere ricevono un supporto maggiore a differenze di quelle false; si è giunti alla conclusione che applicare queste analisi a dati e conoscenze di scala maggiore, può portare a nuove strategie per analizzare automaticamente le notizie e le informazioni che contengono. In particolare, oltre all'analisi degli strumenti di rilevamento delle fake news, si esaminano due facciate relative al problema del *fact checking*:

- *Automatic fact checking*: gli approcci computazionali al riconoscimento delle notizie (*fact-checking*) sono considerati come la chiave per il contrasto della diffusione delle fake news. Questi approcci sono scalabili e efficaci nel valutare l'accuratezza di avvenimenti esposti in notizie dubbiose.

- *Affidabilità e credibilità*: la facilità dell'accesso alle notizie sui social media risulta in una grande quantità di contenuto che può essere acceduto. Sia chi consuma che chi condivide queste notizie deve anche controllare la credibilità di esse. Se un utente è interessato nel ricevere un'informazione, il suo compito principale è quello di controllare la credibilità, per esempio: osservando se altri utenti hanno condiviso l'informazione, accertandosi del giudizio di esperti e della credibilità della fonte.

Negli anni, particolare attenzione è stata rivolta al fenomeno del *clickbait* a cui è assegnata una certa rilevanza nel modo e nella velocità con cui si diffondono le notizie false. Tale fenomeno si lega ai motivi per cui sono stati sviluppati metodi per la rilevazione delle fake news *basati sullo stile*, di cui sopra. Le notizie false sono create intenzionalmente per ottenere un guadagno finanziario o politico piuttosto che per riportare affermazioni obiettivi; spesso, quindi, contengono un linguaggio supponente e provocatorio, creato come "acchiappaclic" (ovvero, per invogliare gli utenti a fare clic sul collegamento per leggere l'intero articolo) o creare confusione [32]. In [33] gli autori hanno sviluppato un tool chiamato *Lit.RL*, che automaticamente distingue i titoli clickbait da quelli non clickbait. Questo software si basa sull'utilizzo di tecniche NLP e ML raggiungendo alti valori di accuratezza utilizzando *Support Vector Machine*. *Lit.RL*, scritto in *Python*, classifica i testi in input

come clickbait o non clickbait (classificazione binaria). Gli autori hanno evidenziato come gli attributi chiave più importanti per determinare se un titolo è *clickbait* sono:

- la frequenza dei pronomi, visto che i titoli *clickbait* hanno un'alta frequenza di pronomi nelle frasi;
- la frequenza degli articoli determinativi;
- articoli che iniziano con dei numeri e propongono delle liste;
- Il numero delle parole effettive, che tende ad essere minore nei titoli *clickbait*;
- volgarità e imprecazioni.

2.4 Studio e contrasto alle fake news: limiti negli approcci attuali

Da quanto emerge dalle sezioni precedenti, sebbene si siano profusi numerosi sforzi, c'è ancora molta ricerca che deve essere fatta per studiare e contrastare il dilagante fenomeno delle fake news. Vediamo ora i limiti principali sia per gli approcci *model-driven* che *data-driven*:

- *model-driven*: essi analizzano solo la diffusione delle fake news. Speculano sulle possibilità di intervento per contrastare la disinformazione, ma non le implementano né fanno esperimenti a riguardo. Il focus in letteratura si è spostato principalmente sui meccanismi di diffusione, sul rendere più realistico lo scenario simulato, sul migliorare il modello simulativo in sé, più che nello sperimentare meccanismi di contrasto.
- *data-driven*: l'utilizzo di AI per riconoscere le notizie è solo una parte di tutte le risorse necessarie per il contrasto efficace della disinformazione. Infatti, se le tecniche di riconoscimento non vengono integrate insieme ad altri strumenti, non verranno sfruttate mai a pieno. Inoltre, sorvolando sul fatto che gli sforzi della ricerca sul riconoscimento di notizie false prendono principalmente di mira il fenomeno del *clickbaiting*, non è stato ancora progettato un sistema capace di riconoscere fake news prescindendo dal dominio specifico, cioè qualcosa di "*general-purpose*", bensì i risultati mostrano dei sistemi accurati solo in scenari scelti, es. campagne politiche per le elezioni tra due esponenti opposti.

Nella ricerca [30], sono stati esposti alcuni metodi che suggeriscono come migliorare il riconoscimento delle fake news. Questi sono:

- il riconoscimento di fake news non tradizionali;
- riconoscere le fake news negli stadi iniziali;
- identificare il contenuto che è più rilevante;
- *Cross-Domain*, cioè l'utilizzo e il confronto di più fonti;
- rilevamento di fake news interpretabili attraverso ML e *Intelligenza Artificiale*;
- intervenire durante la diffusione delle fake news.

Questi suggerimenti, mostrano implicitamente, quali sono le debolezze nel contrasto delle fake news allo stato dell'arte attuale. Il limite principale negli approcci attuali è che anche se esistono molti tool per il riconoscimento delle fake news [31], non è detto che tutte le persone siano disposte a utilizzare questi strumenti per informarsi su un qualche avvenimento. Inoltre, non tutti i siti che pubblicano notizie adottano questi tipi di strumenti per filtrare le notizie false che vengono pubblicate sui loro domini. Per questo, nel lavoro di tesi, si è pensato di trovare un modo diverso di intervenire per contrastare la disinformazione, utilizzando un approccio sia *model-driven* che *data-driven*. Visto che esistono numerosissimi strumenti per identificare le notizie false, si è pensato di intervenire, piuttosto che sul rilevamento, sul processo di diffusione di esse all'interno di una rete sociale. Tramite una simulazione ad agenti, quindi, possiamo rilevare i pattern di diffusione delle fake news, cercando di intervenire proprio durante questi meccanismi, facendo sì che la notizia falsa non arrivi oppure si convince l'utente di quali siano i fatti veritieri, puntando su tecniche cognitive. Infine, dall'analisi effettuata non emergono studi che combinano gli approcci *model-driven* e *data-driven* per studiare il fenomeno e contrastarlo.

Capitolo 3

Background scientifico metodologico della ricerca

In questo capitolo verrà esposto un quadro generale su aspetti di base che caratterizzano metodologie e tecnologie adoperate in questo lavoro di tesi; in particolare si espongono:

- una overview sulle tecnologie utilizzate (Sez. 3.1);
- una introduzione sui principi base delle simulazioni ad agenti (Sez. 3.2);
- una breve introduzione al RL (Sez. 3.3).

3.1 Overview

Il lavoro di tesi, cioè la creazione di una simulazione che rappresenta la diffusione delle fake news all'interno di una rete, è stato ottenuto mediante l'utilizzo di due tecnologie che sono state messe in comunicazione. Da una parte abbiamo la simulazione basata ad agenti che vuole rappresentare, approssimativamente, la diffusione di una notizia all'interno di un social network, ed è, già di per sé, un sistema complesso. Dall'altra abbiamo l'utilizzo di un'intelligenza artificiale, in grado di poter addestrare un modello sulle osservazioni effettuate sulla simulazione, e intervenire, tramite un *super-agent* presente nella simulazione, nel modo migliore possibile per riuscire a contrastare la diffusione delle notizie false. La comunicazione tra le due tecnologie avviene appunto, quando devono essere prese le informazioni dello stato della rete, che vengono poi passate al modello che addestra il *super-agent*. Una volta scelta l'azione da intraprendere, questa viene eseguita sulla simulazione e se ne studiano gli effetti.

3.2 La simulazione sociale basata su agente (ABM) in pillole

Le simulazioni sociali sono nate dalla necessità di rappresentare delle società in cui attori individuali e collettivi, come le organizzazioni, possano essere direttamente rappresentati, osservando gli effetti delle loro interazioni. Questo ha data via alla possibilità di usare metodi sperimentali con fenomeni sociali, o almeno con le loro rappresentazioni tramite programmi per computer; riuscendo così a studiare direttamente le emergenze di istituzioni sociali da interazioni fra individui.

In questa sezione verranno analizzate alcune caratteristiche della simulazione sociale basata su agenti; vedremo in particolare: l'approccio delle simulazioni sociali in generale nella sotto sezione 3.2.2 e i tool più famosi per le simulazione ad agenti nella sottosezione 3.2.3.

3.2.1 Premesse (scienza sociale computazionale)

Lo sviluppo di programmi per computer che simulino gli aspetti dei comportamenti sociali può contribuire a comprendere i processi sociali. La maggior parte delle ricerche scientifiche può o sviluppare o usare, alcuni tipo di modelli teorici. In generale, queste teorie sono sviluppate sotto forma di testo, anche se qualche volta la teoria è rappresentata con un'equazione. Un terzo modo è quello di esprimere le teorie come programmi per computer. I processi sociali, possono essere simulati tramite programmi. In alcune circostanze, è anche possibile fare esperimenti su sistemi sociali artificiali che sarebbero impossibili o immorali da condurre sugli esseri umani. Un vantaggio di usare le simulazioni è quello che è necessario pensare attraverso le assunzioni di qualcuno in maniera basilare in modo da poter creare dei modelli simulativi utili. A ogni parametro deve essere assegnato un valore, altrimenti sarebbe impossibile eseguire la simulazione. Questo approccio permette anche ad altri ricercatori di poter ispezionare il modello, in tutti i suoi dettagli. Questi benefici di chiarezza e precisione hanno però anche degli svantaggi. Le simulazioni di processi sociali complessi involvono la stima di molti parametri, e adeguare i dati affinché possano essere fatte delle stime può essere difficile. Un altro beneficio della simulazione è che, in alcune circostanze, può dare dei punti di vista sui "pericoli" dei macro fenomeni a partire da azioni a livello microscopico.

3.2.2 L'approccio visto più da vicino

Il modello implementato nello studio di tesi rappresenta un modello *multi-agente*. Questo tipo di modello ha degli agenti che interagiscono in un ambiente virtuale. Gli agenti sono programmati ad avere un certo grado di autonomia, per reagire ed agire sull'ambiente e sugli altri agenti, e di avere un obiettivo che mirano a raggiungere. In questi modelli, gli agenti possono avere una corrispondenza uno a uno con gli individui che esistono nella società del mondo reale che si vuole modellare, mentre le interazioni tra gli agenti possono corrispondere alle interazioni tra gli attori nel mondo reale. Con questi modelli, è possibile inizializzare il mondo virtuale a un'impostazione iniziale su cui poi il modello verrà eseguito e si osserveranno i comportamenti. Nello specifico, è possibile analizzare pattern di azioni che possono emergere dalla simulazione. Gli agenti sono generalmente programmati usando o un linguaggio di programmazione *object-oriented* o delle librerie simulative specifiche o modellando l'ambiente, e sono costruiti usando collezioni di regole condizione-azione che li rendono in grado di "percepire" e "reagire" alla loro situazione, per seguire l'obiettivo che gli è stato dato, e a interagire con gli altri agenti, per esempio, mandano messaggi. Anche se l'utilizzo di una simulazione per generare pattern che uno si aspetta di trovare (se il modello è corretto) e poi confrontare questi con le osservazioni del mondo reale è considerabile più semplice rispetto a provare ad estrarre direttamente dati dettagliati sui processi sociali, ci sono due complicazioni che devono essere considerate. La prima è che la maggior parte dei modelli e delle teorie sulla quale si basano sono stocastiche, cioè si basano in parte su delle possibilità casuali. La seconda è che molti modelli differenti possono far emergere gli stessi pattern. Quindi, una corrispondenza tra quello che uno vede emergere dal modello e quello che uno vede nel mondo sociale reale è una condizione necessaria, ma non sufficiente per arrivare alla conclusione che il modello sia corretto. Tutto quello che uno può fare è quello di incrementare gradualmente la precisione di un modello testandolo su tante osservazioni diverse. Da questo punto di vista, la metodologia della simulazione non è differente da altri approcci nelle scienze sociali [34].

3.2.3 ABM: I tool disponibili

Attualmente esistono vari tool che permettono di creare simulazioni agent-based, mostrati nella tabella 3.1. Il tool utilizzato per creare il modello sviluppato nel lavoro di tesi è *NetLogo 6.2.0*.

Piattaforma	**Dominio Primario**	**Organizzazione**	**Linguaggio di programmazione**
Adaptive Modeler	Costruire simulazioni agent-based per previsioni di mercato su azioni finanziare reali.	Altreva; Utrecht, Netherlands	Genetic programming engine
AgentScript	Creare simulazioni agent-based tramite una piattaforma web	Owen Densmore, RedfishGroup LLC	Javascript
AnyLogic	Simulazioni agent-based *general purpose*	The AnyLogic Company; Oakbrook Terrace, Illinois, USA	Java
FAME	Simulazioni agent-based distribuite che modellano sistemi energetici e mercati	German Aerospace Center, Germany	Java; Python
Framsticks	Simulazioni 2D/3D di sistemi multi-agente e vita artificiale	Poznan University of Technology, Poznan, Poland	FramScript
GAMA Platform	Ambiente di modellazione e simulazione di sviluppo agent-based	IRD/SU international research unit UMMISCO, France	GAML
MASON	Simulazioni agent-based *general purpose*, complessità sociale, modelli fisici, modellazione astratta, intelligenza artificiale e machine learning	George Mason University, Fairfax, Virginia, USA	Java
NetLogo	Simulazioni agent-based per scienze sociali e naturali	Northwestern University, Evanston, Illinois, USA	Netlogo
Repast	Scienze sociali	Argonne National Laboratory, University of Chicago; Lemont, Illinois, USA	Java; Python

Tabella 3.1: Tools attualmente utilizzati per creare simulazioni ad agenti

3.3 Reinforcement Learning in pillole

Il RL è una branca dell'intelligenza artificiale che si base sull'apprendere tramite delle interazioni in un ambiente di riferimento che puntano ad ottenere un obiettivo. RL ha a che fare con diversi problemi rispetto all'apprendimento supervisionato o non supervisionato, che è, quello di osservare un agente che agisce all'interno di un ambiente e decide che azione prendere sulla base di *reward* assegnate da condizioni associate all'ambiente stesso. Invece, nell'apprendimento supervisionato, l'agente apprende come classificare i dati che riceve in input per ottenere un output. Per ottenere ciò, durante un processo di apprendimento supervisionato, il classificatore apprende sia dai dati di input, sia dalle categorie (cioè l'output desiderato). Nell'apprendimento non supervisionato, l'algoritmo ha a disposizione solo dei dati in input che non

sono etichettati/categorizzati, e ha come scopo quello di cercare di scoprire strutture nascoste e pattern intrinsechi al fine di raggruppare (si parla quindi di clustering) oppure di ridurre le dimensionalità (si parla quindi di features extraction / dimensionality reduction / matrix factorization).

3.3.1 Il modello di apprendimento per rinforzo: principi generali

Nel RL, l'agente interagisce con l'ambiente scegliendo di volta in volta quale azione compiere con lo scopo di raggiungere un obiettivo. Ciascun azione dell'agente cambia lo stato dell'ambiente e influisce sulle sue scelte future (azioni). Al fine di osservare gli effetti non prevedibili delle azioni, l'agente prende in considerazione alcuni elementi: *policy*, *reward* e una *value function*. La *policy* è in breve, l'associazione di un'osservazione dello stato dell'ambiente ad un'azione dell'agente e indica quale azione è preferibile compiere in corrispondenza ad un particolare stato. La *reward* indica quanto sia desiderabile per l'agente di essere in quel determinato stato; in questo senso può essere inteso come l'obiettivo a breve termine dell'agente. L'intero processo è diviso in una successione di azioni, prese nel tempo, da parte dell'agente, ciascuna corrispondente a un cambio di stato dell'ambiente e ad una *reward* data all'agente in base all'azione presa. Lo scopo principale dell'agente è quello di massimizzare le *reward* ricevute nel tempo. La *value function* è la ricompensa a lungo termine data all'agente. Dato uno stato, la *value function* corrispondente prevede le ricompense che sono determinate da esso, per esempio, la quantità totale delle *reward* che accumulerà l'agente a partire da quello stato. L'obiettivo del RL, è quello di costruire una *policy* e una *value function* che l'agente userà per massimizzare le *reward*.

3.3.2 Apprendimento per rinforzo: il Q-Learning e deep Q-Learning

L'algoritmo del *Q-Learning* si basa sulla nozione di funzione Q. La funzione Q (anche detta la funzione valore dello stato-azione) di una *policy* $\pi, Q^{\pi}(s, a)$, misura il rendimento atteso o la somma scontata delle *reward* ottenute dallo stato s prendendo un'azione a prima, e usare una *policy* π dopo. Definiamo la *Q-function* ottimale $Q^*(s, a)$ come il rendimento massimo ottenibile partendo dall'osservazione s, intervenendo e seguendo la *policy* ottimale successivamente. La *Q-function* ottimale obbedisce alla seguente equazione dell'ottimalità di Bellman:

$$Q^*(s, a) = \mathbb{E}[r + \gamma \ max_{a'} \ Q^*(s', a')]$$

Ciò significa che il rendimento massimo da uno stato s e un'azione a, è la somma delle *reward* immediate r e il valore di ritorno (ridotto da γ) ottenuto seguendo la *policy* ottimale fino alla fine dell'episodio. L'idea di base dietro il *Q-Learning* è di usare l'equazione dell'ottimalità di Bellman come un aggiornamento iterativo (vedi Figura 3.1):

$$Q_{i+1}(s,a) \leftarrow \mathbb{E}[r + \gamma \; max_{a'} \; Q_i(s',a')]$$

Può essere mostrato che questa funzione converge verso la *Q-function* ottimale, per esempio $Q_i \rightarrow Q^*$ per $i \rightarrow \infty$ [35].

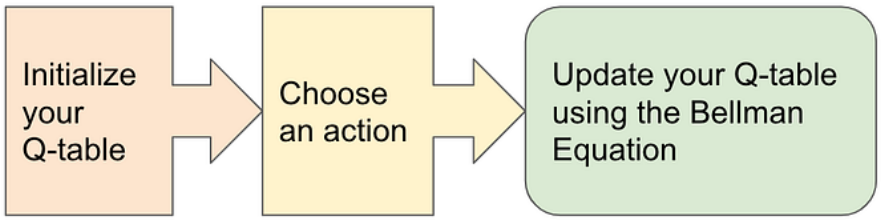


Figura 3.1: L'algoritmo di Q-Learning

Per la maggior parte dei problemi, non è pratico rappresentare la *Q-function* come una tabella contente i valori per ogni combinazione di s e a. Invece, viene addestrata una funzione di approssimazione, come una *neural network* con parametri θ, per stimare le *Q-values*, per esempio $Q(s,a;\theta) \approx Q^*(s,a)$. Questo può essere ottenuto minimizzando la seguente perdita a ogni step i:

$$L_i = \mathbb{E}_{s,a,r,s' \sim (\cdot)} \; [(y_i - Q(s,a;\theta_i))^2]$$

dove

$$y_i = r + \gamma max_{a'} Q(s',a';\theta_{i-1})$$

Qui, y_i è chiamata *differenza temporale* obiettivo, e $y_i - Q$ è chiamato *errore della differenza temporale* (Temporal Difference). ρ rappresenta la distribuzione del comportamento, la distribuzione sulle transizioni s, a, r, s' collezionate dall'ambiente. Bisogna notare che i parametri dell'iterazione precedente θ_{i-1} sono fissi e non vengono aggiornati. In pratica, si utilizza uno snapshot dei parametri della network di alcune iterazione precedenti invece di prendere l'ultima. Questa copia è chiamata *target network*.

Il *Q-learning* è un algoritmo che impara tramite una *policy greedy* $a = max_a Q(s,a;\theta)$ mentre viene utilizzata una *policy* di comportamento diversa

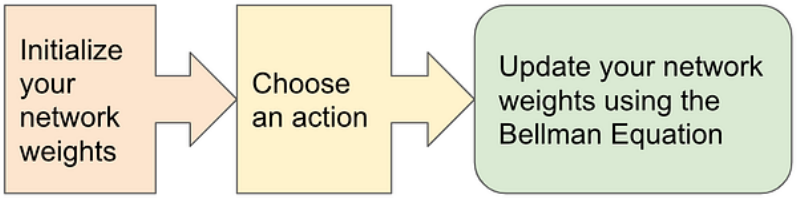


Figura 3.2: L'algoritmo di Deep Q-Learning

per agire nell'ambiente. Questa *policy*, di solito, è di tipo $\epsilon - greedy$ che seleziona l'azione più vantaggiosa con probabilità $1-\epsilon$ e un'azione casuale con probabilità ϵ per assicurarsi una buona copertura dello spazio stato/azioni.

