Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE

Corso di Laurea in Informatica

Text Watermarking e Social Network: uno studio sperimentale

Relatore:
Chiar.mo Prof.
Danilo Montesi

Presentata da:
Carlo Stomeo

Correlatori:
Stefano Giovanni Rizzo
Flavio Bertini

Sessione II
Anno Accademico 2015-2016

A chi ha creduto in me quando
anche io avevo smesso di farlo
e a te che pur non essendoci più
mi hai dato la forza per
andare avanti...

Indice

Elenco delle figure

Elenco delle tabelle

Introduzione

I Social Network stanno diventando una figura sempre più presente nella vita quotidiana delle persone; nel tempo sono diventati sempre più diffusi, contano ormai milioni di utenti e sempre più ricchi di funzionalità.
Sono utilizzati per comunicare, per condividere contenuti multimediali di ogni tipo, ed anche per raccontare se stessi.
Questi motivi rendono di particolare interesse metodi che possano dare qualche garanzia in più, come copyright e autenticazione, ai file che condividiamo con il mondo attraverso queste piattaforme; ovvero i metodi di watermarking.
Il watermarking consiste nell'inserimento di qualcosa di unico nel file, che sia un'immagine, un video, un file audio o anche un file di testo, al fine di garantirne l'autenticità e l'integrità, per avere un'idea elementare di questo metodo lo si può immaginare come una sorta di impronta digitale.
Grazie a questa tecnica è possibile ottenere principalmente due grandi garanzie: nel caso in cui un file digitale di nostra proprietà venga ripubblicato da terzi senza il nostro consenso sarà possibile dimostrarne la vera proprietà estraendo appunto il watermark da noi inserito; nel caso invece si voglia distribuire un file digitale, è possibile utilizzare il watermark per evitare che venga ridistribuito illegalmente, inserendo un watermark unico per ogni copia distribuita, in questo modo se il file dovesse essere copiato e ridistribuito sarebbe possibile risalire al distributore non autorizzato, ovvero colui a cui è legato il particolare watermark estratto dalla copia illegale.
Questa tecnica in realtà non si limita solamente alle due applicazioni appena citate, per questo motivo verrà descritta più dettagliatamente nel capitolo

successivo.
In questo scritto illustrerò quindi un metodo di watermarking su testo, e ne studierò l'applicazione sui Social Network.
In questa prima fase spiegherò in che cosa consiste il nostro studio e le sue motivazioni, successivamente verrà introdotto il watermarking ed in particolare il watermarking testuale. Nel terzo capitolo spiegherò l'algoritmo che è stato utilizzato per svolgere gli esperimenti ed infine nei capitoli successivi saranno spiegati nel dettaglio gli esperimenti stessi ed i loro risultati.

Obiettivi

Gli obiettivi principali di questo studio sono stati due:

- capire se i social network inserissero watermarking nei testi su di essi caricati;
- testare l'algoritmo di text watermarking proposto, basato sulla sostituzione degli omoglifi, nell'ambiente dei social network.

Per fare questo sono quindi stati predisposti diversi esperimenti.
In una prima fase una serie di testi sono stati pubblicati ed in seguito analizzati per capire come i social network si comportassero con determinate stringhe.
Se queste avessero effettuato consistenti modifiche al testo sarebbe stata concreta la possibilità dell'utilizzo di tecniche di waterking da parte delle stesse.
Successivamente si è passati ad analizzare i limiti che le piattaforme impongono all'algoritmo da noi utilizzato per inserire il watermark, questo infatti si basa sulla sostituzione degli omoglifi, e le piattaforme potrebbero non accettarne alcuni.
Infine, considerando questi limiti si è cercato di misurare la qualità del metodo utilizzato, valutandone percentuali di successo ed invisibilità.
Infatti non è sempre garantito il successo dell'algoritmo, per via delle limitazioni appena citate e soprattutto non è detto che una stringa pubblicata

con successo su una piattaforma possa essere ripubblicata, senza perdere il watermark, su di un'altra.
Per questo motivo il testing non si è limitato alle sole piattaforme prese singolarmente, ma è stata posta attenzione anche alla possibilità di veicolare messaggi con relativo watermark attraverso piattaforme diverse.