Experience Replay. Per evitare di computare tutto in termini di perdita temporale, si può minimizzare l'utilizzo tramite un gradiente discendente stocastico. Se la perdita è computata usando soltanto le ultime transizioni s, a, r, s', viene ridotto soltanto a un *Q-learning* standard. Per far sì che gli aggiornamenti della network fossero più stabili è stata introdotta una tecnica chiamata *Experience Replay*. A ogni passo della collezione delle informazioni, le transizioni sono aggiunte a un buffer circolare chiamato *replay buffer*. Poi, durante l'addestramento, invece di usare solo l'ultima transizione per calcolare la perdita e il suo gradiente, si computa utilizzando della transizioni composte da *mini-batch* che vengono campionati dal buffer *replay*. Questo porta a due vantaggi: maggior efficienza dei dati rispetto al riuso di ogni transizione in molti aggiornamenti dello stato, e una migliore stabilità usando transizioni non correlate nello stesso *batch*.

Capitolo 4

Analisi e contrasto delle dinamiche di diffusione delle fake news: un approccio sperimentale

In questo capitolo, si presentano innanzitutto i dettagli della simulazione ad agenti come modello di diffusione delle fake news, e successivamente quelli relativi al modello di DRL progettato per contrastarne la viralità. Nello specifico si espongono:

- una overview sul lavoro con le domande/obiettivi di ricerca di questo studio (Sez. 4.1),
- la strategia adottata alla base dell'architettura presentata (Sez. 4.2),
- la descrizione della simulazione nel dettaglio (Sez. 4.3),
- dettagli sul modello di Deep Q-Learning utilizzato come *super-agent* (Sez. 4.4).

4.1 Overview: domande di ricerca e obiettivi

In questo lavoro, utilizziamo un modello di simulazione ad agenti per studiare la diffusione virale delle fake news in un social network in presenza di una echo chamber e sfruttiamo un metodo di DRL per sviluppare strategie di contenimento alla diffusione virale delle fake news.

La nostra idea è quella di esplorare metodi innovativi per lo studio di questi fenomeni. L'innovazione risiede nello sfruttamento di due approcci spesso visti in contrapposizione tra loro, *model driven* e *data driven*. Al meglio della nostra conoscenza, al momento non esistono studi di questo genere, cioè che hanno combinato simulazioni agent-based con metodi di deep learning quali il RL, nel contesto del fenomeno delle fake news.

4.2 Architettura della strategia

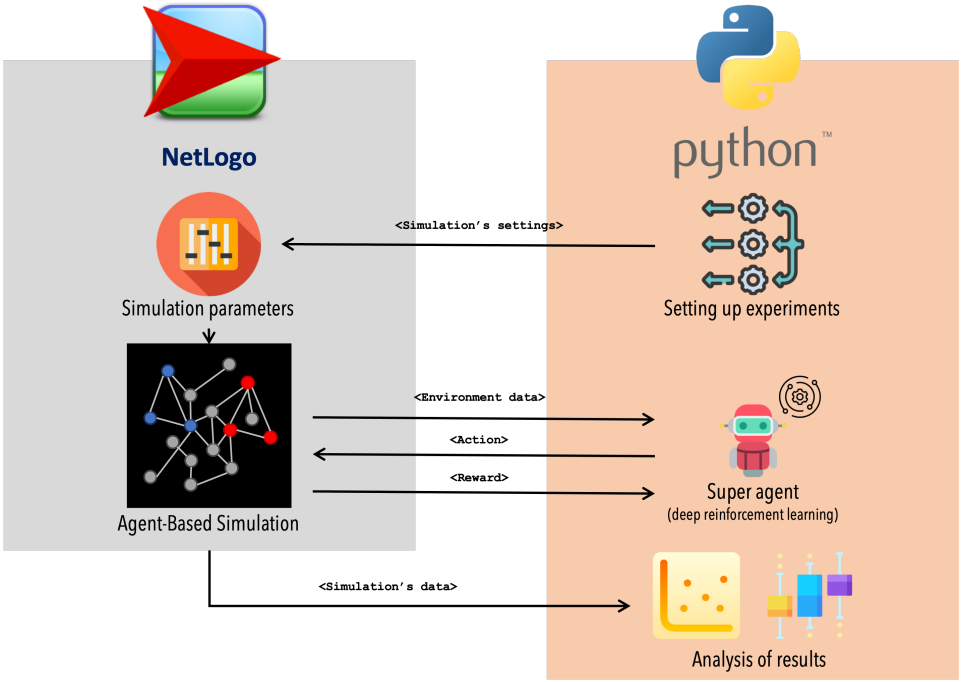


Figura 4.1: Abstract visuale dell'architettura della strategia.

L'architettura della strategia proposta si basa sull'uso combinato di *Python* e *NetLogo*, sfruttando le API NetLogo per Python, cioè *PyNetLogo* [36] (vedi Figura 4.1). Innanzitutto, l'inizializzazione dei parametri della simulazione avviene tramite uno script *Python* che invia le impostazioni a *NetLogo* e fa partire la simulazione (`Simulation's settings`). Successivamente, a simulazione avviata, si raccolgono le osservazioni sul network ed i suoi agenti

dall'ambiente *NetLogo*; a ogni *tick*[1] di simulazione vengono prelevati alcuni parametri rilevanti dell'ambiente simulato (`Environment data`) e sfruttati dal *super-agent* per prendere decisioni (`action`) che si ripercuotono sulla simulazione ed i suoi agenti. Da tali operazioni, il *super-agent* ottiene una ricompensa che sfrutterà per prendere sempre meglio le proprie decisioni in futuri *tick* (`reward`). Al termine della simulazione, tutti i parametri di interesse per rispondere alle domande di ricerca di cui sopra (Sez. 4.1) vengono prelevati e resi fruibili tramite infografiche intuitive.

Questa strategia combinata sarà utilizzata sia per effettuare esperimenti che coinvolgono il *super-agent* (in fase di training come in fase di testing) sia per tutti quei test che non coinvolgono questa entità.

Nelle successive sezioni, verranno illustrati i dettagli degli elementi singoli della strategia.

4.3 Simulazione della diffusione delle fake news: modello agent-based

Qui viene presentata la descrizione completa del modello che simula la diffusione di una fake news all'interno di una rete sociale. La descrizione segue il protocollo *ODD* [37]. Il modello è stato implementato su *NetLogo ver. 6.2.0*, eseguito tramite *Python ver. 3.10*.

4.3.1 Scopo

Il modello simula come una fake news può circolare all'interno di una rete in cui gli agenti possono avere tre tipi di opinioni: l'opinione A che supporta la notizia falsa (nodi di colore arancione), l'opinione B che è a favore di fatti veritieri (nodi di colore blu) e l'opinione neutra che indica un'indecisione o ignoranza sull'argomento (nodi di colore grigio). Nella simulazione è presente un *super-agent* in grado di intervenire a ogni passo e decidere quale azione intraprendere per riuscire a contrastare l'andamento dell'opinione A.

4.3.2 Entità, variabili di stato, e misure

Il modello include due tipi di entità: i *basic-agents* e il *super-agent*. I *basic-agents* hanno come variabili: activation treshold (Θ), parametri di network analysis (betweenness, eigenvector, closeness, clustering, community, page-rank, in-degree, out-degree, degree), is-a-active, is-b-active, is-in-cluster, war-

[1] Una unità di tempo che determina istanti diversi nella simulazione.

ning, reiterate, is-opinion-b-static, opinion-metric, received-a news-counter, received-b-news-counter. Sono presenti inoltre le variabili globali, visibili da tutti i nodi all'interno della rete, che sono: neutral-agents, active-a-agents, active-b-agents, in-cluster-agents, total-number-agents, total-links, k, global-cascade, is-warning-active, is-reiterate-active. Tutte le variabili sono spiegate nella tabella 4.1 Il *super-agent* non ha alcun tipo di attributo visto che il suo scopo è quello di intervenire su alcuni parametri della simulazione per contrastare la diffusione dell'opinione A. La misura dell'ambiente in cui si svolge la simulazione è proporzionale al numero di agenti presenti all'interno della rete e dai collegamenti che si creano tra di loro.

4.3.3 Panoramica del processo e pianificazione

All'inizio della simulazione vengono scelti il numero di nodi N da cui deve essere composta la rete, il numero di *tick* che determinano la durata della simulazione e il tipo di rete che si vuole costruire, che può essere di tipo: *Erdős–Rényi*, *Small World* o *Preferencial Attachment*. In base al tipo di rete scelta, si impostano i parametri della creazione dei collegamenti fra i nodi. Nel caso di una rete *Erdős–Rényi*, che segue un distribuzione di *Poisson*, viene impostato un parametro k che determina il valore di distribuzione normale con cui vengono generati il numero collegamenti medi che devono avere i nodi, insieme alla deviazione standard. Es. se imposto $k = 7$, allora mediamente i nodi della rete avranno 7 link (7 vicini).

I collegamenti tra i nodi possono essere direzionati o meno, ma nella maggior parte delle nostre esecuzioni del modello abbiamo usato i collegamenti non direzionati. Per la rete di tipo *Small World*, vengono selezionate la *neighborhood-size* e la *rewire-probability*, che corrispondono rispettivamente al numero di nodi a cui deve essere collegato un nodo e alla probabilità che ciascun arco sia disconnesso da una delle sue estremità e connesso a un altro nodo nella rete scelto a caso.

Infine per il *Preferencial Attachment* viene generata una network di tipo "*scale free*", in cui gli agenti sono aggiunti, uno alla volta, con i nodi aggiunti precedentemente. Più collegamenti ha un nodo, maggiore è la possibilità che i nuovi nodi creino dei collegamenti con esso quando sono aggiunti.

Echo Chamber. Una volta creata la rete viene inizializzata la *echo chamber* (o cluster). La *echo chamber* viene vista come un insieme di utenti caratterizzata da due proprietà: l'*opinion polarization* (P_o) e la *network polarization* (P_n). L'*opinion polarization* implica che gli utenti, in relazione a un'opinione, sono più inclini a condividerne le vedute. La *network pola-*

Attributo	Descrizione	Dominio
`activation Threshold` (Θ)†	Soglia di attivazione.	$\mathbb{R}[0,1]$
`betweenness centrality`	Viene prese ogni altra possibile coppia di *basic-agent* e, per ogni coppia, viene calcolata la proporzione dei percorsi più brevi tra i membri della coppia che passa attraverso l'agente attuale. La *betweenness* di un agente è la somma di questi.	$\mathbb{R}[0,\infty]$
`community`	Numero rappresentante la *community* di cui fa parte il *basic-agent*.	$\mathbb{N}[1,\infty]$
`eigenvector centrality`	Può essere pensata come l'influenza che un nodo ha sulla network. In pratica, gli agenti che sono connessi a molti altri agenti che sono a loro volta ben connessi ottengono un valore di *eigenvector centrality* più alto.	$\mathbb{R}[0,1]$
`closeness centrality`	Definita come l'inverso della media della sua distanza da tutti gli altri nodi.	$\mathbb{R}[0,1]$
`clustering centrality`	Rappresenta come il nodo è connesso ai suoi vicini. É definita come il numero di collegamenti tra i nodi collegati ad esso, diviso per il numero totale dei possibili collegamenti tra i suoi vicini.	$\mathbb{R}[0,1]$
`page rank centrality`	Può essere visto come la proporzione di tempo che un agente, camminando all'infinito in modo casuale sulla rete, spenderà su questo nodo.	$\mathbb{R}[0,1]$
`in degree`	Numero di collegamenti in entrata sul nodo.	$\mathbb{N}[0,N]$
`out degree`	NUmero di collegamenti in uscita sul nodo.	$\mathbb{N}[0,N]$
`degree`	Somma di *in-degree* e *out-degree*	$\mathbb{N}[0,N]$
`is-a-active`	Indica se il *basic-agent* sostiene l'opinione A	{0,1}
`is-b-active`	Indica se il *basic-agent* sostiene l'opinione B	{0,1}
`warning`	Indica se è stato mandato un segnale di *warning* dal *super-agent*	{0,1}
`reiterate`	Indica se è stato mandato un segnale di *reiterate* dal *super-agent*	{0,1}
`is-opinion-b-static`	Viene impostato dal *super-agent*, costringendo il nodo a sostenere l'opinione B fino alla fine della simulazione	{0,1}
`opinion-metric`	Indica più dettagliatamente, verso quale opinione si trova il nodo (più dettagli nella Sez. 4.3)	$\mathbb{R}[0,1]$
`neutral-agents`	Indica il numero di *basic-agent* neutrali	$\mathbb{N}[0,N]$
`active-a-agents`	Indica il numero di *basic-agent* che hanno la variabile *is-a-active* == *true*	$\mathbb{N}[0,N]$
`active-b-agents`	Indica il numero di *basic-agent*che hanno la variabile *is-b-active* == *true*	$\mathbb{N}[0,N]$
`in-cluster-agents`	Indica il numero di *basic-agent* che hanno la variabile *is-in-cluster* == *true*	$\mathbb{N}[0,N]$
`total-number-agents`	Indica il numero totale di *basic-agent*	$\mathbb{N}[0,\infty]$
`total-links`	Indica il numero totale degli archi	$\mathbb{N}[0,\infty]$
`k`	Indica il grado medio dei collegamenti che hanno i *basic-agent*	$\mathbb{N}[0,\infty]$
`global-cascade`	Indica la frazione dei nodi che hanno *is-a-active* = *true* (più delttagli nella Sez. 4.3))	$\mathbb{R}[0,1]$
`is-warning-active`	Indica se è stata attivata la procedura di *warning* (più dettagli nella Sez.4.3)	{0,1}
`is-reiterate-active`	Indica se è stata attivata la procedura di *reiterate* (più dettagli nella Sez. 4.3)	{0,1}

† Intesa come l'inverso della creduloneria. Più è bassa più il basic-agentè *credulone*.

Tabella 4.1: Attributi delle entità di tipo basic-agents nella simulazione di diffusione delle fake news.

rization indica che gli utenti sono più connessi gli uni con gli altri rispetto al resto del network. In altre parole, una *echo chamber* ha dei nodi più strettamente connessi che tendono a condividere la stessa opinione su una determinata narrativa.

Vengono quindi impostati i valori di P_o e P_n, e per la creazione della *echo chamber* viene scelta anche la frazione dei nodi che ne devono far parte, tramite il parametro *echo chamber fraction* (ECF).

Prima dello start della simulazione, viene moltiplicata la ECF per il numero totale di nodi N, così da ottenere il numero di nodi che devono essere inseriti nella *echo chamber*, `c` $N \times ECF$. Si scelgono a caso `c` nodi dalla rete e si imposta il loro attributo `is-in-cluster = true`.

Il calcolo per ottenere il numero di archi è $E = N \times \frac{k}{2}$, dove N è il numero di nodi e k è il grado medio che hanno i nodi, impostato all'inizio della simulazione. Il grado medio viene diviso per due, visto che i collegamenti non sono direzionati e ogni arco ha due lati. Successivamente il valore ottenuto viene moltiplicato per P_n. L'equazione finale sarà:

$$E' = E \times P_n$$

Il valore E', serve a determinare quanti archi devono essere scelti dall'insieme totale degli archi della rete. Su E', in modo casuale, viene eseguito un controllo per verificare se esattamente un nodo, collegato a una delle due estremità, appartiene alla *echo chamber* (cioè ha l'attributo `is-in-cluster == true`). In caso di successo, l'arco viene eliminato e viene sostituito con un collegamento tra il nodo che già faceva parte dell'*echo chamber* scelto prima con un altro all'interno del cluster. Così facendo aumenta il `degree-centrality` medio tra i nodi appartenenti alla *echo chamber*, creando più coesione.

Successivamente, viene scelto un nodo casuale *init* che fa parte della *echo chamber*, che serve come punto di attivazione, e viene impostata `is-a-active = true`. Si attribuisce, inoltre, un valore casuale dell'*opinion metric* come meglio definito successivamente. Quindi, si selezionano i vicini di *init* e anche il loro attributo `is-a-active` viene impostato a `true`.

Tale procedura di attivazione viene ripetuta in modo speculare per i nodi di tipo B: viene scelto un nodo all'esterno della *echo chamber* (cioè ha l'attributo `is-in-cluster == false`) e il suo attributo `is-b-active` viene impostato a `true`; tutti i nodi a esso collegati, che non abbiano già un'opinione di tipo A e che non facciano parte della *echo chamber*, sono soggetti al cambiamento di opinione con `is-b-active = true`.

Una volta terminata questa procedura, viene impostata l'`activation threshold` a ciascuno dei nodi della rete; in particolare, ai nodi appartenenti

alla *echo chamber*, viene impostato un Θ pari a $\Theta - P_o$, mentre ai nodi al di fuori viene impostato semplicemente il valore di Θ, scelto all'inizio della simulazione. Ciò significa che i nodi all'interno del cluster sono più creduloni (più influenzabili dalle opinioni).

super-agent. Dopo aver creato la rete e costruita la *echo chamber*, viene inserito il *super-agent* all'interno della rete. Questo tipo di agente non ha attributi (come evidenziato in Sez. 4.3.2) e ha a disposizione quattro azioni: tre di queste hanno lo scopo di contrastare la diffusione dell'opinione A durante la simulazione e ridurre la *global cascade*, corrispondente alla frazione dei nodi con `is-a-active == true` (vedi Sez. 4.3.6); l'ultima delle quattro azioni lascia il *super-agent* in attesa. Le tre azioni di contrasto sono denominate *warning*, *reiterate* e *static b nodes* (vedi Sezione 4.3.5). Per ogni azione sceglie una frazione (percentuale) di *basic-agents* su cui può intervenire, tramite il parametro `node-range`.

Go. Dopo aver inizializzato la rete, può iniziare la simulazione. Il modello procede in istanti di tempo determinati dai *tick*. All'inizio si controlla se il *super-agent* ha attivato una delle tre azioni di contrasto alla diffusione delle fake news, ed si esegue un ulteriore controllo per evitare di attivare più azioni allo stesso *tick*. Se un agente ha cambiato la sua opinione in un determinato *tick*, questo verrà impostato come attivo verso l'opinione A o B solo al *tick* successivo. Successivamente viene chiesto a tutti i *basic-agents* ($BA = \{ba_1, ba_2, \ldots, ba_N\}$) di contare il numero di nodi vicini di tipo A o B e calcolarne la frazione rispetto al numero totale di vicini. In altre parole, ogni *basic-agent* ba_i, $1 \leq i \leq N$, calcola:

$$fraction_a^{ba_i} = \frac{|linkneighbors(ba_i, \texttt{basic-agents[is-a-active==true]})|}{|linkneighbors(ba_i)|}$$

$$fraction_b^{ba_i} = \frac{|linkneighbors(ba_i, \texttt{basic-agents[is-b-active==true]})|}{|linkneighbors(ba_i)|}$$

dove $linkneighbors(ba_i)$ è una funzione che restituisce i *basic-agents* collegati direttamente con un *basic-agent* ba_i (cioè i vicini del nodo).

In base al tipo di frazione predominante, viene fatto il confronto di quest'ultima con la `activation-threshold`.

$$\begin{cases} \max(fraction_a^{ba_i}, fraction_b^{ba_i}) > \Theta & \text{viene modificata l'}\textit{opinion-metric} \\ fraction_a^{ba_i} = fraction_b^{ba_i} & ba_i \text{ non verrà influenzato dai vicini} \end{cases}$$

Dopo aver eseguito la procedura principale di diffusione delle notizie, entrano in atto le azioni del *super-agent*, se attive. Alla fine della simulazione, una volta raggiunto il numero di *tick* stabilito, viene calcolata la *global cascade* (Sez. 4.3.5).

4.3.4 Concetti del design

Principi di base. Il modello è costruito sull'assunzione che ci sia una *echo chamber* nella rete da cui inizia la diffusione della fake news. Nella stessa rete sono presenti agenti con opinioni sia di tipo A che B che permettono di simulare due opinioni contrastanti su un argomento. Uno dei fattori principali che determina l'andamento di un'opinione è la soglia di attivazione Θ dei nodi (o credduloneria).

Emergenza. L'equilibrio della rete sociale dipende da quanto i *basic-agents* riescano a convincere i *basic-agents* adiacenti dell'opinione che supportano, così da riuscire ad avere il sopravvento rispetto all'opinione opposta. Il *super-agent* è l'unico agente in grado di poter spostare gli equilibri, cercando di convincere più *basic-agents* possibili, a sostenere l'opinione di tipo B, sia per importanza (valori di network analysis) che per numerosità.

Adattamento. L'unica entità che può adattarsi durante l'avanzamento della simulazione è il *super-agent*, in grado di identificare i nodi critici, che stanno diffondendo le notizie di tipo A, e intervenire.

Obiettivi. L'obiettivo dei *basic-agents* è quello di diffondere l'opinione che stanno supportando in un determinato *tick* di simulazione. Differentemente, l'obiettivo del *super-agent* è quello di ridurre la *global cascade*, tramite le sue azioni che mirano a convincere i *basic-agents* dell'opinione B.

Apprendimento. Durante la simulazione, i *basic-agents* non apprendono da quello che sta accadendo sia nella rete che intorno a loro. Il sistema di apprendimento che abbiamo usato riguarda solo il *super-agent*, che viene addestrato tramite tecniche di DRL.

Predizione. I *basic-agents* non adottano nessun tipo di predizione sul loro comportamento. Il *super-agent* invece, è governato da una rete neurale profonda che permette di prevedere quale sia l'azione migliore da intraprendere in un determinato istante di tempo (*tick*), data un'osservazione dello stato della rete e dei *basic-agents*.

Rilevamento. Il super-agent è in grado di rilevare quasi tutti i cambiementi della rete, sia dal punto di vista della *network analysis* sia dal punto di vista dello stato dei *basic-agents*. I basic-agents sono obbligati a rilevare l'opinione dei basic-agents vicini (collegati a loro a distanza 1) per decidere da che parte stare.

Interazione. I *basic-agents* interagiscono tramite i collegamenti con gli altri *basic-agents*. La diffusione delle opinioni avviene proprio tramite i collegamenti con i *basic-agents* vicini.

Stocastica. Gli eventi stocastici sono determinati dal numero di collegamenti che cambia a ogni simulazione della rete, visto che ogni tipo di rete a disposizione nel modello permette di creare, ogni volta, network casuali. Quando viene inizializzata la *echo chamber*, viene scelto un nodo casuale che serve come punto di attivazione per l'opinione A, in quanto tutti i nodi a esso collegati vengono impostati come attivi verso l'opinione A. La stessa cosa succede al di fuori della *echo chamber*, dove viene scelto un nodo casuale che serve come punto di attivazione per l'opinione B, in quanto tutti i nodi a esso collegati vengono impostati come attivi verso l'opinione B. Altri eventi casuali si verificano con le azioni del *super-agent*, in particolare il *Warning* e il *Reiterate*. Queste azioni sono descritte nella Sez. 4.3.

Collettivi. I *basic-agents* rappresentano la possibile struttura di un social network, dove i nodi stessi corrispondono agli utenti iscritti al network, mentre gli archi rappresentano i collegamenti fra gli utenti (follower). Il *super-agent* può rappresentare un'ente governativo o un gestore della rete che vuole contrastare le fake news all'interno del social network.

Osservazioni. Le *opinion metric* dei *basic-agents* vengono osservate e aggiornate a ogni *tick*, così come le misure di network analysis che permettono al *super-agent* di intervenire in modo puntuale.

Dati in input. I dati principali inseriti in input[2] prima di far partire la simulazione sono: `nb-nodes` che indica il numero di nodi, il tipo di collegamenti fra i nodi(direzionati o non direzionati), il tipo di network (può essere di tipo: *Erdős–Rényi*, *Small World* o *Preferencial Attachment*), `k-value`, *std-dev*, `rewire-prob`, `neighborhood-size`, il numero di `tick`, P_n, P_o, la soglia di

[2]I parametri scelti dall'interfaccia grafica di NetLogo, vengono considerati come variabili globali

attivazione (Θ), `echo-chamber-fraction` (ECF), cioè la frazione di nodi che fanno parte dell' *echo chamber*, l'`initial-opinion-metric-value` che sarebbe il valore di `opinion-metric` iniziale a cui vengono impostati i nodi neutri con il relativo valore di avanzamento `opinion-metric-step` e infine, `is-a-active` che determina se è attivo o meno il `super-agent` durante la simulazione. Altri dati in input sono considerabili dinamici, in quanto essi possono essere cambiati durante la simulazione come: il `Warning Impact` sia per i nodi con un'opinione che per quelli neutri, il `node-range` che indica la frazione dei nodi a cui si collega il `super-agent`, il `node-range-static-b` che indica la frazione dei nodi a cui il *super-agent* obbliga il mantenimento dell'opinione B, `choose-method` che determina il metodo di scelta dei collegamenti che crea il `super-agent`, in base a `degree`, `betweenness` e `page-rank`, e il `global-warning` che indica se il `warning` può essere globale o meno. Infine ci sono i comandi che gestiscono la simulazione: il *setup* che inizializza la rete, il *go* che fa avanzare di un *tick* la simulazione e le tre azioni che può fare il *super-agent*.

4.3.5 Sotto-modelli

In questa sottosezione verranno analizzate le azioni che può compiere il *super-agent* e alcuni parametri di osservazione della rete.

Warning

Se il *Warning* è globale, quando viene attivato, viene impostata la variabile `warning` a `true` di tutti i *basic-agents* della rete, rimanendo attivo fino alla fine della simulazione. Se il *Warning* non è globale, il *super-agent* manda il segnale di `warning` solo agli agenti scelti sulla base di alcune proprietà rilevanti. Inoltre, sono presenti due parametri che regolano l'influenza del *Warning*: uno per i nodi con l'opinione di tipo A e uno per i nodi neutrali. La procedura di *Warning* (per le motivazioni vedere tabella 4.2) ha effetto solo sui nodi di tipo A o neutrali e può essere di tue tipi: globale o non globale, e una volta attivata la variabile `warning` su un *basic-agent*, permane fino alla fine della simulazione. Questo perché se un *basic-agent* è indeciso è più probabile che il *Warning* abbia un effetto maggiore su di esso e viceversa per i nodi che già ne hanno una. Quando un *basic-agent* sta per essere influenzato da una opinione, controlla se la sua variabile di `warning` è impostata a `true`. In caso affermativo, viene generato un numero casuale $x \in \mathbb{R}[0, 1]$ e viene verificato che questo valore sia minore o uguale della soglia del `warning-impact` impostata all'inizio della simulazione. Nel caso in cui il

numero generato sia minore viene avviata la procedura di aggiornamento dell'`opinion-metric`.

```
1  Warning
2  // si controlla se il warning ha effetto su un nodo che
       ha già una opinione A
3  if ba_i[is-a-active==true] then
4  |  x = rand(0,1);
5  |  if x ≥ warning-impact then
6  |  |  // aggiorna il valore di opinion-metric sul
   |  |     basic-agent verso l'opinione B, di un valore
   |  |     pari all'opinion-metric-step
7  |  |  calculateOpinionMetric(B);
8  |  end
9  end
10 // si controlla se il warning ha effetto su un nodo che
       non ha un'opinione
11 if ba_i[is-a-active==false] ∧ ba_i[is-b-active==false] then
12 |  x = rand(0,1);
13 |  if x ≥ warning-impact-neutral then
14 |  |  calculateOpinionMetric(B);
15 |  end
16 end
```

Reiterate

La procedura di *Reiterate* (per le motivazioni vedere tabella 4.2) avviata dal *super-agent* imposta a `true` la variabile corrispettiva di *basic-agents* scelti ad hoc (vedremo in seguito le politiche). Quando un *basic-agent* sta per essere influenzato da una opinione, controlla lo stato della variabile `reiterate`. I nodi con questa variabile attiva, `reiterate == true`, riceveranno un'altra notizia di tipo B ad ogni *tick* successivo per un numero di volte pari al numero di vicini. Per esempio, se un nodo ha dieci collegamenti, per dieci *tick* riceverà un'opinione di tipo B, ripetendo il controllo sulla soglia di attivazione. A differenza della diffusione normale, in cui viene analizzata la frazione dei nodi di tipo A o B, in questo caso verrà generato un numero casuale $y \in \mathbb{R}[0,1]$ e se il valore generato è minore o uguale del Θ del nodo, viene aggiornato il valore dell'`opinion-metric` verso l'opinione B. L'operazione di *Reiterate*

può anche terminare prima dello scadere, se il nodo cambia opinione prima che raggiunga il numero massimo di iterazioni stabilite.

```
1  Reiterate
2  // si controlla se il contatore delle iterazioni di
       reiterate ha raggiunto il numero di collegamenti del
       basic agent
3  if reiterate-counter < |linkneighbors(ba_i)| then
4  |   reiterate-counter = reiterate-counter + 1;
5  |   y = rand(0,1);
6  |   if y ≤ activation-threshold then
7  |   |   calculateOpinionMetric(B);
8  |   |   // si controlla se dopo l'aggiornamento
   |   |      dell'opinion metric, verso l'opinione B, il
   |   |      basic agent sta per passare allo stato B al
   |   |      passo successivo, se così l'attributo reiterate
   |   |      viene impostato a false così come il contatore
9  |   |   if is-inactive-next==true then
10 |   |   |   reiterate=false;
11 |   |   |   reiterate-counter=0;
12 |   |   end
13 |   end
14 else
15 |   // se si è raggiunto il numero di reiterazioni
   |      stabilito, la procedura si ferma
16 |   reiterate=false;
17 end
```

Static B Agents

L'azione *Static B agents* (per le motivazioni vedere tabella 4.2) permette al *super-agent* di forzare alcuni nodi, scelti con tre criteri diversi, di mantenere l'opinione B fino alla fine della simulazione. Quindi questi anche se ricevessero notizie di tipo A, non ne sarebbero influenzati. Vengono scelti i nodi con *betweenness*, *page rank* o *degree* più alti. Il numero di nodi scelto è selezionato dal parametro *node-range-static-b* (frazione). Quello che sostanzialmente fa il *super-agent* è impostare l'`opinion-metric` a 0 e imposta a *true* una variabile presente sugli agenti chiamata `is-opinion-b-static`. Così, durante la

simulazione viene controllata se questa variabile è attiva così da non far cambiare opinione al nodo.

4.3.6 Metriche simulative

Opinion metric. Ogni nodo ha un'*opinion metric* che può assumere valori nel range $\mathbb{R}[0, 1]$, in particolare:

$$opinionmetric = \begin{cases} 0 \leq x \leq 0.33 & \text{l'agente ha una opinione di tipo B} \\ 0.33 < x < 0.66 & \text{l'agente è neutrale} \\ 0.66 \leq x \leq 1 & \text{l'agente ha una opinione di tipo A} \end{cases}$$

Inoltre, all'inizio della simulazione viene scelto di quanto deve cambiare l'*opinion metric* quando l'agente riceve un opinione e supera la sua soglia di attivazione, tramite l'*opinion metric step*.

Global Cascade. La *global-cascade* è definita come la frazione del numero nodi attivi di tipo A in un determinato istante della simulazione. Per ogni *basic-agent* ba_i viene controllato se *is-a-active == true*, la somma ottenuta dei nodi che rispettano questa condizione viene divisa per il numero totale dei nodi N. Il calcolo è:

$$global\ cascade = \frac{\sum_{i=1}^{N} ba_i \texttt{[is-a-active==true]}}{N}$$

Viralità. La viralità, V, è definita come la frazione delle volte in cui è la *global-cascade* è maggiore di 0.5 in un numero determinato di simulazioni al termine di esse. Per esempio, se su 100 simulazioni, aventi gli stessi parametri, il numero di volte in cui la *global-cascade* ha superato 0.5 è pari a 30, la viralità sarà 0.30. Chiaramente, V dipende dai parametri impostati per la simulazione, come Θ, P_n ed altri. Sia $T \in \mathbb{N}$ il numero di simulazioni lanciate per un esperimento e $global\ cascade_i$ il valore della *global-cascade* in una generica simulazione i, sia $CGC : \mathbb{R}[0, 1] \to \{0, 1\}$ una funzione che

$$CGC(global\ cascade_i) = \begin{cases} global\ cascade_i > 0.5 & \text{return 1} \\ global\ cascade_i <= 0.5 & \text{return 0} \end{cases}$$

, allora la viralità è calcolata come:

$$V = \frac{\sum_{i=0}^{T} CGC(global\ cascade_i)}{T} \tag{4.1}$$

Global Opinion Metric mean. La *Global Opinion Metric Mean* indica la media globale dei valori di *opinion-metric* di ogni *basic-agent* nella simulazione. $\forall ba_i$:

$$\text{Global Opinion Metric Mean} = \frac{\sum_{i=1}^{N} ba_i\,\texttt{[opinion-metric]}}{N}$$

Get Most Influent A Nodes. La frazione dei nodi più influenti di tipo A, ordinati in modo decrescente per valore di *betweenness centrality* (oppure *degree centrality* oppure *page rank*)[3];

4.4 Modello di contrasto alla diffusione di fake news: un *super-agent*

Avendo ora un quadro generale di come è strutturata la simulazione su *NetLogo*, si può mostrare come il *super-agent* interviene durante la simulazione per contrastare l'andamento delle notizie false (tipo A). Innanzitutto, l'ambiente usato per permettere al modello di addestrarsi tramite DRL, è quello di *OpenAI Gym*. Il framework di *Gym* permette di creare degli ambienti in cui è possibile addestrare degli agenti tramite *RL*. Per creare l'ambiente bisogna implementare le seguenti funzioni:

- `step()`: Aggiorna l'ambiente con l'azione da eseguire ritornando: l'osservazione fatta dopo il compimento, la *reward* in relazione all'azione presa, se l'ambiente ha terminato o troncato la simulazione dovuta alle ultime azioni e informazione dall'ambiente riguardo lo *step* e informazioni di debug.
- `reset()`: Resetta l'ambiente a uno stato iniziale ed è necessario da compiere prima che venga eseguito uno *step*. Ritorna la prima osservazione dell'agente di un episodio e informazioni di debug.
- `render()`: Renderizza l'ambiente per aiutare a visualizzare cosa vede l'agente (non utilizzata nel nostro caso).
- `close()`: Chiude l'ambiente. Importante da utilizzare quando vengono utilizzati software esterni (come nel nostro caso).

[3]Personalizzabile dall'utente, e potenziale oggetto di sperimentazione.

Per inizializzare l'ambiente bisogna definire due parametri principali: lo spazio di osservazione e il numero di azioni. Il *super-agent* ha uno spazio di osservazione composto da tre dimensioni: la *global-cascade*, il *global-opinion-metric-mean* e i *most-influent-a-nodes*. Tutti e tre i parametri hanno come dominio $\mathbb{R}[0, 1]$. Lo spazio delle azioni è uno spazio discreto composto da quattro azioni che sono:

- `go`: che mette in pausa il *super-agent* e fa andare avanti la simulazione.
- `activate-warning`: che attiva la procedura di *Warning*, descritta nella Sezione 4.3.5.
- `activate-reiterate`: che attiva la procedura di *Reiterate*, descritta nella Sezione 4.3.5.
- `activate-static-b-nodes`: che attiva la procedura di *Static B Nodes*, descritta nella Sezione 4.3.5.

Una volta creato l'ambiente, devono essere definite le *reward*, che permettono al *super-agent* di capire se le azioni che sta compiendo stanno avendo effetti positivi o negativi sul network, in base alle osservazioni fatte sull'ambiente. Infatti, per calcolare le *reward*, si utilizzano i tre parametri di osservazione: `global-cascade` (sotto sezione 4.3.5), `global-opinion-metric-mean` (sotto sezione 4.3.5) e `most-influent-a-nodes` (sotto sezione 4.3.5).

Informalmente, il sistema di reward ideato è pensato per premiare il *super-agent* quando la `global-cascade` ≤ 0.50. Inoltre, il premio è tanto maggiore quanto l'efficacia dell'intervento del *super-agent*. In altre parole, se il *super-agent* ha preso una contromisura che ha portato effetti positivi nel network (es. `activate-warning`), allora questi avrà una ricompensa ancora maggiore rispetto a situazioni in cui non interviene, ma la `global-cascade` resta comunque al di sotto della soglia critica di 0.50. In queste casistiche, i motivi per cui il parametro `global-cascade` ≤ 0.50 sono da ricondurre alle impostazioni del modello simulativo e non direttamente al *super-agent*. Infine, in tutti i casi in cui non si ottengono effetti positivi, tendenzialmente la ricompensa è nulla.

Di seguito una più specifica disamina del sistema di reward. Il calcolo delle *reward* avviene dopo il primo *tick* della simulazione. Inizialmente viene calcolata l'`action-weight` che è uguale alla somma dell'inverso delle osservazioni su `global-opinion-metric-mean` e `most-influent-a-nodes`. Viene usato l'inverso delle osservazioni per via della definizione di questi parametri: più si avvicinano allo 0 e più le notizie vere si stanno diffondendo nella rete (cioè c'è una maggioranza di nodi blu, che supportano l'opinione

di tipo B). Successivamente, viene calcolato l'`action-result`, che è pari alla differenza tra la `global-cascade` attuale, cioè al *tick* in cui viene chiamata la funzione, e quella al *tick* precedente (o anche $tick - 1$).

$$\texttt{action-result} = \texttt{global-cascade}_{tick} - \texttt{global-cascade}_{tick-1}$$

Quindi `action-result` può trovarsi in tre casi diversi:

$$\begin{cases} \texttt{action-result} < 0 & \texttt{global-cascade} \text{ è migliorata rispetto al } tick - 1 \\ \texttt{action-result} == 0 & \texttt{global-cascade} \text{ invariata dal } tick - 1 \\ \texttt{action-result} > 0 & \texttt{global-cascade} \text{ è peggiorata rispetto al } tick - 1 \end{cases}$$

In base, quindi, ai possibili casi in cui può trovarsi l'`action-result`, ci sono due approcci di ricompensa diversi.

Se il *super-agent* ha eseguito un'azione (qualsiasi tra `activate-warning`, `activate-reiterate`, `activate-static-b-nodes`) che ha avuto un impatto positivo sulla `global-cascade` (`action-result` ≤ 0), viene ricompensato con $(1 + \texttt{action-weight} \cdot 0.5) - action - result$. In questo circostanza si è incluso anche l'eventualità in cui la `global-cascade` sia rimasta invariata.

Nel caso in cui la la `global-cascade` stia peggiorando (`action-result` > 0), il *super-agent* viene ricompensato con $(0 + \texttt{action-weight} \cdot 0.5) - action - result$. Nel caso in cui il *super-agent* abbia già eseguito un'operazione di *Warning* o *Static B Nodes*, la *reward* per queste azioni sarà ridotta, in quanto non è plausibile / pensabile / fattibile reiterare queste operazioni una volta eseguite. Ad esempio, il *Warning* una volta attivato, vale per tutti i *basic-agents* fino al termine della simulazione. Questo vale anche per *Static B Nodes*, un'azione molto "onerosa" da condurre nel mondo reale e che quindi deve essere usata con parsimonia. Nelle suddette situazioni (ripetizione di *Warning* o *Static B Nodes*), le *reward* sono calcolate in base alla `global-cascade`: se `global-cascade` > 0.5 la *reward* è 0, 1 altrimenti.

L'attesa da parte del *super-agent* è mappata con il *go* e ricompensata nel seguente modo: se `action-result` ≤ 0, la *reward* sarà 1, 0 altrimenti.

In generale, quindi, la *reward* minima ottenibile è 0, mentre la massima ottenibile è $(1 + \texttt{action-weight} \cdot 0.5) - action - result$.