Motivazioni

Il successo del nostro metodo darebbe la possibilità di autenticare file testuali all'interno delle piattaforme, in una realtà in cui verificare la fonte di un determinato post è sempre più difficile; oltre che la possibilità di inserire messaggi invisibili all'interno delle stesse pubblicazioni.
Il watermarking infatti permetterebbe di porre una firma unica sui file testuali, trasmessi sui social ad un ritmo sempre maggiore, rendendo così possibile l'identificazione del vero proprietario per motivi di copyright.
Inoltre il watermark potrebbe anche fungere da messaggio steganografico, ovvero nascosto, fornendo così un canale di comunicazione "out of band".
Questo avrebbe utili applicazioni nell'ambiente della crittografia, ad esempio per la trasmissione delle chiavi private, per il quale è appunto necessario trovare un metodo di trasmissione che non possa essere intercettato da terzi in alcun modo.

Per autenticare un contenuto in modo efficace però il watermark deve essere robusto, ed in particolare resistere al passaggio da una piattaforma all'altra, è infatti frequente la condivisione di un contenuto attraverso diversi social; ragion per cui sono stati svolti gli studi cross-social, ovvero atti a capire in che percentuale i post pubblicati su un Social possano essere ripubblicati su di un altro senza danneggiare il watermark in essi contenuto.

Capitolo 1

Il Watermarking

1.1 Cos'è il watermarking

Il termine watermarking significa filigrana, ed infatti assume la stessa funzione in termini digitali; consiste cioè nell'inclusione del watermark all'interno di un file digitale, al fine di garantirne provenienza e autenticità.
Similmente a quanto accade per le banconote non sarà sufficiente la presenza della filigrana, watermark, per essere sicuri dell'autenticità di un file, ma sarà necessaria una fase di estrazione e verifica dello stesso; chiunque infatti può inserire un watermark all'interno del file, ma solo uno sarà quello autentico.
Si ricordi che l'autenticazione dei file è solo uno dei possibili utilizzi del watermark, che quindi assumerà diverse proprietà, esposte in Sezione 1.2, a seconda dell'utilizzo.
Lo sviluppo dei social , e del mondo digitale in generale ha portato alla condivisione di un'enorme quantità di file di ogni tipo, per cui risulta evidente la necessità di protezione e identificazione dei file stessi.
A tal fine esistono diversi metodi, divisibili in tre gruppi [14]:

- crittografia, volta a rendere un contenuto illeggibile se non per il possessore di una chiave specifica;
- steganografia, volta a nascondere nuove informazioni all'interno del file originale;

- watermarking, allo scopo di garantire l'autenticità del file , inserendone appunto all'interno qualcosa di unico.

Quest'ultima tecnica risulta quindi la migliore per garantire l'integrità e l'origine di un file digitale e non solo [22].
In particolare per garantire l'authorship di un file sarà sufficiente estrarre il watermark, che in questo caso dovrà essere robusto, ovvero resistere a tentativi di alterazione, e verificare che sia quello che identifica un determinato autore.
Per garantire l'integrità di un file invece è sufficiente inserirne un watermark alla creazione, e far si che anch'esso venga modificato se dovesse esserlo il file, questo tipo di watermark viene detto fragile. Così facendo solo un file integro avrà preservato il watermark così com'è stato inserito.
Il watermarking può anche essere utilizzato per prevenire la distribuzione non autorizzata dei dati, in particolare inserendo un watermak univoco e robusto per ogni utente. In questo modo se il file dovesse venire copiato e ridistribuito, estraendone il marchio si potrebbe risalire al distributore.
Infine questa tecnica può essere utilizzata per inserire informazioni extra all'interno dei contenuti digitali, come data, posizione, autore, o anche restrizioni, come il periodo di validità del file stesso.

1.2 Classificazione e Proprietà

1.2.1 Classificazione

Come spiegato nell'articolo [15] il watermarking può essere classificato nel modo seguente:

- *Visibile o invisibile* : il watermark è invisibile se nascosto nel documento, per cui non appare all'utente; altrimenti il watermark sarà visibile e quindi riconoscibile da un utente.

- *Leggibile o rilevabile*: il watermark si dice leggibile se è appunto facilmente leggibile da un utente, altrimenti si dice rilevabile e ci sarà bisogno di una funzione apposita per rilevarne la presenza.

- *Robusto, fragile o semi-fragile* : il watermark robusto è rilevabile, ma non cancellabile, questo viene usato per garantire il copyright del documento. Un watermark fragile è rilevabile, ma può essere modificato o addirittura cancellato, per questo motivo viene utilizzato per controlli di integrità. Infine il watermark semi-fragile è utilizzato per fare autenticazione dei file.

- *Cieco , non cieco o zero watermarking*: un watermark si dice cieco se nella fase di estrazione non c'è bisogno del file originale, altrimenti si dice non cieco. Infine la classe zero watermarking identifica gli algoritmi in cui il documento non viene modificato, ma si usano alcune caratteristiche dello stesso, per generare il watermark.