Di seguito viene riportato il codice *Python* che mostra la funzione utilizzata per il calcolo delle *reward*.

```
1 def CalculateReward(self, action, tick, global_cascade,
2                      most_influent_a_nodes,
                                opinion_metric_mean, warning,
                                static_b):
```

```
reward = 0
action_weight = (1 - most_influent_a_nodes) + (1 -
                                opinion_metric_mean)

if (tick != 1):
    latest_global_cascade = self.global_cascade_values[tick-1]
    action_result = global_cascade - latest_global_cascade

    if (action == 1):
        if (warning == False):
            if (action_result <= 0):
                reward = (1 + action_weight) * 0.5
                reward -= action_result
            else:
                reward = (0 + action_weight) * 0.5
        else:
            if (global_cascade > 0.5):
                reward = 0
            else:
                reward = 1
    elif (action == 2):
        if (action_result <= 0):
            reward = (1 + action_weight) * 0.5
            reward -= action_result
        else:
            reward = (0 + action_weight) * 0.5
    elif (action == 3):
        if (static_b == False):
            if (action_result <= 0):
                reward = (1 + action_weight) * 0.5
                reward -= action_result
            else:
                reward = (0 + action_weight) * 0.5
        else:
            if (global_cascade > 0.5):
                reward = 0
            else:
                reward = 1
    else:
        if (action_result <= 0):
            reward = 1
        else:
            reward = 0
return reward
```

Una volta definito il sistema di *reward*, può essere mostrato come avviene l'addestramento del modello. Viene usato il DRL, descritto dettagliatamen-

te nella Sez. 3.3. Una differenza fondamentale tra il *Deep reinforcement Leaarning* e il *RL* classico, è l'implementazione della *Q-table*, che nel primo caso, viene sostituita con una rete neurale. Piuttosto che mappare uno stato-azione a una *q-value*, la rete neurale mappa gli stati in input in coppie $(azione, q - value)$. Una delle cose interessanti riguardo il DRL, è che il processo di apprendimento usa due reti neurali. Queste reti hanno la stessa architettura ma pesi differenti. Ogni N passi, i pesi dalla **main network** sono copiati sulla **target network**. Usando entrambe le reti si ottengono processi di apprendimento più stabili e aiuta l'algoritmo a imparare più efficacemente. Nel modello implementato,i pesi della *main network* sostituiscono quelli della *target network* ogni 100 passi.

La figura 4.2, mostra come la rete neurale mappa gli stati in input come coppie $(azione, q - value)$. Nell'esempio mostrato, ogni nodo in output (che rappresenta un'azione) contiene la *q-value* dell'azione, salvata come un tipo *float*.

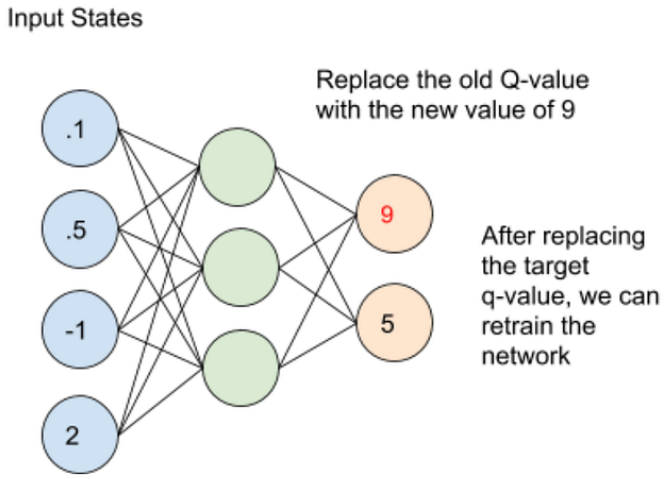


Figura 4.2: Aggiornamento della rete neurale con una nuova Temporal difference obiettivo, usando l'equazione di Bellman

In questa implementazione, la network principale e obiettivo sono composte da tre *layer* densamente connessi con delle funzioni *Relu* di attivazione. La caratteristica che si può notare è che c'è un'inizializzazione di tipo *HeUniform* e una *Huber loss function* per ottenere performance migliori. Il codice utilizzato per strutturare la rete è il seguente:

```
def agent(state_shape, action_shape):
    learning_rate = 0.001
    init = tf.keras.initializers.HeUniform()
    model = keras.Sequential()
    model.add(keras.layers.Dense
        (24, input_shape=state_shape, activation='relu',
                                kernel_initializer=init))
    model.add(keras.layers.Dense
        (12, activation='relu', kernel_initializer=init))
    model.add(keras.layers.Dense
        (action_shape, activation='linear', kernel_initializer=
                                init))
    model.compile(loss=tf.keras.losses.Huber(),
        optimizer=tf.keras.optimizers.Adam(lr=learning_rate),
                                metrics=['accuracy'])
    return model
```

Dopo aver scelto un'azione, l'agente deve eseguire l'azione e aggiornare la *main network* e la *target network* in relazione all'equazione di *Bellman*. Gli agenti che usano il DRL usano la *Replay Memory* per apprendere dall'ambiente in cui si trovano e per aggiornare i pesi delle network. Riassumendo, la *main network* rileva i campioni e si addestra su un *batch* delle esperienze passate ogni quattro passi. I pesi della *main network* sono poi copiati sui pesi della *target network* ogni 100 passi.

4.4.1 Motivazione e spiegazione delle azioni del *super agent*: un confronto con la realtà

Visto che le simulazioni ad agenti hanno come scopo quello di rappresentare fenomeni sociali e studiarne i pattern, bisogna capire su quale basi sono state scelte le azioni che compie il *super-agent* per contrastare le fake news. Bisogna capire quindi quali sono i problemi cognitivi associati alla disinformazione e quali sono le possibili soluzioni [38]. Due problemi che sono stati analizzati sono: il *Continued Influence Effect* e il *Familiarity Backfire Effect*, mostrati in Fig. 4.3. La spiegazione di questi problemi e le relative soluzioni sono esposti nella tabella 4.2.

Azione	Problema	Soluzione
`Warning`	**Familiarity Backfire Effect**: nonostante vengano esposti avvenimenti contrastanti le fake news, le persone continuano a credere alle notizie false.	Viene mostrato all'utente un avvertimento preventivo che la notizia che sta per leggere non ha fonti verificate, e quindi può essere una fake news.
`Reiterate`	**Continued Influence Effect**: la ripetizione di una notizia false ne incrementa la familiarità, rinforzandola.	senza rinforzare la notizia falsa, viene mostrata la notizia vera più volte così da favorire il cambio di opinione.
`Static B Agents`	Avere delle entità nella rete su cui si può fare affidamento per la trasmissione di notizie vere.	Avvisare le testate giornalistiche o enti importanti, sulla verità dei fatti. Avendo così dei diffusori di notizie vere a prescindere da quello che circola sulla rete..

Tabella 4.2: Associazione dei problemi cognitivi alle azioni del super-agent

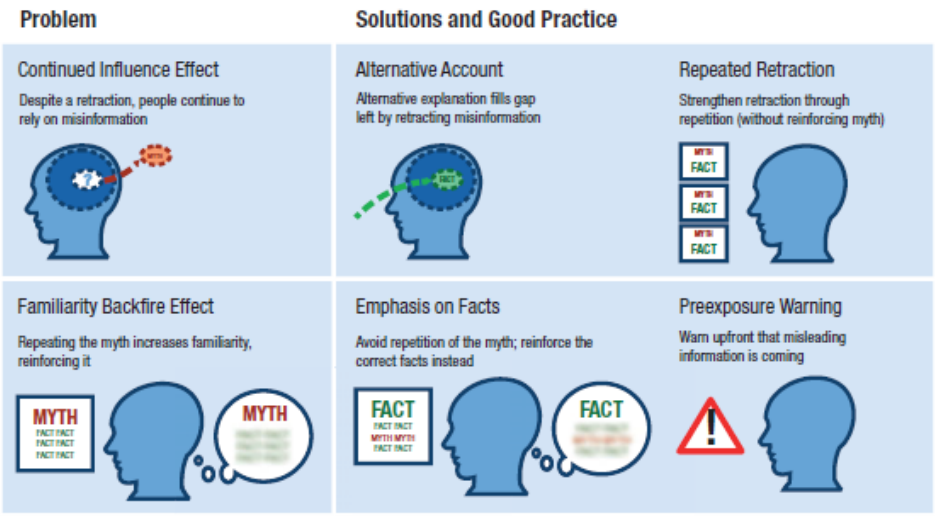


Figura 4.3: Grafico riassuntivo delle ricerche sulla disinformazione in ambito comunicativo [38]

Capitolo 5

Esperimenti

In questo capitolo, saranno mostrati gli esperimenti condotti sfruttando la strategia combinata vista nel Cap. 4. Si ricorda che l'obiettivo di questo lavoro è sfruttare un modello di simulazione ad agenti per studiare la diffusione virale delle fake news in un social network in presenza di una echo chamber e adoperare un metodo di DRL per analizzare i dati di queste simulazioni e sviluppare strategie di contenimento / contrasto alla diffusione virale delle fake news.

In termini generali, gli esperimenti possono essere suddivisi in due macro-blocchi: *(a)* esperimenti per studiare la diffusione virale delle fake news in un social network in presenza di una echo chamber, *(b)* esperimenti per studiare la diffusione virale delle fake news in un social network in presenza di una echo chamber e di un *super-agent* in grado di azionare strategie di contenimento / contrasto alla diffusione virale delle fake news. Per ciascuno di essi, per semplicità e chiarezza, saranno individuate delle sezioni il cui titolo rappresenta una domanda di ricerca a cui si tenta di rispondere. Sarà prima presentato l'esperimento – con configurazioni e parametri impostati – e le motivazioni alla base di esso, successivamente saranno illustrati e discussi i risultati ottenuti.

Il setting sperimentale è costituito da due ambienti: *Netlogo 6.2.0* e *Python 3.10.11*. Inoltre, le caratteristiche della macchina utilizzata per condurre gli esperimenti, sono esposti nella tabella 6.1. Le esecuzione dei test sono state eseguite in parallelo utilizzando l'*IDE Visual Studio Code*.

5.1 Studio della diffusione virale delle fake news in un social network in presenza di una echo chamber

In questa sezione, si presentano gli esperimenti condotti sfruttando solo una parte dell'architettura in Fig. 4.1. Nello specifico non si fa uso del *super agent*.

Prima di passare alla presentazione degli esperimenti condotti e dei risultati ottenuti è necessario riportare tutti i valori di `default` del modello simulativo. Di volta in volta, saranno mostrati solo i valori oggetto di indagine; tutti gli altri saranno lasciati al valore di `default`, mostrati nella tabella 5.1.

Attributo	**Valore Default**
`nb-nodes`	100
`total-ticks`	100
`links-to-use`	undirected
`network`	*Erdős–Rényi*
`k-value`	8
`standard-deviation`	2
P_o	0
P_n	0.40
`initial-opinion-metric-value`	0.5
`opinion-metric-step`	0.10
`activation Threshold` (Θ)	0.270
`echo-chamber-fraction`	0.20

Tabella 5.1: Valori di default degli esperimenti sul modello simulativo con la presenza di una echo chamber (tutte le descrizioni degli attributi, si trovano nella sotto sezione 4.3.4

Si ricorda che la *creduloneria* va intesa come $1 - \Theta$. In pratica, se i e j sono agenti e con $theta_i < theta_j$ di i è più credulone di j. Inoltre, nei seguenti esperimenti ci rifacciamo a [12] per l'interpretazione della Viralità alta/bassa. In [12], gli autori suggeriscono che Viralità ≤ 0.50 sono da considerarsi "basse", viceversa "alte".

Infine, ciascun esperimento è stato lanciato 100 volte (a meno di specificazioni).

5.1.1 Come cambia la Viralità al variare della network polarization e Θ (creduloneria)?

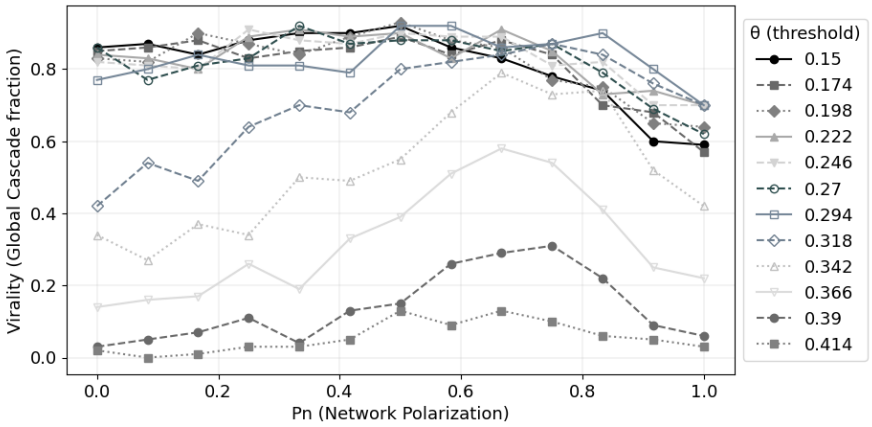


Figura 5.1: Andamento medio della Viralità delle fake news al variare di P_n e Θ. $P_n = \{0, 0.08, 0.16, 0.25, 0.33, 0.41, 0.50, 0.58, 0.66, 0.75, 0.83, 0.91, 1.00\}$.

Il grafico in Fig. 5.1 riporta sull'asse y la Viralità (dove la Viralità è calcolata nella Formula 4.1), sull'asse x il valore di P_n (*Network Polarization*, cioè la densità di connessioni interne al cluster). Ciascuna serie riportata corrisponde ad esperimenti condotti con un certo valore di creduloneria Θ (`activation Threshold`). Per rendere la discussione più fluida, qui di seguito ci si riferisce a V per indicare la Viralità.

Dagli esperimenti risulta che all'aumentare della creduloneria (cioè per Θ bassi) aumenta V, infatti per $\Theta < 0.318$ si hanno $0.7 \leq V \leq 0.9$ per $P_n < 0.8$. Per $\Theta \geq 0.318$ si hanno dei picchi di V per $0.6 \leq P_n \leq 0.8$ per poi diminuire successivamente. Con $P_n \geq 0.8$ la Viralità inizia a diminuire. Questo perché il cluster resta quasi sconnesso dal resto della rete e i nodi arancioni (coloro che supportano opinione di tipo A) possono influenzare perlopiù quelli appartenenti al cluster. Al diminuire della creduloneria, vedi $\Theta = \{0.366, 0.390, 0.414\}$, la Viralità resta sempre bassa. Per $0.6 \leq P_n \leq 0.8$ si tendono ad avere sempre degli innalzamenti di V, perché questi valori permettono alla echo chamber di avere una buona densità di connessione

all'interno di essa e allo stesso tempo non li isolano dal resto della rete, permettendo così di avere una diffusione rapida delle fake news.

5.1.2 Il numero di nodi nel network (*nb-nodes*) impatta la Viralità?

In questa sezione, si mostrano esperimenti al variare del numero di nodi nel network, `nb-nodes=` {100, 300, 400, 500}, della network polarization P_n, e al variare di alcuni valori di creduloneria interessanti (emersi dalla Fig. 5.1). In dettaglio, le `activation-threshold` prese in considerazione sono: $\Theta = \{0.270, 0.366, 0.414\}$. Ipotizziamo che la taglia del network possa influenzare la Viralità (es. i *basic-agents* con opinione A non riescono a far diventare virale la fake news in un network grande).

Con $\Theta = 0.270$ (vedi Fig. 5.2(a)) risulta che all'aumentare del numero di nodi appartenenti al network, tende a diminuire la Viralità. Si note inoltre, con $N > 100$ a partire da $P_n > 0.6$ si ha un innalzamento di V costante. Come visto in precedenza (vedi Fig. 5.1), questo intorno di valori per *Network Polarization* sembra essere particolarmente impattante sulla Viralità, un valore del parametro che da un lato rende coesa la echo chamber, dall'altro non causa un completo distacco dal resto del network.

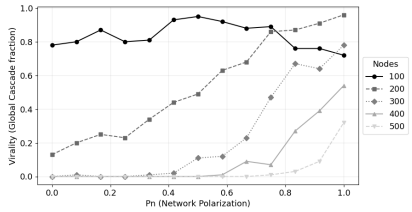
Inoltre, con un minor numero di nodi ($N < 200$) è più facile avere V alte, visto che ci sono meno collegamenti tra i nodi al di fuori della echo chamber.

Con $\Theta = 0.366$ (vedi Fig. 5.2(b)) la Viralità rimane bassa solo per $N > 100$. Aumentando il Θ, il network tende a rimanere in uno stato stabile, avendo un numero maggiore di nodi neutri (vedi Fig. 5.8 in Sezione 5.1.6). Quindi solo con $N = 100$ si riescono a ottenere delle oscillazioni rilevanti su V, che raggiungono i picchi con $0.5 \leq P_n \leq 0.8$.

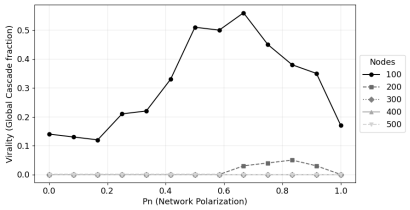
Lo stesso discorso vale con $\Theta = 0.414$ (vedi Fig. 5.2(c)): all'aumentare del Θ neanche le reti con un numero di nodi basso ($N < 200$) ottengono valori alti di V ($V < 0.15$). Questo accade perché aumentando N, aumentano i collegamenti tra i nodi che, per come è strutturato il modello diffusione, rende più difficile il cambio di opinione. Questo tipo di situazione rende difficile il diffondersi delle fake news.

Abbiamo voluto inoltre, effettuare i test al variare del numero di nodi del network come in [12]. Si imposta $\Theta = 0.270$ (vedi Fig. 5.2(d)) e `nb-nodes=` {75, 100, 125, 150, 175, 200, 225}. Si può notare che le Viralità tendono a rimanere più alte ($0.6 \leq V \leq 1$) per $P_n > 0.6$. Può essere confermata quindi l'intuizione che avere un numero troppo elevato di nodi fa diminuire la Viralità, soprattutto all'aumentare di Θ. Per come è strutturata la simulazione ed il

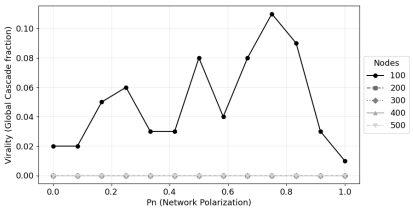
modello di diffusione, i *basic-agents* che supportano opinione A non riescono a diffondere la loro credenza alla maggioranza degli agenti del network.



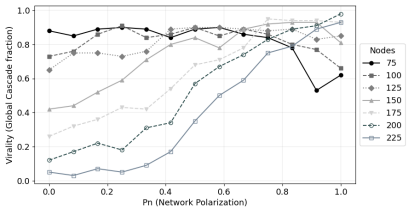
(a) $\Theta = 0.270$, `nb-nodes=` {100, 200, 300, 400, 500}.



(b) $\Theta = 0.366$, `nb-nodes=` {100, 200, 300, 400, 500}.



(c) $\Theta = 0.414$, `nb-nodes=` {100, 200, 300, 400, 500}.



(d) $\Theta = 0.270$, `nb-nodes=` {75, 100, 125, 150, 175, 200, 225} come in [12].

Figura 5.2: Andamento medio della Viralità delle fake news al variare della dimensione del network (numero di nodi, `nb-nodes`*) e P_n.*

5.1.3 Come varia la Viralità al variare della opinion polarization?

Finora abbiamo analizzato le simulazioni avendo valori di $P_o = 0$. In questo esperimento si analizzano alcune simulazioni che mostrano come varia la Viralità al variare di P_o. In questo tipo di esperimenti i nodi che fanno parte della echo chamber hanno `activation-treshold` $= \Theta - P_o$, rendendoli così più "creduloni", facilitando il cambio di opinione.

Con $P_o = 0.15$ (vedi Fig. 5.4), con valori di Θ più bassi ($\Theta < 0.390$) otteniamo V più alte ($0.6 \leq V \leq 1.0$, mediamente) soprattutto con P_n nel range $0.4 \leq P_n \leq 0.7$). Con $\Theta \geq 0.390$ si hanno V più alte ($0.2 \leq V \leq 0.7$, mediamente), fino $P_n < 0.6$ a differenze del test con $P_o = 0$ (vedi Fig. 5.3), dove per gli stessi range di valori hanno Viralità $0 \leq V \leq 0.2$.

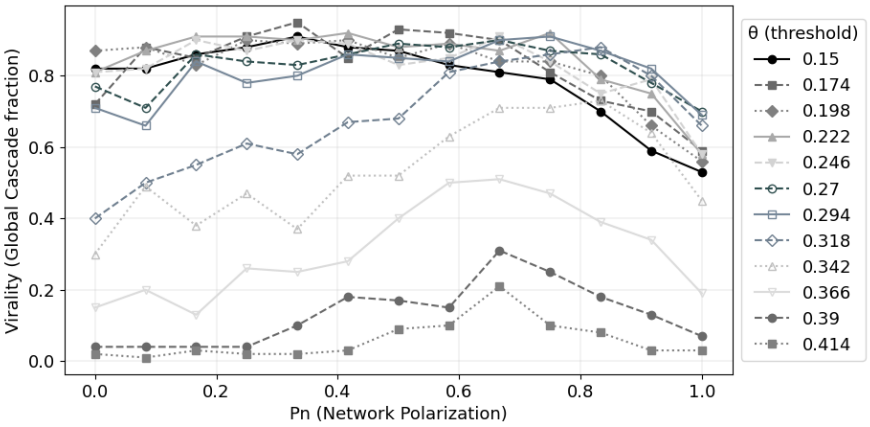


Figura 5.3: Andamento della Viralità delle fake news al variare di P_n e Θ, $P_o = 0$.

Con $P_n > 0.6$, si inizia ad avere un calo drastico della Viralità, questo perché anche se il cluster risulta coeso e allo stesso tempo ben collegato con il resto della rete, ora però i nodi all'interno della echo chamber sono più creduloni e possono essere influenzati più facilmente dall'opinione B. Avendo effettuato le simulazioni con $ECF = 0.20$, è plausibile che questa frazione di nodi non riesca a prendere il sopravvento sul resto del network così facilmente.

Con $P_o = 0.27$ (vedi Fig. 5.5), si tendono ad avere Viralità alte ($0.7 \leq V \leq 1$) con $\Theta < 0.39$, per $P_n < 0.8$. In questo esperimento bisogna considerare che fino a $\Theta \leq 0.270$, l'`activation-treshold` dei nodi che fanno parte della echo chamber è pari a 0. Quindi i cambiamenti si possono notare a partire da $\Theta > 0.270$, dove ci sono V alte ($0.55 \leq V \leq 0.9$) per $P_n < 0.6$. Per $P_n \geq 0.6$, si ha un calo di V, e addirittura si riescono ad avere $V = 0$ per $\Theta \geq 0.39$ e $P_n = 1$.

5.1.4 Come varia la Viralità variando la dimensione della echo chamber?

Ci siamo chiesti se la Viralità fosse influenzata dalla dimensione della echo chamber iniziale. Così abbiamo effettuato degli esperimenti in cui variamo tale dimensione tra 0.20 e 0.50.

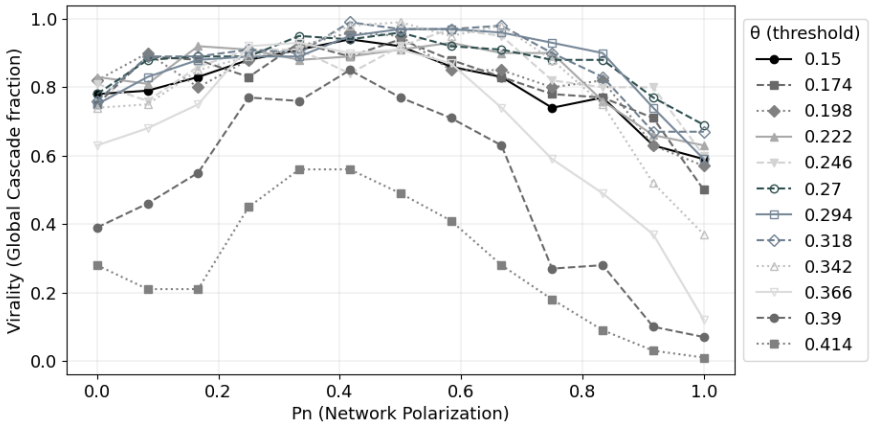


Figura 5.4: Andamento della Viralità delle fake news al variare di P_n e Θ, $P_o = 0.15$.

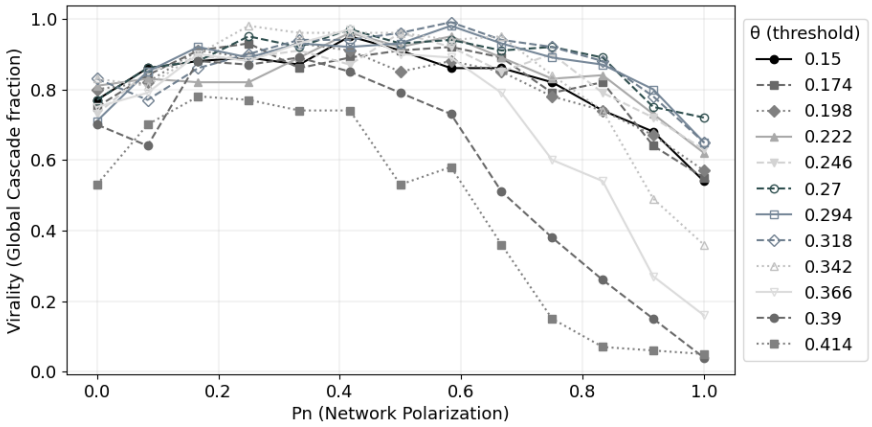


Figura 5.5: Andamento della Viralità delle fake news al variare di P_n e Θ, $P_o = 0.27$.

Rispetto ai test precedenti, in cui abbiamo impostato il valore di $ECF = 0.20$, vediamo ora come varia la Viralità all'aumentare di questo parametro mantenendo fisso $P_o = 0$. Per ciascun grafico (Figure 5.6(a):5.6(d)) si mostra la Viralità al variare dell'`echo-chamber-fraction`, su 100 esperimenti. Per rendere la discussione più fluida, qui di seguito ci si riferisce a ECF per

indicare la `echo-chamber-fraction`.

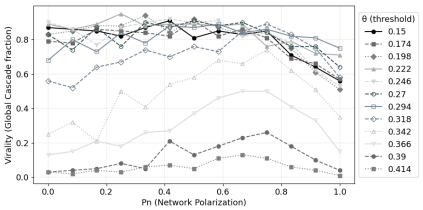
Con $ECF = 0.20$ (vedi Fig. 5.6(a)), poiché la echo chamber tende a preservare il suo stato, con nodi aventi opinioni di tipo A all'interno, possiamo notare valori alti di V ($0.75 \leq V \leq 0.9$) con P_n bassi ($0 \leq P_n \leq 0.5$). In particolare, con $\Theta < 0.342$ risulta un andamento più omogeneo di V, mantenendo valori alti ($V \geq 0.75$ mediamente) fino a $P_n \leq 0.7$ per poi decadere nei P_n rimanenti.

Possiamo notare che con $ECF = 0.30$ (vedi Fig. 5.6(b)), la Viralità tende a rimanere abbastanza alta ($0.8 \leq V < 0.1$ mediamente) per i $\Theta < 0.318$. Inoltre, questa osservazione vale per tutti i P_n presenti. Questo fa capire come la grandezza della echo chamber riesca a influenzare il resto della rete al crescere della grandezza. Possiamo notare, inoltre, che solo per $P_n > 0.8$ si può notare un abbassamento della Viralità, a differenza del primo test mostrato in cui decade già attorno a 0.7. Per i $\Theta \geq 0.318$, invece, si tende ad avere un aumento di V a partire da $P_n > 0.5$.

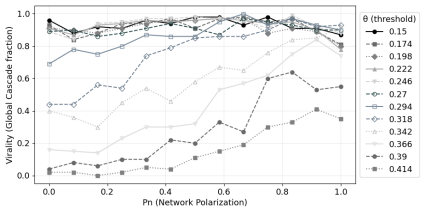
Bisogna valutare innanzitutto che con P_n bassi ($P_n < 0.5$), i nodi all'interno della echo chamber sono più connessi con il resto della rete, e viceversa con P_n alti ($P_n \geq 0.5$). Quindi un fattore fondamentale che fa variare la Viralità è proprio la dimensione del cluster; con cluster piccoli è più probabile che gli archi scelti a caso (il numero di archi scelti aumenta all'aumentare di P_n), durante la procedura di creazione della echo chamber (vedi sotto sezione 4.3.3), siano proprio quelli che devono essere eliminati e sostituiti con archi che collegano due nodi all'interno della echo chamber (la condizione di eliminazione è che esattamente un nodo appartenente alle due estremità faccia parte dell'echo chamber). Al crescere della dimensione del cluster, diminuisce la probabilità che questa procedura descritta sopra accada, proprio perché scegliendo a caso si può ricadere in un arco che già colleghi due nodi all'interno del cluster. Quindi al crescere della dimensione del cluster, non diventa così coeso con l'aumentare di P_n, portando ad un aumento di V.

Con $ECF = 0.40$ (vedi Fig. 5.6(c)), si riescono a non avere più abbassamenti di Viralità al variare di P_n, in virtù di quello che è stato detto precedentemente. Quello che possiamo notare è che per $\Theta \geq 0.342$, si tende comunque a mantenere una Viralità bassa ($0 \leq V < 0.5$) con $P_n < 0.7$.

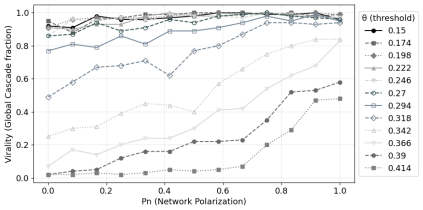
Con $ECF = 0.50$ (vedi Fig. 5.6(d)), in cui metà della rete fa parte dell'echo chamber, solo i $\Theta \geq 0.39$ riescono a non ottenere mai $V > 0.5$ per tutti i valori di P_n.



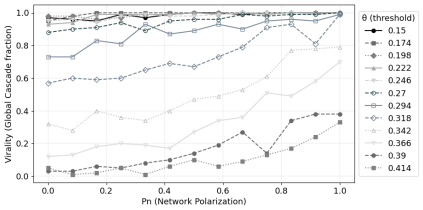
(a) `echo-chamber-fraction` = 0.20.



(b) `echo-chamber-fraction` = 0.30.



(c) `echo-chamber-fraction` = 0.40.



(d) `echo-chamber-fraction` = 0.50.

Figura 5.6: Andamento medio della Viralità delle fake news al variare di P_n *e* Θ*.* $P_n = \{0.00, 0.08, 0.16, 0.25, 0.33, 0.41, 0.50, 0.58, 0.66, 0.75, 0.83, 0.91, 1.00\}$.

5.1.5 Come varia la Viralità al variare dell'*opinion-metric-step*?

In questa sezione verranno esaminate più simulazioni al variare dell' `opinion-metric-step` (vedi sezione 4.3.5) su tre valori di Θ che sono risultati più interessanti dai risultati precedenti. Questo parametro determina con quale velocità i *basic-agents* tendono a cambiare opinione, più i valori sono alti più è rapido il cambiamento. Per ciascun grafico (Figure 5.7-5.9) si mostra la Viralità al variare dell'`opinion-metric-step`. Nei grafici, la serie rappresentante i valori della Viralità con `opinion-metric-step` = 0.10, è colorata in **verde** poiché si tratta del valore di `default` degli esperimenti.

Con $\Theta = 0.270$ (vedi Fig. 5.7), si può notare come non ci siano molte differenze tra i vari `opinion-metric-step`; in particolare l'unico valore che si differenzia è 0.16 che raggiunge picchi di Viralità più alti rispetto ai restanti, infatti per $P_n \geq 0.2$ si hanno V, $0.90 \leq V \leq 0.98$. Nonostante ciò tutte i valori di valori di V sono maggiori di 0.65, favorendo la diffusione delle fake news. Infatti la V minima si ha con `opinion-metric-step` = 0.01, cioè il più basso possibile e con $P_n = 1$, la massima possibile.

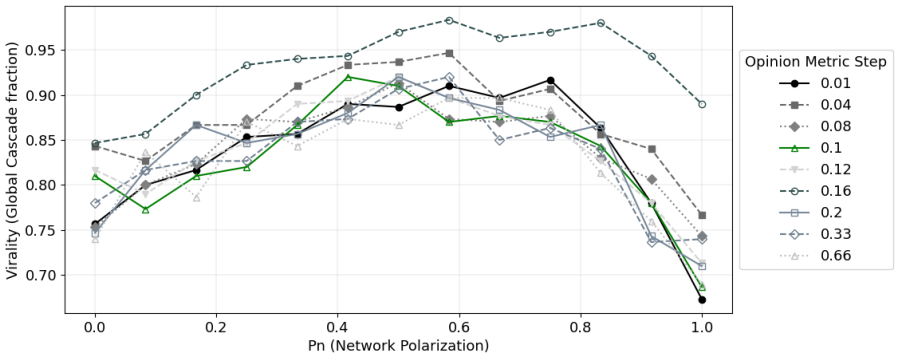


Figura 5.7: Andamento della Viralità delle fake news al variare di P_n e ***opinion-metric-step****, $\Theta = 0.270$. La serie rappresentante l'* `opinion-metric-step` $= 0.10$ *è di colore* ***verde*** *visto che rappresenta il valore di* ***default*** *degli esperimenti.*

Con $\Theta = 0.342$ (vedi Fig. 5.8), anche qui si può notare un andamento omogeneo di V, tranne con `opinion-metric-step` $= 0.01$, che permette di raggiungere Viralità più basse rispetto al resto ($0.13 \leq V \leq 0.50$). Questo perché all'aumentare di Θ diminuisce la Viralità visto che le notizie si diffondono meno facilmente.

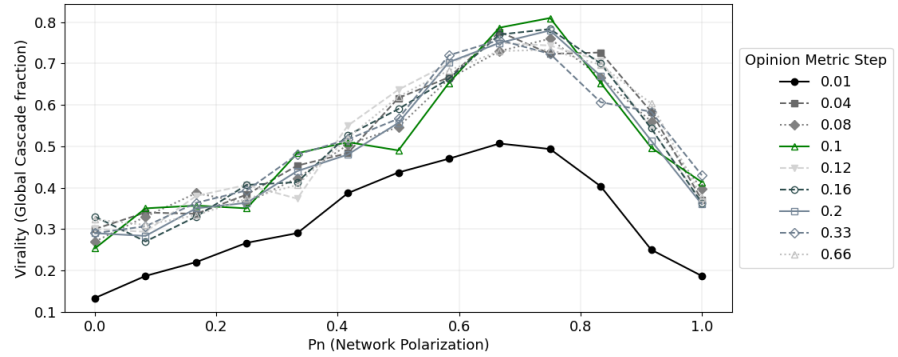


Figura 5.8: Andamento della Viralità delle fake news al variare di P_n e ***opinion-metric-step****, $\Theta = 0.342$. La serie rappresentante l'* `opinion-metric-step` $= 0.10$ *è di colore* ***verde*** *visto che rappresenta il valore di* ***default*** *degli esperimenti.*

Infine, con $\Theta = 0.414$ (vedi Fig. 5.9), anche qui si può notare un andamento omogeneo tranne con `opinion-metric-step` $= 0.01$, che permette di raggiungere Viralità più basse rispetto al resto, per lo stesso motivo precedente. Inoltre, i valori medi di V sono $0.003 \leq V \leq 0.16$,

Quindi, sia in Fig. 5.8 che in Fig. 5.9, l'unico `opinion-metric-step` che discosta dalla media è 0.01. Questo perché è il valore minimo a cui può essere impostato e a parità di *tick* della simulazione è normale che questo valori porti a Viralità minori rispetto agli altri `opinion-metric-step`, non avendo abbastanza tempo per far cambiare opinione ai nodi.

In tutte e tre le figure si può notare che all'aumentare di Θ diminuisce la Viralità e i picchi massimi si hanno sempre con $0.66 \leq P_n \leq 0.83$.

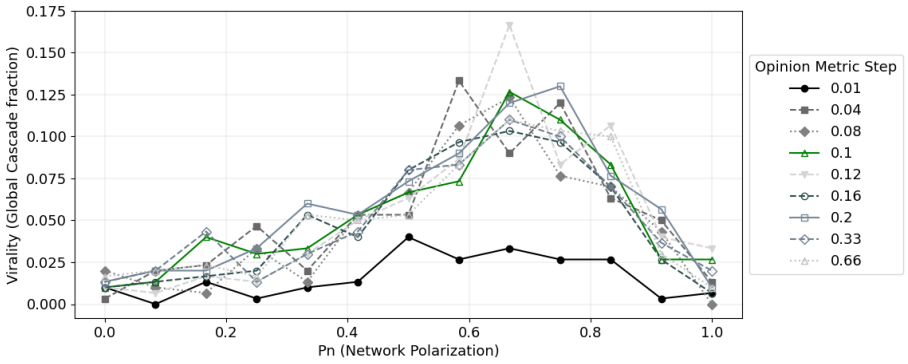


Figura 5.9: Andamento della Viralità delle fake news al variare di P_n e ***`opinion-metric-step`****, $\Theta = 0.414$. La serie rappresentante l'* `opinion-metric-step` $= 0.10$ *è di colore* ***verde*** *visto che rappresenta il valore di* ***`default`*** *degli esperimenti.*

5.1.6 Quanti nodi cambiano opinione col passare del tempo al variare di P_n e Θ?

In questa sezione si analizza come i nodi cambiano opinione con lo scorrere del tempo, cioè all'avanzare dei *tick* della simulazione. In particolare, ogni 10 tick vengono catturate le informazioni riguardanti: *(i)* il numero di nodi di tipo A (arancioni), *(ii)* il numero di nodi di tipo B (blu) e *(iii)* il numero di nodi neutri (grigi). Sono state eseguiti tre esperimenti al variare del $\Theta = \{0.270, 0.342, 0.414\}$. Per ciascun grafico (Figure 5.10(a):5.10(c)) si mostrano la media dei nodi di un certo tipo e la deviazione standard su 300 esperimenti. Quindi ogni barra impilata rappresenta μ e STD. Il colore delle

barre corrisponde al colore dei nodi nell'ambiente NetLogo per semplicità di lettura, cioè la barra blu corrisponde alla media (μ) dei nodi con opinione B. Le linee verticali nere al centro delle barre corrispondono alla rispettiva deviazione standard (STD).

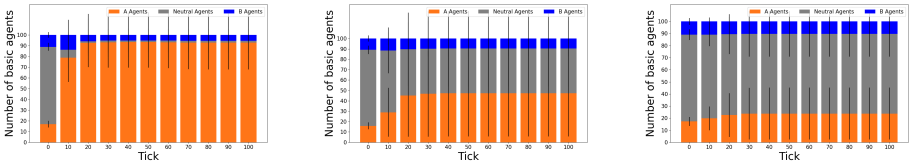
Per rendere la discussione più fluida, qui di seguito ci si riferisce a μ_A e STD_A per indicare rispettivamente la media dei nodi con opinione A e la rispettiva deviazione standard. Vale lo stesso principio per i nodi con opinione B (μ_B e STD_B) e i nodi neutrali (μ_N e STD_N).