- *Semplice o multiplo*: un watermark semplice è applicato una sola volta, mentre uno multiplo viene applicato diverse volte, facendo si che che le nuove applicazioni non comportino modifiche alle precedenti.

1.2.2 Proprietà

Nell'articolo [23] sono spiegate le proprietà fondamentali del watermark:

- *Robustezza* - È la capacità del watermark di rimanere inalterato dopo diverse operazioni , o attacchi; questo infatti non deve essere rimosso o modificato, altrimenti perderebbe la sua utilità.
- *Fedeltà o Impercettibilità* - È una proprietà fondamentale per il watermark e consiste nell'impossibilità di essere riconosciuto, se non attraverso processi specifici, ed in particolare solo dalle persone autorizzate.
- *Sicurezza* - Consiste nell'impossibilità di rimuovere o modificare il watermark da parte di una persona non autorizzata, se non in possesso della piena conoscenza del metodo o del watermark stesso.
- *Data-payload* - Consiste nel massimo numero di bit inseribile nel documento originale.
- *Complessità computazionale* - È una misura del tempo richiesto per inserire ed estrarre il watermark.
- *Inevitabilità* - È definita come la probabilità di ottenere i dati originali durante l'estrazione del watermark.

1.3 Watermarking su testo

Il watermarking può essere fatto su file digitali di diverso tipo, come immagini, video, file audio e file di testo; il nostro studio però si è concentrato solo su quest'ultimo, per cui spiegherò in dettaglio solamente i diversi approcci volti al watermarking testuale.

- *Tecniche basate su immagini* - Dalla semplicità dei dati di tipo testuale scaturisce una certa difficoltà nell'inserimento del watermark, sono infatti molte meno le caratteristiche del documento che si possono sfruttare rispetto a file di natura più complessa, per cui un primo approccio

è stato quello di convertire il testo in immagine, e innestare in essa il watermark. Questo tipo di watermark però ha il grande difetto di modificare la natura del documento.

- *Tecniche sintattiche* - Con sintassi indichiamo l'insieme di regole che vanno a formare una frase, questo metodo quindi va a modificare la struttura delle frasi per inserire il watermark. Tuttavia, pur preservando la natura del file, in questo caso ne viene modificato il contenuto.

- *Tecniche semantiche* - Con semantica si intende il significato del testo, questo approccio utilizzerà quindi il contenuto del testo per inserire il watermark, in particolare vengono sfruttati nomi, verbi, preposizioni , lo spelling delle parole etc. Similmente al caso precedente però viene appunto alterato il contenuto.

- *Metodi strutturali* - Questo è l'approccio più moderno al watermarking e consiste nello sfruttamento di doppie occorrenze di alcune lettere, nello shift di linee o parole ed anche degli standard unicode. Il vantaggio di questo metodo è che non modifica il testo, ne la natura del documento.

1.4 Stato dell'arte

Questa tesi si basa sull'uso di un watermarking di tipo strutturale, con utilizzo di cifratura, ma come precedentemente accennato in letteratura si trovano molte altre tecniche per questo scopo, che si dividono essenzialmente in 4 gruppi: basati su immagini, sintattici, semantici e strutturali.

1.4.1 Tecniche basate su immagini

Questo tipo di watermarking è il più diffuso e si basa sulla conversione del testo in un'immagine, in modo da ridurre il problema a quello dell'image-watermarking.

Una volta ottenuta l'immagine ci sono diversi approcci per inserire il watermarking; tra cui possiamo individuare due gruppi principali, ovvero i metodi che lavorano su blocchi di pixel, come caratteri, parole o intere linee e quelli che lavorano sui singoli pixel.

Tecniche che operano su blocchi di pixel: tra questi possiamo individuare la tecnica di shifting orizzontale o verticale di parti del testo in accordo ai bit di watermark da inserire; in particolare orizzontale per le parole, mentre verticale per le righe di testo [11, 18].
Un'altra tecnica consiste nella leggera alterazione di alcune parti dei caratteri, come ad esempio modificare il tratto per renderlo più spesso o più sottile [7].
Infine, alternativamente allo shift di gruppi di caratteri è possibile modificare lo spazio all'interno delle parole per inserire il watermark [13, 16].

Tecniche che operano su singoli pixel: tipiche dei documenti in scala di grigi queste tecniche vengono applicate modificando la luminosità di alcuni pixel [10], oppure attraverso istogrammi relativi alla direzione dei bordi [17]. Tuttavia questo tipo di cambiamenti sui pixel risulta più visibile, soprattutto a causa dello sfondo bianco.