Nella Fig. 5.10(a), con un $\Theta = 0.270$, si può notare come già dopo 20 *tick* i nodi della simulazione tendono a convergere verso uno stato che non cambia fino al termine. Infatti, dopo il 20^{mo} *tick*, $\mu_A \geq 90$ tendenzialmente, mentre μ_B oscilla tra $5 \leq \mu_B \leq 6$ e, infine, $\mu_N < 2$. Le deviazioni standard, dopo il 30^{mo} *tick*, dei nodi con opinione A e con opinione B hanno valori $20 \leq STD_A, STD_B \leq 25$. Allo stesso tempo, i nodi neutri hanno meno oscillazione ($10 \leq STD_N \leq 11$). Una volta che la rete raggiunge uno stato stabile è difficile riuscire a ribaltarne gli equilibri. Visto che il Θ è basso, i nodi neutri restano in una numerosità limitata all'avanzare dei *tick*, i nodi di tipo A, prendono il sopravvento più rapidamente e più facilmente, ottenendo quasi sempre delle `global-cascade` > 0.5.

Nella Fig. 5.10(b), con un $\Theta = 0.342$, la rete tende a mantenere uno stato meno stabile, perché se si osservano le deviazioni standard rilevate, si può notare come queste siano elevate ($39 \leq STD_A \leq 42$, $15 \leq STD_B \leq 18$, $35 \leq STD_N \leq 37$) dal *tick* 30 a seguire. Questo significa che fino al termine della simulazione ci sono molti cambiamenti di opinioni e non si raggiunge facilmente uno stato stabile. In questo caso i nodi neutri hanno dei valori μ_N simili a μ_A. Visto che il Θ scelto è impostato a un valore compreso intermedio (vedi anche esperimenti in Sezione 5.1.1), le opinioni si diffondono meno facilmente, ma si riescono comunque a ottenere delle `global-cascade`> 0.5.

Nella Fig. 5.10(c), con un $\Theta = 0.414$, il network tende a mantenere uno stato stabile già allo start; basti osservare i valori medi rilevati: $17 \leq \mu_A \leq 24$, $10 \leq \mu_B \leq 11$, $65 \leq \mu_N \leq 72$. Ciò è dovuto al Θ elevato che rende più difficile il cambio di opinione. A supporto di questa osservazione, infatti, alcuni nodi che sono neutri / neutrali all'inizio della simulazione, rimarranno tali fino alla fine. Si può notare che le notizie di tipo B hanno più difficoltà a diffondersi ($\mu_B \leq 11$ tendenzialmente). Ciò è dovuto al fatto che nella rete è presente una echo chamber di taglia pari al 20% dei nodi totali e con un $P_n = 0.40$, che facilita la diffusione della notizia A all'interno di essa e contemporaneamente rende più difficoltoso ai nodi con opinione B la diffusione delle notizie vere.

Questo pattern si può riscontrare in tutte e tre le simulazioni e rispecchia l'andamento reale delle notizie all'interno di una rete, in quanto le notizie vere si diffondono meno facilmente delle fake news (vedi Cap. 2).



(a) Con $\Theta = 0.270$. (b) Con $\Theta = 0.342$. (c) Con $\Theta = 0.414$.

Figura 5.10: Andamento del numero di nodi che cambiano opinione al variare di P_n e Θ. $P_n = \{0, 0.08, 0.16, 0.25, 0.33, 0.41, 0.50, 0.58, 0.66, 0.75, 0.83, 0.91, 1.00\}$. Eseguite 300 run.

5.2 Studio della diffusione virale delle fake news in un social network in presenza di una echo chamber e di un *super-agent* che contrasta il fenomeno

Per ogni test con il *super-agent* è stato utilizzato il DRL (Sez. 3.3). Sia per l'addestramento che per il testing vengono analizzati 100 episodi, dove ogni episodio è una simulazione di 100 *tick*. Un numero di esperimenti così elevato richiede un tempo computazionale piuttosto lungo. Per questo motivo, in questo lavoro di tesi, si sono effettuati una serie di esperimenti preliminari scelti per la loro importanza rispetto alle domande di ricerca.

Prima di passare alla presentazione degli esperimenti condotti e dei risultati ottenuti è necessario riportare tutti i valori di `default` del modello simulativo. Di volta in volta, saranno mostrati solo i valori oggetto di indagine; tutti gli altri saranno lasciati al valore di `default`, mostrati nella tabella 5.2.

In questi test, inoltre, è presente il parametro `sa-delay` che indica dopo quanti *tick* di simulazione può intervenire il *super-agent*. L'idea è che il *super-agent* non dovrebbe poter intervenire ad ogni *tick* per realizzare i suoi scopi, quindi viene "rallentato".

I valori di `default` del *super-agent* sono accoppiati con i valori della simulazione senza la sua presenza, mostrati nella tabella 5.1.

Attributo	Valore Default
`node-range`	0.10
`node-range-static-b`	0.05
`global-warning`	*True*
`choose-method`	*degree*
`warning-impact`	0.10
`warning-impact-neutral`	0.30
`sa-delay`	2

Tabella 5.2: Valori di default degli esperimenti sul modello simulativo con la presenza di un super-agent

L'efficacia del *super-agent* è misurata in termini di Viralità, più è bassa più il *super-agent* è stato efficace nelle sua azioni di constrasto.

5.2.1 Qual è l'impatto del *super-agent* sulla Viralità al variare della network polarization e Θ (creduloneria)?

In questa sezione, si studia il fenomeno della diffusione delle fake news (Viralità V) al variare di P_n e Θ, quando un *super-agent* può effettuare azioni di contenimento / contrasto a diverse "velocità".

In particolare, si mostrano una serie di risultati al variare di `sa-delay`, cercando di comprendere il massimo valore per `sa-delay` tale che le azioni del *super-agent* abbiano un impatto concreto sui valori di Viralità (nello specifico, $V \leq 0.5$)). I risultati di tali esperimenti saranno utilizzati come "baseline" per i test successivi.

Sono stati considerati i valori di creduloneria più interessanti emersi dai test precedenti, si veda Sezione 5.1, cioè $\Theta = \{0.270, 0.342, 0.414\}$.

I risultati sono riportati in Fig. 5.11.

Con *sa-delay* $= 5$ (Fig. 5.11(a)) e *sa-delay* $= 4$ (Fig. 5.11(b)), si ottengono risultati molti simili in termini di V. Infatti, per entrambi, con $\Theta = 0.270$, la Viralità ha come range di valori $0.6 \leq V \leq 0.8$ e con $\Theta = 0.342$ si ottiene $0.1 \leq V \leq 0.45$.

Con *sa-delay* $= 2$ e con $\Theta = 0.270$ la Viralità ha come range di valori $0.2 \leq V \leq 0.6$ e con $\Theta = 0.342$ si ha $0.05 \leq V \leq 0.3$.

Quindi, emerge che al diminuire del valore di `sa-delay` si ottengono effetti più tangibili nel contrasto della diffusione della fake news. Se l'agente riesce ad

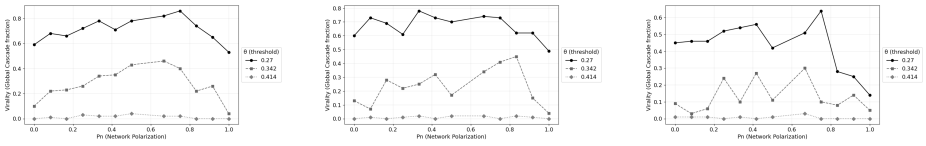
intervenire più frequentemente nel network, la Viralità resta tendenzialmente sotto la soglia 0.5.

A prescindere dal valore di `sa-delay` e dal valore di P_n, con $\Theta = 0.414$ si ottengono Viralità sempre prossime allo zero. In altre parole, uno scetticismo più elevato nei confronti delle notizie unito agli interventi del *super-agent*, fa si che la fake news "non diventi mai" virale.

Per capire quale sia stato l'impatto del *super-agent* sulla Viralità verrà utilizzata la variazione percentuale[1].

Facendo un confronto con gli esperimenti visti in Sezione 5.1, prendiamo in considerazione la Fig. 5.1. In quest'ultimo esperimento con $\Theta = 0.270$ si ha $\mu_V = 0.81$, con $\Theta = 0.342$ si ha $\mu_V = 0.51$ e infine con $\Theta = 0.414$ si ha $\mu_V = 0.05$. Confrontiamo questi valori con quelli nella figura dell'esperimento di questa sezione (Fig. 5.11).

Con $\Theta = 0.270$, considerando *sa-delay* $= 5$ e con *sa-delay* $= 4$, si ha un miglioramento di V del 14%. Con *sa-delay* $= 2$, si ha un miglioramento di V del 46%. Con $\Theta = 0.342$, considerando *sa-delay* $= 5$ e con *sa-delay* $= 4$, si ha un miglioramento di V del 50%. Con *sa-delay* $= 2$, si ha un miglioramento di V del 74%. Con $\Theta = 0.414$, a prescindere dal valore di `sa-delay`, si è avuto un miglioramento del 50%. Questi risultati sono riassunti nel grafico a barre in Fig. 5.12.



(a) Con `sa-delay` $= 5$. (b) Con `sa-delay` $= 4$. (c) Con `sa-delay` $= 2$.

Figura 5.11: Andamento medio della Viralità delle fake news, in presenza di un super-agent, al variare di P_n e Θ. $P_n = \{0, 0.08, 0.16, 0.25, 0.33, 0.41, 0.50, 0.58, 0.66, 0.75, 0.83, 0.91, 1.00\}$, $\Theta = \{0.270, 0.342, 0.414\}$.

[1]Quest'ultima è un valore percentuale che esprime la differenza tra il valore finale e il valore iniziale di una grandezza (nel nostro caso la Viralità) in termini percentuali, considerando come valore di riferimento quello iniziale. Si calcola con la formula: $P = \frac{F-I}{I} \times 100\%$, dove F rappresenta il valore finale e I il valore iniziale. Nei nostri esperimenti, il valore iniziale è la media delle Viralità (μ_V) in un range di P_n di un test senza *super-agent*. Il valore finale si ottiene con la stessa procedura ma prendendo in considerazione le medie dei valori di Viralità nei test con il *super-agent*.

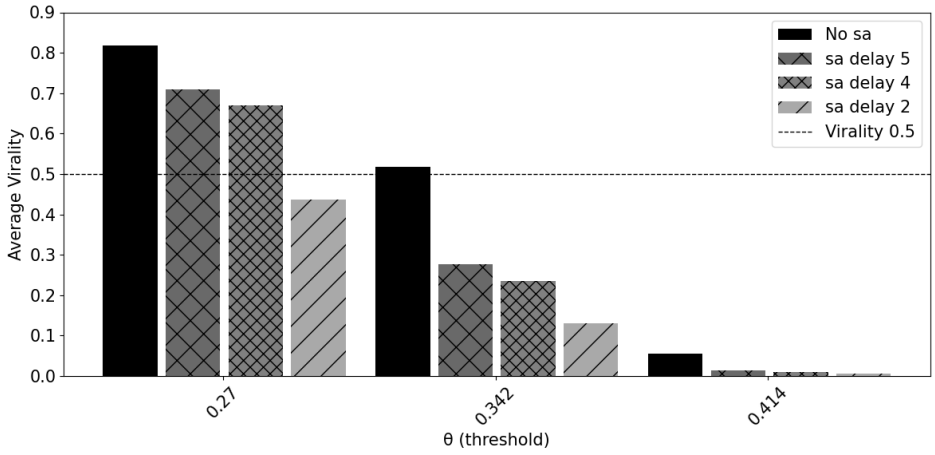


Figura 5.12: Media della Viralità raggiunta al variare di Θ (asse x) negli esperimenti senza super-agent ("No sa") e con super-agent al variare di `sa-delay`.

In conclusione, visto che con *sa-delay* = 2 il *super-agent* ha ottenuto risultati migliori in termini di V, abbiamo deciso di usare questo valore per i nostri seguenti esperimenti.

5.2.2 Qual è l'impatto del *super-agent* sulla Viralità al variare della opinion polarization?

In questa sezione si analizza, al variare dell'*Opinion Polarization*, come il *super-agent* riesce ad impattare la Viralità.

I risultati sono riportati in Fig. 5.13.

Impostare $P_o = 0.15$ (Fig. 5.13(a)) non impatta particolarmente sull'efficacia del *super-agent* con $\Theta = \{0.342, 0.414\}$, in quanto esso riesce a mantere la Viralità sotto la soglia di 0.5 (fatta eccezione per $0.4 \leq P_n \leq 0.5$ con $\Theta = 0.342$). Con $P_o = 0.27$ (Fig. 5.13(b)), in termini generali, il *super-agent* non riesce a contrastare la Viralità. Con $P_n > 0.8$ il *super-agent* riesce a combattere la diffusione della fake news.

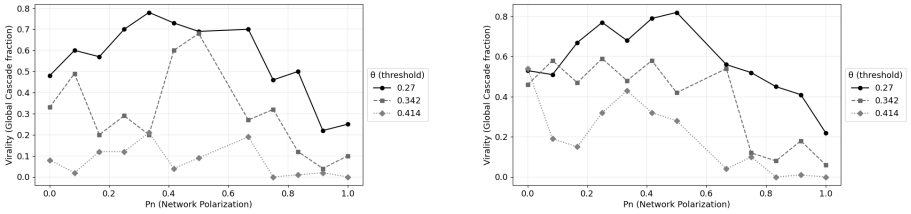
Consideriamo $P_o = 0.15$ (Fig. 5.13(a)). Analizziamo i miglioramenti in termini di V confrontando i valori ottenuti con quelli dello stesso test senza la presenza del *super-agent* (Fig. 5.4).

Con $\Theta = 0.270$ si ha $\mu_V = 0.87$, mentre per la Fig. 5.13(a), si ha $\mu_V = 0.55$. Si è avuto un miglioramento del 36% su V. Con $\Theta = 0.342$ si

ha $\mu_V = 0.81$, mentre per la Fig. 5.13(a), si ha $\mu_V = 0.30$. Si è avuto un miglioramento del 62% su V. Infine, Con $\Theta = 0.414$ si ha $\mu_V = 0.28$, mentre per la Fig. 5.13(a), si ha $\mu_V = 0.07$. Si è avuto un miglioramento del 75% su V.

Con $P_o = 0.27$ (Fig. 5.13(b)). Con $\Theta = 0.270$ per la Fig. 5.5 si ha $\mu_V = 0.87$, mentre per la Fig. 5.13(b), si ha $\mu_V = 0.57$. Si è avuto un miglioramento del 34% su V. Con $\Theta = 0.342$ per la Fig. 5.5 si ha $\mu_V = 0.82$, mentre per la Fig. 5.13(b), si ha $\mu_V = 0.37$. Si è avuto un miglioramento del 54% su V. Con $\Theta = 0.414$ per la Fig. 5.5 si ha $\mu_V = 0.46$, mentre per la Fig. 5.13(b), si ha $\mu_V = 0.19$. Si è avuto un miglioramento del 58% su V.

Se vogliamo prendere in considerazione solo i $P_n > 0.7$ si può notare in Fig. 5.13 come il *super-agent* sia riuscito ad avere un impatto maggiore su V. Infatti con $\Theta = 0.270$ e $P_o = 0.27$, si ha $0.06 \leq V \leq 0.2$. In Fig. 5.4, per $P_n > 0.7$ si ha $0.65 \leq V \leq 0.9$. In questo caso, si è avuto un miglioramento dell'83% su V. Questi risultati sono riassunti nel grafico a barre in Fig. 5.14.



(a) $P_o = 0.15$. (b) $P_o = 0.27$.

Figura 5.13: Andamento medio della Viralità delle fake news, in presenza di un super-agent, al variare di P_n e Θ. $P_n = \{0, 0.08, 0.16, 0.25, 0.33, 0.41, 0.50, 0.58, 0.66, 0.75, 0.83, 0.91, 1.00\}$, $\Theta = \{0.270, 0.342, 0.414\}$.

5.2.3 Qual è l'impatto del *super-agent* sulla Viralità al variare del parametro node-range-static-b?

In questa sezione si analizza, al variare di `node-range-static-b`, come il *super-agent* riesca ad impattare la Viralità. Questo attributo permette scegliere la frazione dei nodi su cui poi il *super-agent*, tramite l'operazione *Static B Nodes* (Sez. 4.3.5), possa forzare un *basic-agent* a sostenere l'opinione B fino alla fine della simulazione. Questa procedura, se applicata in scenari reali, può risultare onerosa da attuare (in quanto mappa

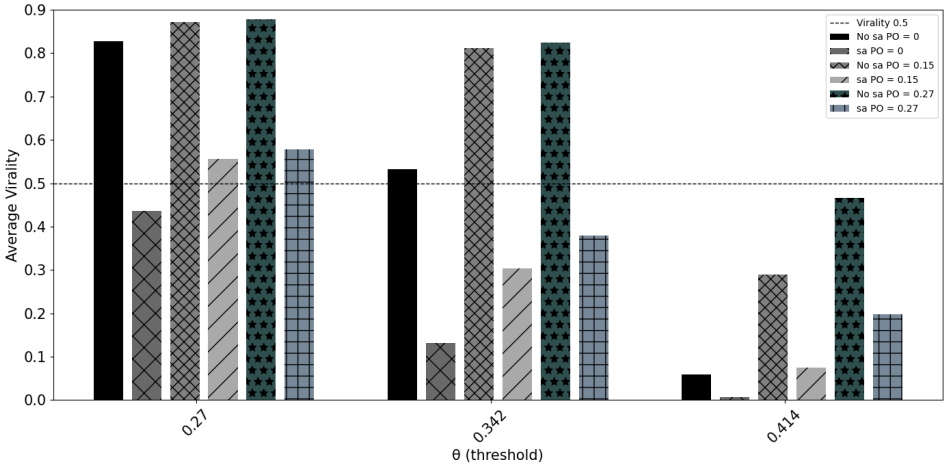


Figura 5.14: Media della Viralità raggiunta al variare di Θ (asse x) negli esperimenti senza super-agent ("No sa") e con super-agent al variare di P_o.

imposizioni alle testate giornalistiche o enti importanti alla diffusione di notizie vere) e quindi per i nostri esperimenti abbiamo deciso di usare un valore molto basso (`node-range-static-b` $= 0.05$). Però ci siamo chiesti quanto effettivamente questa procedura possa impattare V con valori più alti e quindi in questo esperimento abbiamo usato i seguenti valori: `node-range-static-b` $= \{0.05, 0.10, 0.20\}$. I risultati sono riportati in Fig. 5.15.

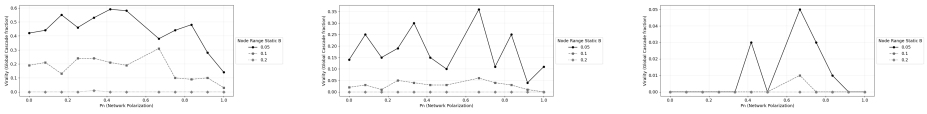
In termini generali, si nota che quando il *super-agent* ha accesso al 20% dei nodi più influenti del network la Viralità non si alza mai oltre la soglia di 0.5. Ad ogni modo, la Viralità resta molto bassa anche con `node-range-static` $- b = 0.05$ quando il $\Theta = \{0.342, 0.414\}$. Soltanto nel caso di $\Theta = 0.270$ e `node-range-static-b` $= 0.05$, si hanno casi in cui $V > 0.5$.

Consideriamo $\Theta = 0.270$ (Fig. 5.15(a)). Analizziamo i miglioramenti in termini di V confrontando i valori ottenuti con quelli dello stesso test senza la presenza del *super-agent* (Fig. 5.1). Con $\Theta = 0.270$ si ha $\mu_V = 0.81$, mentre per la Fig. 5.15(a), per `node-range-static-b` $= 0.05$ si ha $\mu_V = 0.44$. Si è avuto un miglioramento del 45% su V. Per `node-range-static-b` $= 0.10$ si ha $\mu_V = 0.17$.Si è avuto un miglioramento del 79% su V. Infine, per `node-range-static-b` $= 0.2$ si ha $0\mu_V = 0.02$. Si è avuto un miglioramento del 98% su V. Questi risultati sono riassunti nel grafico a barre in Fig. 5.16.

Con $\Theta = 0.342$ e con $\Theta = 0.414$, si ottengono valori sempre vicini allo zero

a prescindere dal `node-range-static-b` scelto. Questo è dovuto al fatto che i nodi cambiano opinione con più difficoltà e quindi avere un'operazione che permette di forzare un'opinione risulta ancora più impattante.

Quindi, abbiamo visto come con `node-range-static-b` $= 0.1$ e con `node-range-static-b` $= 0.2$, si ottengono degli impatti su V troppo elevati, a dimostrare quanto `node-range-static-b` in una simulazione possa impattare la diffusione delle notizie di tipo B all'interno della rete.



(a) Con $\Theta = 0.270$. (b) Con $\Theta = 0.342$. (c) Con $\Theta = 0.414$.

Figura 5.15: Andamento medio della Viralità delle fake news, in presenza di un super-agent, al variare di P_n e `node-range-static-b`. $P_n = \{0, 0.08, 0.16, 0.25, 0.33, 0.41, 0.50, 0.58, 0.66, 0.75, 0.83, 0.91, 1.00\}$, `node-range` $= \{0.05, 0.1, 0.2\}$

.

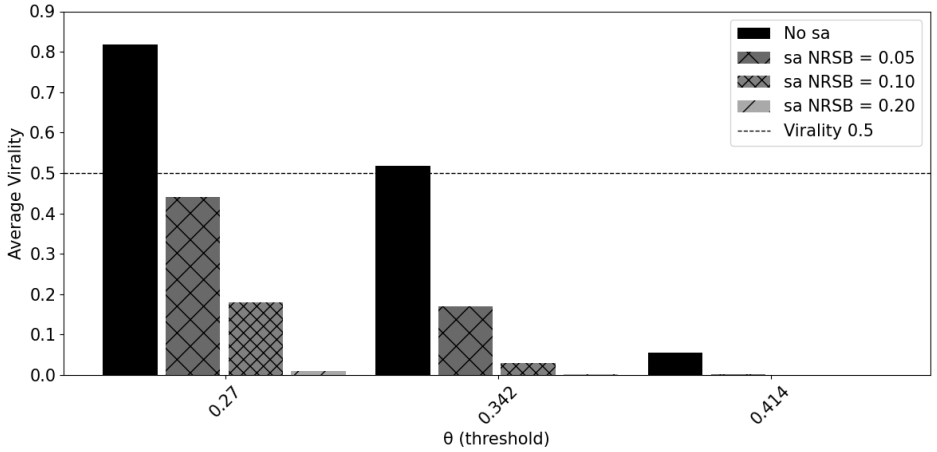


Figura 5.16: Media della Viralità raggiunta al variare di Θ (asse x) negli esperimenti senza super-agent ("No sa") e con super-agent al variare di `node-range-static-b` (NRSB).

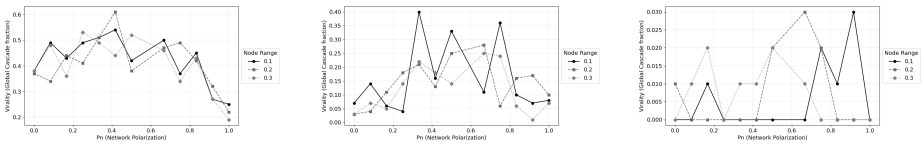
5.2.4 Qual è l'impatto del *super-agent* sulla Viralità al variare del parametro node-range?

In questa sezione si analizza, al variare di `node-range`, come il *super-agent* riesca ad impattare la Viralità. Questo attributo permette scegliere la frazione dei nodi su cui poi il *super-agent* dovrà intervenire tramite l'operazione *Reiterate* o *Warning*, se il *Warning* non fosse globale (Sez. 4.3.5). In questo esperimento abbiamo usato i seguenti valori: `node-range` $= \{0.10, 0.20, 0.30\}$.

I risultati sono riportati in Fig. 5.17. In termini generali, si nota che a prescindere dalla frazione dei nodi su cui interviene il *super-agent*, la Viralità non subisce cambiamenti notevoli per i diversi valori di Θ considerati. Questo ci fa capire che l'operazione di *Reiterate* ha meno impatto del *Warning*, in quanto il primo ha una influenza sui *basic-agents* di durata minore rispetto al *Warning*, che permane fino alla fine della simulazione.

Consideriamo $\Theta = 0.270$ (Fig. 5.17(a)). Analizziamo i miglioramenti in termini di V confrontando i valori ottenuti con quelli dello stesso test senza la presenza del *super-agent* (Fig. 5.1). Con $\Theta = 0.270$ si ha $\mu_V = 0.81$, mentre per la Fig. 5.17(a), per `node-range` $= 0.10$ si ha $\mu_V = 0.42$. Si è avuto un miglioramento del 48% su V. Per `node-range` $= 0.2$ si ha $\mu_V = 0.41$. Si è avuto un miglioramento del 49% su V. Infine, per `node-range` $= 0.3$ si ha $\mu_V = 0.40$. Si è avuto un miglioramento del 50% su V. Questi risultati sono riassunti nel grafico a barre in Fig. 5.18.

Con $\Theta = 0.414$ si ottengono valori sempre vicini allo zero a prescindere dal `node-range` scelto. Questo è dovuto al fatto che i nodi cambiano opinione con più difficoltà e quindi avere un'operazione che permette sposare più facilmente l'*opinion-metric* porta a una diffusione più agevolata delle notizie di tipo B.



(a) Con $\Theta = 0.270$ (b) Con $\Theta = 0.342$ (c) Con $\Theta = 0.414$

Figura 5.17: Andamento medio della Viralità delle fake news, in presenza di un super-agent, al variare di P_n e `node-range`. $P_n = \{0, 0.08, 0.16, 0.25, 0.33, 0.41, 0.50, 0.58, 0.66, 0.75, 0.83, 0.91, 1.00\}$, `node-range-static-b` $= \{0.1, 0.2, 0.3\}$.

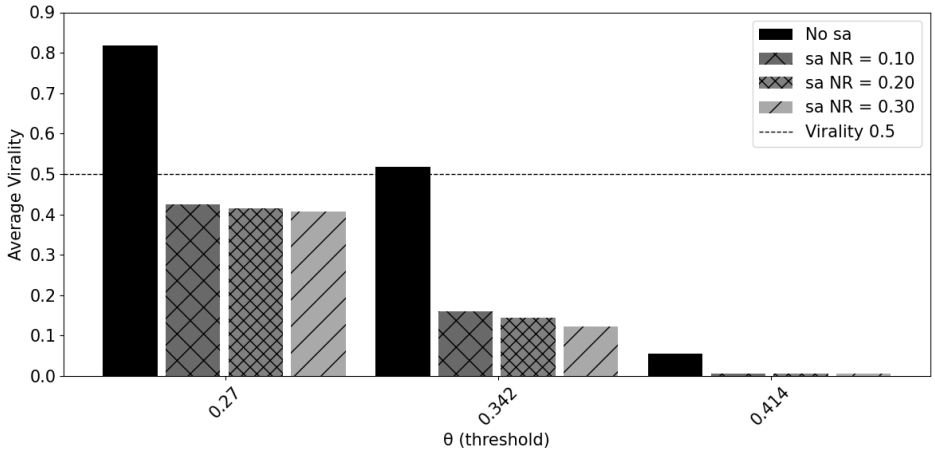


Figura 5.18: Media della Viralità raggiunta al variare di Θ (asse x) negli esperimenti senza super-agent ("No sa") e con super-agent al variare di `node-range` *(NR).*

5.2.5 Qual è l'impatto del *super-agent* sulla Viralità al variare dei parametri legati al warning?

In questa sezione si analizza, al variare dei parametri legati all'azione *Warning*, cioè `warning-impact` e `warning-impact-neutral`, come il *super-agent* riesca ad impattare la Viralità. Questi attributi permettono di determinare quanto la procedura di *Warning* riesca a influenzare sia i *basic-agents* che già hanno un'opinione, sia quelli neutri (Sez. 4.3.5).

In questo esperimento, abbiamo usato i seguenti valori: `warning-impact` $= \{0.10, 0.20\}$ e `warning-impact-neutral` $= \{0.30, 0.50\}$.

Sono stati eseguiti tre test accoppiando di volta in volta un valore per il `warning-impact` con un valore `warning-impact-neutral`. In questi test si vuole analizzare quale dei due attributi riesca ad avere un impatto maggiore sulla diffusione di notizie di tipo A, renendo queste ultime meno virali. I P_n che abbiamo preso in considerazione per questo test sono $P_n = \{0.66, 0.75, 0.83\}$, che sono i valori più interessanti che abbiamo osservato negli esperimenti senza *super-agent*.

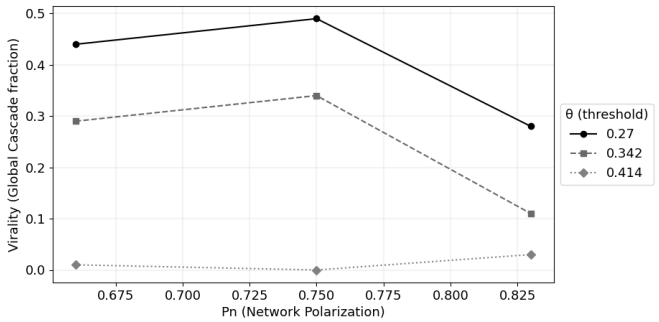
I risultati sono riportati in Fig. 5.19. In generale ciò che si può notare è che per i P_n presi in considerazione si tende ad avere una Viralità al di sotto di 0.5. In particolare, con $\Theta > 0.270$ il il `warning-impact` e il `warning-impact-neutral` hanno più impatto su V.

Con `warning-impact` $= 0.2$ e `warning-impact-neutral` $= 0.3$ (Fig. 5.19(a)) analizziamo i miglioramenti in termini di V confrontando i valori ottenuti con quelli dello stesso test senza la presenza del *super-agent* (Fig. 5.1). Per $0.66 \leq P_n \leq 0.83$ con $\Theta = 0.270$ si ha $\mu_V = 0.83$, mentre per la Fig. 5.19(a), si ha $\mu_V = 0.4$. Si è avuto un miglioramento del 51%. Con `warning-impact` $= 0.1$ e `warning-impact-neutral` $= 0.5$, si ha $\mu_V = 0.47$. Si è avuto un miglioramento del 43%. Con `warning-impact` $= 0.2$ e `warning-impact-neutral` $= 0.5$, si ha $\mu_V = 0.42$. Si è avuto un miglioramento del 49%.

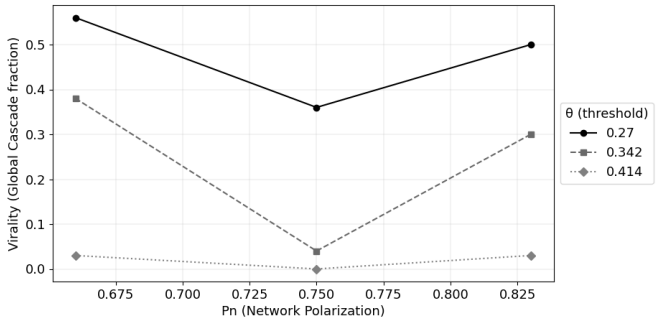
Con `warning-impact` $= 0.2$ e `warning-impact-neutral` $= 0.3$ (Fig. 5.19(a)) analizziamo i miglioramenti in termini di V confrontando i valori ottenuti con quelli dello stesso test senza la presenza del *super-agent* (Fig. 5.1). Per $0.66 \leq P_n \leq 0.83$ con $\Theta = 0.342$ si ha $\mu_V = 0.75$, mentre per la Fig. 5.19(a), si ha $\mu_V = 0.25$. Si è avuto un miglioramento del 67%. Con `warning-impact` $= 0.1$ e `warning-impact-neutral` $= 0.5$, si ha $\mu_V = 0.24$. Si è avuto un miglioramento del 68%. Con `warning-impact` $= 0.2$ e `warning-impact-neutral` $= 0.5$, si ha $\mu_V = 0.18$. Si è avuto un miglioramento del 76%.

Questi risultati sono riassunti nel grafico a barre in Fig. 5.20.

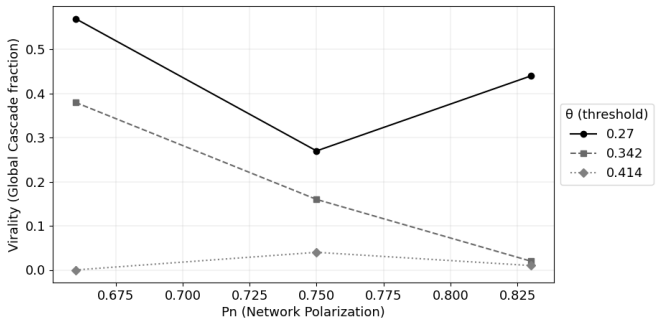
Quindi dagli esperimenti possiamo constatare che il `warning-impact` abbia un impatto maggiore su V con $\Theta = 0.270$. Mentre, con $\Theta = 0.342$ il `warning-impact` fa ridurre di più la Viralità. Questo perché con Θ bassi ($\Theta < 0.342$), i *basic-agents* tendono sempre ad avere un'opinione, quindi `warning-impact` risulta più impattante su V, mentre con Θ alti ($\Theta \geq 0.342$), i *basic-agents* tendono ad non cambiare opinione facilmente (si veda Fig. 5.10), portando così la rete ad avere più nodi neutri rendendo il `warning-impact-neutral` più impattante su V.



(a) Con `warning-impact` $= 0.2$, `warning-impact-neutral` $= 0.3$.



(b) Con `warning-impact` $= 0.1$ e `warning-impact-neutral` $= 0.5$.



(c) Con `warning-impact` $= 0.2$ e `warning-impact-neutral` $= 0.5$.

Figura 5.19: Andamento medio della Viralità delle fake news, in presenza di un super-agent, al variare di P_n e Θ. $P_n = \{0.66, 0.75, 0.83\}$, $\Theta = \{0.270, 0.342, 0.414\}$.

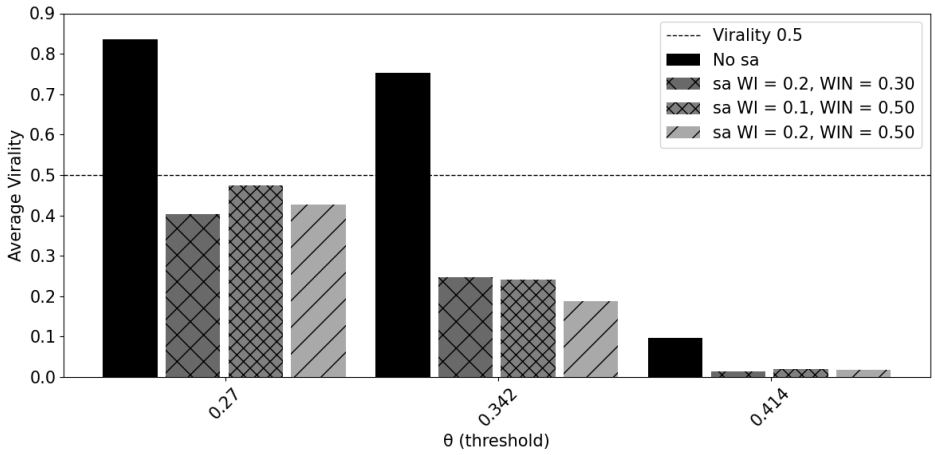


Figura 5.20: Media della Viralità raggiunta al variare di Θ (asse x) negli esperimenti senza super-agent ("No sa") e con super-agent al variare di `warning-impact` *(WI) e* `warning-impact-neutral` *(WIN).*

Capitolo 6

Conclusioni

La popolarità dei social network, negli ultimi anni ha permesso di rivoluzionare la generazione e la distribuzione di informazioni. Al contempo, la facilità nell'accesso alle informazioni è un terreno fertile per la diffusione delle fake news.

Questo flusso enorme di informazioni può generare diversi problemi. In *primis*, l'assenza di un controllo editoriale per la qualità dei contenuti che circolano sulle piattaforme Web ha determinato l'assenza di controlli sulla veridicità e qualità. Questo problema ha portato alla creazione di alcuni fenomeni online, ad esempio: le *echo chamber*, disinformazione politica e in generale, manipolazione delle opinioni del pubblico tramite le notizie false.

Diversi studi hanno studiato ed affrontato il problema della diffusione delle fake news ma sempre limitandosi ad uno tra i due approcci fondamentali, *model-driven* (simulando comportamenti sociali per comprendere il fenomeno fake news) e *data-driven* (cercando meccanismi per individuare e riconoscere le fake news).

In questo lavoro di tesi si è sperimentato un approccio alternativo per lo studio ed il contrasto alle fake news, utilizzando un duplice approccio, *model-driven* e *data-driven*. Da un lato, abbiamo implementato una simulazione ad agenti per modellare la diffusione delle fake news (con annessa creazione delle cosiddette *echo chambers*), dall'altro abbiamo utilizzato un *super-agent* basato su DRL per apprendere dai dati della simulazione a prendere decisioni atte al contenimento della diffusione delle fake news(intervenendo in tempo reale sulla simulazione).

I risultati del lavoro di tesi dimostrano, che inserendo un *super-agent* in grado di intervenire all'interno della simulazione questo può avere un impatto sulla Viralità di una fake news. Il *super-agent* può riuscire a diminuire la

Viralità della disinformazione fino al 55% rispetto alle simulazioni in cui non fosse presente.

Riteniamo che lo strumento possa essere un valido supporto per i ricercatori impegnati nella ricerca al contrasto delle fake news in quanto il sistema così realizzato rappresenta una tela per successivi esperimenti e studi che coinvolgono approcci ibridi *model-driven* e *data-driven*.

6.1 Limitazioni e sviluppi futuri

Il lavoro svolto non è esente da limitazioni che qui di seguito si tenta di riassumere classificandole in tre grandi categorie:

- *Limitazioni del modello simulativo basato ad agenti e possibili migliorie*: il modello simulativo su *NetLogo*, sebbene più ricco di altri (si veda [12]) può essere ancora raffinato. Infatti, uno dei possibili sviluppi futuri è quello di implementare i bias che governano le azioni dei *basic-agents*, rappresentati tramite reti *bayesiane* (ad esempio, come in [27]), così da poter rendere più simile alla realtà la simulazione. In particolare, in [27], sono presenti due bias che influenzano gli agenti: la probabilità che la fonte abbia notizie accurate e la probabilità che la fonte voglia condividere informazioni accurate. Queste perché le persone tendono a considerare sia la competenza che l'affidabilità della fonte delle notizie per aggiornare le proprie opinioni ([39]). Altri bias che possono essere mappati sugli agenti sono presenti in [14]. Gli agenti potrebbero avere ad esempio i seguenti bias: *Confirmation bias*, *Attentional bias*, *Selective exposure*, *Congruence Bias*, *Belief bias* e *Emotional bias*. Una possibile limitazione della nostra simulazione è dovuta al fatto che i network generati e utilizzati per i test non corrispondo a reti sociali reali. Con l'accesso, ad esempio, alle API di *Twitter*[1] si potrebbero ottenere le informazioni riguardanti una sotto-rete (a partire da un gruppo di utenti e dai loro follower) che può essere replicata all'interno di *NetLogo*. Infine, il modello sviluppato prevede soltanto network statici, cioè che hanno una struttura definita al momento del set up (tick 0) che resta la stessa sino al termine della simulazione. Sarebbe molto interessante, in questo senso, prevedere funzionalità in grado di rendere il network dinamico, con ingresso-uscita di *basic-agents*, creazione-cancellazione di collegamenti, ecc.

[1] Con il cambio di proprietà la piattaforma di micro-blogging ha interrotto il supporto per le API non Enterprise.

- *Limitazioni del modello di DRL alla base del super agent e possibili spunti per ulteriori indagini e sviluppi*: per quanto riguarda il modello DRL, ci sono molti migliorie che possono essere prese in considerazione. Innanzitutto, il sistema di *reward* del *super-agent* può essere migliorato e modificato per includere più casi ed al fine di renderlo più "su misura". Si può pensare a un sistema di *reward* basato sul concetto che le azioni abbiano un peso, e questo peso influisce sulla *reward* finale ottenibile dal *super-agent*. Ad esempio, alla procedura di *Static B Nodes* può essere associato un peso molto alto (questa procedura se applicata in scenari reali può risultare onerosa da attuare in quanto mappa imposizioni alle testate giornalistiche o enti importanti alla diffusione di notizie vere) e quindi se viene utilizzata senza ottenere dei risultati rilevanti, sia a breve che a lungo termine, le ricompense vengono ridotte. Inoltre, si può migliorare il *super-agent* andando a modificare lo spazio di osservazione dell'ambiente. Ad esempio, e questo si collega ad uno degli sviluppi futuri riguardo il modello di simulazione suddetti, si può pensare di tener traccia dell'evoluzione del network, della sua struttura. Un fattore fondamentale che può migliorare il calcolo delle *reward* è anche la creazione di uno spazio di osservazione più veritiero. In quanto, il nostro si basa su tre dimensioni ma è possibile tramite le librerie di *OpenAI Gymnasium* avere uno spazio di osservazione che rappresenta un grafo. Quindi è possibili ricreare la struttura del grafo presa da *NetLogo* su *Gymnasium*, ottenendo così dei risultati più accurati per l'addestramento del *super-agent*. Inoltre, si potrebbero aggiungere altri azioni a disposizione del *super-agent*, visto che sono presenti altri problemi che sono dovuti alla diffusione della disinformazione [38], e fare operazioni di tuning su quelle sviluppate. Altri miglioramenti sono ottenibili facendo operazioni di tuning sul modello di addestramento che usa il DRL: cambiando i *layer* da cui è composta la rete neurale, cambiando la *batch size* oppure modificando la taglia della *replay memory*, il *learning rate*, ecc.

- *Limitazioni degli esperimenti condotti e piano di lavoro*: il numero degli esperimenti diversi che è possibile eseguire, solo per il modello simulativo è pari a 10^{23}, considerando una discretizzazione dello spazio del dominio dei parametri con uno slice del 5% (es. sia $x \in \mathbb{R}[0,1]$, allora gli esperimenti da condurre includono i seguenti valori: $exp_x = \{0.00, 0.05, 0.10, \ldots, 0.95, 1.00\}$; per quanto concerne invece il modello di DRL alla base del *super-agent*, ci sono ulteriori 448 "parametri addestrabili", quindi nell'ordine di 10^3. Il tutto risulta

in $4,4 * 10^{25}$ diversi esperimenti. In questo lavoro, in via preliminare, ci siamo limitati a condurre esperimenti utilizzando solo una rete di tipo *Erdős–Rényi*, ma sarebbe interessante osservare le dinamiche del fenomeno effettuando test su reti di tipo *Small World* o *Preferencial Attachment*. Altri esperimenti potrebbero essere eseguiti tenendo conto della direzione dei collegamenti tra i nodi, simulando il rapporto dei seguaci su un social network. Inoltre, è possibile utilizzare degli strumenti, come *OpenMPI*, libreria MPI4py[2], che permettano di parallelizzare meglio le esecuzioni dei test con il *super-agent*, visto che l'uso del DRL richiede un tempo di esecuzione elevato. Questo permetterebbe di analizzare in meno tempo più esperimenti, il che non è poco visto che il numero degli esperimenti possibili è molto elevato.

[2]https://mpi4py.readthedocs.io/en/stable/

Bibliografia

[1] Anna Gausen, Wayne Luk e Ce Guo.
≪Can we stop fake news? using agent-based modelling to evaluate countermeasures for misinformation on social media≫.
In: *International AAAI Conference on Web and Social Media (ICWSM). https://doi. org/10.36190*. 2021.

[2] Giancarlo Ruffo et al. ≪Surveying the research on fake news in social media: a tale of networks and language≫.
In: *arXiv preprint arXiv:2109.07909* (2021).

[3] Marcella Tambuscio et al. ≪Network segregation in a model of misinformation and fact-checking≫.
In: *Journal of Computational Social Science* 1 (2018), pp. 261–275.

[4] Roberto Abbruzzese et al. ≪Detecting influential news in online communities: an approach based on hexagons of opposition generated by three-way decisions and probabilistic rough sets≫.
In: *Information Sciences* 578 (2021), pp. 364–377.

[5] Nicola Capuano et al. ≪Content Based Fake News Detection with machine and deep learning: a systematic review≫.
In: *Neurocomputing* (2023).

[6] Mohamed K Elhadad, Kin Fun Li e Fayez Gebali.
≪Fake news detection on social media: a systematic survey≫.
In: *2019 IEEE Pacific Rim conference on communications, computers and signal processing (PACRIM)*. IEEE. 2019, pp. 1–8.

[7] Wenlin Han e Varshil Mehta. ≪Fake news detection in social networks using machine learning and deep learning: Performance evaluation≫.
In: *2019 IEEE International Conference on Industrial Internet (ICII)*. IEEE. 2019, pp. 375–380.

[8] Rohit Kumar Kaliyar, Anurag Goswami e Pratik Narang. ≪FakeBERT: Fake news detection in social media with a BERT-based deep learning approach≫. In: *Multimedia tools and applications* 80.8 (2021), pp. 11765–11788.

[9] Syed Ishfaq Manzoor, Jimmy Singla et al. ≪Fake news detection using machine learning approaches: A systematic review≫. In: *2019 3rd international conference on trends in electronics and informatics (ICOEI)*. IEEE. 2019, pp. 230–234.

[10] Andrea Renda. *The legal framework to address" fake news": Possible policy actions at the EU level*. European Parliament, 2018.

[11] Walter Quattrociocchi, Antonio Scala e Cass R Sunstein. ≪Echo chambers on Facebook≫. In: *Available at SSRN 2795110* (2016).

[12] Petter Törnberg. ≪Echo chambers and viral misinformation: Modeling fake news as complex contagion≫. In: *PLOS ONE* 13.9 (set. 2018), pp. 1–21. DOI: 10.1371/journal.pone.0203958. URL: https://doi.org/10.1371/journal.pone.0203958.

[13] Damian Tambini. ≪Fake news: public policy responses≫. In: (2017).

[14] Giancarlo Ruffo et al. ≪Studying fake news spreading, polarisation dynamics, and manipulation by bots: A tale of networks and language≫. In: *Computer Science Review* 47 (2023), p. 100531. ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2022.100531. URL: https://www.sciencedirect.com/science/article/pii/S157401372200065X.

[15] Benjamin D. Horne e Sibel Adali. ≪This Just In: Fake News Packs a Lot in Title, Uses Simpler, Repetitive Content in Text Body, More Similar to Satire than Real News≫. In: *CoRR* abs/1703.09398 (2017). arXiv: 1703.09398. URL: http://arxiv.org/abs/1703.09398.

[16] Richard Nadeau, Edouard Cloutier e J.-H. Guay. ≪New Evidence About the Existence of a Bandwagon Effect in the Opinion Formation Process≫. In: *International Political Science Review* 14.2 (1993), pp. 203–213. DOI: 10.1177/019251219301400204. URL: https://doi.org/10.1177/019251219301400204.

[17] C. Nathan DeWall e Brad J. Bushman.
≪Social Acceptance and Rejection: The Sweet and the Bitter≫. In:
Current Directions in Psychological Science 20.4 (2011), pp. 256–260.
DOI: 10.1177/0963721411417545.
URL: https://doi.org/10.1177/0963721411417545.

[18] W. Phillips Davison. ≪The Third-Person Effect in Communication≫.
In: *Public Opinion Quarterly* 47.1 (gen. 1983), pp. 1–15.
ISSN: 0033-362X. DOI: 10.1086/268763.
eprint: https://academic.oup.com/poq/article-pdf/47/1/1/5382397/47-1-1.pdf.
URL: https://doi.org/10.1086/268763.

[19] Lee Ross e Andrew Ward. ≪Naive Realism: Implications for Social
Conflict and Misunderstanding≫. In: gen. 1996, pp. 103–135.

[20] Lee Ross, David Greene e Pamela House.
≪The "false consensus effect": An egocentric bias in social perception
and attribution processes≫. In: *Journal of Experimental Social
Psychology* 13.3 (1977), pp. 279–301. ISSN: 0022-1031.
DOI: https://doi.org/10.1016/0022-1031(77)90049-X.
URL: https://www.sciencedirect.com/science/article/pii/002210317790049X.

[21] Raymond S. Nickerson.
≪Confirmation Bias: A Ubiquitous Phenomenon in Many Guises≫.
In: *Review of General Psychology* 2.2 (1998), pp. 175–220.
DOI: 10.1037/1089-2680.2.2.175.
eprint: https://doi.org/10.1037/1089-2680.2.2.175.
URL: https://doi.org/10.1037/1089-2680.2.2.175.

[22] J Mark G Williams et al.
Cognitive psychology and emotional disorders. Vol. 2.
Wiley Chichester, 1997.

[23] Jonathan L. Freedman e David O. Sears.
≪Selective Exposure11The preparation of this paper was supported in
part by NSF grants to the authors.≫ In: a cura di Leonard Berkowitz.
Vol. 2. Advances in Experimental Social Psychology.
Academic Press, 1965, pp. 57–97.
DOI: https://doi.org/10.1016/S0065-2601(08)60103-3.
URL: https://www.sciencedirect.com/science/article/pii/S0065260108601033.

[24] P. C. Wason.
≪On the Failure to Eliminate Hypotheses in a Conceptual Task≫.
In: *Quarterly Journal of Experimental Psychology* 12.3 (1960),
pp. 129–140. DOI: 10.1080/17470216008416717.
URL: https://doi.org/10.1080/17470216008416717.

[25] Jacqueline P Leighton, Robert J Sternberg et al.
The nature of reasoning. Cambridge University Press, 2004.

[26] Lisa Feldman Barrett e Moshe Bar.
≪See it with feeling: affective predictions during object perception≫.
In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 364.1521 (2009), pp. 1325–1334.

[27] Jan-Philipp Fränken e Toby Pilditch.
≪Cascades Across Networks Are Sufficient for the Formation of Echo Chambers: An Agent-Based Model≫. In: *Journal of Artificial Societies and Social Simulation* 24.3 (2021), p. 1. ISSN: 1460-7425.
DOI: 10.18564/jasss.4566.
URL: http://jasss.soc.surrey.ac.uk/24/3/1.html.

[28] Jens Koed Madsen, Richard M Bailey e Toby D Pilditch.
≪Large networks of rational agents form persistent echo chambers≫.
In: *Scientific reports* 8.1 (2018), pp. 1–8.

[29] Walter Quattrociocchi, Guido Caldarelli e Antonio Scala.
≪Opinion dynamics on interacting networks: media competition and social influence≫. In: *Scientific reports* 4.1 (2014), pp. 1–7.

[30] Xinyi Zhou e Reza Zafarani. ≪A Survey of Fake News: Fundamental Theories, Detection Methods, and Opportunities≫.
In: *ACM Comput. Surv.* 53.5 (set. 2020). ISSN: 0360-0300.
DOI: 10.1145/3395046. URL: https://doi.org/10.1145/3395046.

[31] Tanveer Khan, Antonis Michalas e Adnan Akhunzada.
≪Fake news outbreak 2021: Can we stop the viral spread?≫ In:
Journal of Network and Computer Applications 190 (2021), p. 103112.
ISSN: 1084-8045.
DOI: https://doi.org/10.1016/j.jnca.2021.103112.
URL: https://www.sciencedirect.com/science/article/pii/S1084804521001326.

[32] Yimin Chen, Niall J Conroy e Victoria L Rubin.
≪Misleading online content: recognizing clickbait as" false news"≫.
In: *Proceedings of the 2015 ACM on workshop on multimodal deception detection*. 2015, pp. 15–19.

[33] Victoria L. Rubin et al. ≪A News Verification Browser for the Detection of Clickbait, Satire, and Falsified News≫.
In: *Journal of Open Source Software* 4.35 (2019), p. 1208.
DOI: 10.21105/joss.01208.
URL: https://doi.org/10.21105/joss.01208.

[34] Nigel Gilbert.
≪Agent-based social simulation: dealing with complexity≫.
In: *The Complex Systems Network of Excellence* 9.25 (2004), pp. 1–14.

[35] Volodymyr Mnih et al.
Playing Atari with Deep Reinforcement Learning. 2013.
arXiv: 1312.5602 [cs.LG].

[36] Marc Jaxa-Rozen e Jan H. Kwakkel.
≪PyNetLogo: Linking NetLogo with Python≫. In: *Journal of Artificial Societies and Social Simulation* 21.2 (2018), p. 4. ISSN: 1460-7425.
DOI: 10.18564/jasss.3668.
URL: http://jasss.soc.surrey.ac.uk/21/2/4.html.

[37] V. Grimm et al. ≪The ODD protocol for describing agent-based and other simulation models: A second update to improve clarity, replication, and structural realism≫. In: *Journal of Artificial Societies and Social Simulation* 23.2 (gen. 2020).
URL: http://eprints.bournemouth.ac.uk/33918/.

[38] Stephan Lewandowsky et al. ≪Misinformation and its correction: Continued influence and successful debiasing≫. In: *Psychological science in the public interest* 13.3 (2012), pp. 106–131.

[39] Daniel Hawthorne-Madell e Noah D Goodman.
≪Reasoning about social sources to learn from actions and outcomes.≫
In: *Decision* 6.1 (2019), p. 17.

[40] William Rand et al.
≪An Agent-Based Model of Urgent Diffusion in Social Media≫. In: *Journal of Artificial Societies and Social Simulation* 18.2 (2015), p. 1.
ISSN: 1460-7425. DOI: 10.18564/jasss.2616.
URL: http://jasss.soc.surrey.ac.uk/18/2/1.html.

[41] Soroush Vosoughi, Deb Roy e Sinan Aral.
≪The spread of true and false news online≫.
In: *Science* 359.6380 (2018), pp. 1146–1151.
DOI: 10.1126/science.aap9559. URL:
https://www.science.org/doi/abs/10.1126/science.aap9559.

[42] Antonio F. Peralta, János Kertész e Gerardo Iñiguez.
Opinion dynamics in social networks: From models to data. 2022.
arXiv: 2201.01322 [physics.soc-ph].

[43] Claudio Castellano, Santo Fortunato e Vittorio Loreto.
≪Statistical physics of social dynamics≫.
In: *Rev. Mod. Phys.* 81 (2 mag. 2009), pp. 591–646.
DOI: 10.1103/RevModPhys.81.591.
URL: https://link.aps.org/doi/10.1103/RevModPhys.81.591.

[44] Michele Carillo et al.
≪Sociality, Sanctions, Damaging Behaviors: A Distributed
Implementation of an Agent-Based Social Simulation Model≫.
In: *Euro-Par 2013: Parallel Processing Workshops*.
A cura di Dieter an Mey et al.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 595–604.
ISBN: 978-3-642-54420-0.

[45] Flaminio Squazzoni et al. ≪Computational models that matter during a global pandemic outbreak: A call to action≫. In: (2020).

[46] Volker Grimm et al. ≪The ODD Protocol for Describing Agent-Based and Other Simulation Models: A Second Update to Improve Clarity, Replication, and Structural Realism≫. In: *Journal of Artificial Societies and Social Simulation* 23.2 (2020), p. 7. ISSN: 1460-7425.
DOI: 10.18564/jasss.4259.
URL: http://jasss.soc.surrey.ac.uk/23/2/7.html.

[47] Sandersan Onie et al. ≪Investigating the Effects of Inhibition Training on Attentional Bias Change: A Simple Bayesian Approach≫.
In: *Frontiers in Psychology* 9 (2019). ISSN: 1664-1078.
DOI: 10.3389/fpsyg.2018.02782. URL: https:
//www.frontiersin.org/articles/10.3389/fpsyg.2018.02782.

[48] R. Kelly Garrett. ≪Echo chambers online?: Politically motivated selective exposure among Internet news users1≫. In: *Journal of Computer-Mediated Communication* 14.2 (gen. 2009), pp. 265–285.
ISSN: 1083-6101. DOI: 10.1111/j.1083-6101.2009.01440.x.
eprint: https://academic.oup.com/jcmc/article-

pdf/14/2/265/21491614/jjcmcom0265.pdf.
URL: https://doi.org/10.1111/j.1083-6101.2009.01440.x.

[49] Andrei Boutyline e Robb Willer.
≪The Social Structure of Political Echo Chambers: Variation in Ideological Homophily in Online Networks≫.
In: *Political Psychology* 38.3 (2017), pp. 551–569.
DOI: https://doi.org/10.1111/pops.12337. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/pops.12337.

[50] David O. Sears e Jonathan L. Freedman.
≪Selective Exposure to information: a critical review*≫.
In: *Public Opinion Quarterly* 31.2 (gen. 1967), pp. 194–213.
ISSN: 0033-362X. DOI: 10.1086/267513.
eprint: https://academic.oup.com/poq/article-pdf/31/2/194/5132577/31-2-194.pdf.
URL: https://doi.org/10.1086/267513.

[51] Damon Centola.
≪The Spread of Behavior in an Online Social Network Experiment≫.
In: *Science* 329.5996 (2010), pp. 1194–1197.
DOI: 10.1126/science.1185231. URL: https://www.science.org/doi/abs/10.1126/science.1185231.

[52] Michela Del Vicario et al.
≪Modeling confirmation bias and polarization≫.
In: *Scientific reports* 7.1 (2017), p. 40391.

[53] Yuxi Li. *Deep Reinforcement Learning: An Overview*. 2018.
arXiv: 1701.07274 [cs.LG].

[54] Greg Brockman et al. *OpenAI Gym*. 2016.
arXiv: 1606.01540 [cs.LG].

[55] Seth Tisue e Uri Wilensky.
≪Netlogo: A simple environment for modeling complexity≫.
In: *International conference on complex systems*. Vol. 21.
Citeseer. 2004, pp. 16–21.

Appendice

A. Caratteristiche dell'ambiente di sviluppo

In questo lavoro di tesi è stata utilizzata una macchina equipaggiata come riportato in Tabella 6.1.

Componente	**Caratteristiche**
Processore	`12th Gen Intel(R) Core(TM) i7-12700KF 3.61 GHz`
RAM	`32,0 GB (31,8 GB utilizzabile)`
Tipo di sistema	SO a 64 bit, processore basato su `x64`
Scheda Video	`NVIDIA GeForce RTX 3080 Ti`
Nome SO	`Microsoft Windows 11 Home`
Modello Sistema	`AlienWare Aurora R13`
Archiviazione	`NVMe Micron 3400 NVMe 2048GB`

Tabella 6.1: Configurazione della macchina utilizzata per gli esperimenti

Viene anche mostrata una simulazione su *NetLogo* durante un'esecuzione in Fig. 6.1.

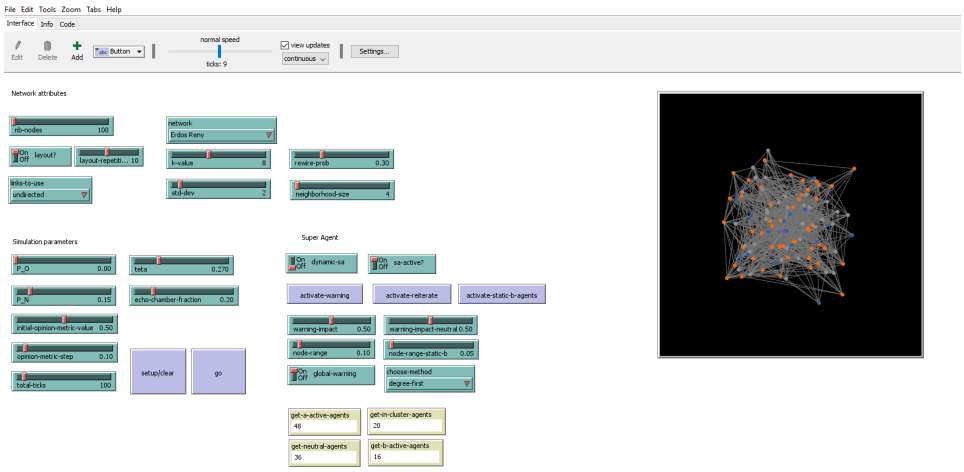


Figura 6.1: Simulazione su NetLogo della diffusione delle fake news all'interno di una rete