Le tecniche basate su immagini però hanno il grande difetto di cambiare il tipo del documento; sui social ad esempio sarebbe estremamente anti intuitivo e poco pratico dover convertire ogni post in un'immagine.

1.4.2 Tecniche sintattiche

Questi metodi si basano sull'operare modifiche alla struttura sintattica della frase, cercando di mantenere inalterato il significato complessivo del testo e la sua natura, hanno però il grande limite di modificare fortemente la struttura del testo, ragion per cui non possono essere sempre applicati, basti

pensare ad una canzone o ad una poesia.
Nonostante ciò sono stati proposti diversi metodi; questi dopo aver costruito un albero sintattico relativo alla frase presa sotto esame, modificano la struttura di quest'ultima per inserire i bit di watermark, in particolare operando alcune operazioni come la dislocazione, lo spostamento o l'aggiunta di alcuni avverbi e preposizioni e la trasformazione al passivo dei verbi transitivi [8].
In alternativa è possibile applicare dei tool sintattici all'albero [19].
Questi metodi hanno anche un altro limite, ovvero richiedono un testo abbastanza lungo per nascondere il watermark, ragion per cui non sarebbero sempre utilizzabili sui social network, che in alcuni casi impongono limiti sulla lunghezza dei post, come Twitter o Instagram.

1.4.3 Tecniche semantiche

In questo caso è la struttura semantica della frase ad essere modificata per inserire il watermark.
Sono stati proposti diversi metodi a tal fine, tra cui la sostituzione dei sinonimi [26]; un metodo basato sull'utilizzo della presupposizione [27]; l'uso di errori di battitura, forme colloquiali o tipiche del linguaggio del web ed altri costrutti simili [25]; fino all'utilizzo di un albero TMR (Text Meaning Representation), questo metodo consiste nell'associazione di ogni parola del testo con un concetto ontologico, al fine di poter procedere con diverse operazioni, come l'inserimento e la rimozione di alcuni pronomi (potatura ed innesto dell'albero) e l'aggiunta o sostituzione di alcuni fatti relativi ad alcuni elementi citati nel testo [21].
Anche in questo caso la natura del documento viene preservata, tuttavia le modifiche possono generare una differenza significativa tra il documento originale e quello a cui è stato applicato il watermark.

1.4.4 Tecniche strutturali

I metodi strutturali si basano sull'utilizzo dei caratteri appartenenti allo standard Unicode. Di particolare interesse sono i caratteri invisibili appartenenti a questo standard, che vengono utilizzati per nascondere il watermark; in particolare le diverse codifiche per i whitespace hanno permesso di inserire il watermark nei documenti di Microsoft Word [20], ed insieme ad alcuni caratteri completamente invisibili, caratterizzati anche dall'assenza di occupazione di spazio , sono stati utilizzati per inserire watermark in file HTML [12, 1].
Nemmeno questi metodi però sono privi di difetti, infatti alterano la lunghezza del documento, poiché inseriscono nuovi caratteri; tuttavia hanno il pregio di preservare la sua natura, ed anche il suo contenuto, a differenza dei metodi precedenti.

Capitolo 2

L'algoritmo proposto

In questo capitolo spiegherò il funzionamento dell'algoritmo utilizzato per inserire il watermark sui social network; si tratta di un metodo di text watermarking basato sulla sostituzione degli omoglifi proposto da S. Rizzo, F. Bertini e D. Montesi in [23].

Tra i caratteri Unicode ne appaiono alcuni totalmente o parzialmente indistinguibili tra loro, se non per la codifica numerica sottostante, che per questo motivo vengono detti confusable. Ciò che li rende indistinguibili è il glifo con cui vengono rappresentati ragion per cui vengono definiti anche omoglifi.
Solitamente questo fenomeno viene sfruttato per truffe, e attacchi di tipo Phishing, in cui, attraverso l'utilizzo degli omoglifi vengono ricostruiti siti, interfacce o mail apparentemente identici agli originali, solitamente di banche, social network e qualunque altra cosa di potenziale interesse, in cui si chiedono dati personali all'utente, che ingannato dall'apparenza crede di fornirli all'ente vero, e non al malintenzionato del caso.
Per poter evitare questo tipo di attacchi la lista di caratteri confusable viene resa pubblica e periodicamente aggiornata dall'Unicode Consortium, al fine di poter verificare quando due stringhe sono visivamente indistinguibili.

L'algoritmo sfrutta proprio questi simboli per inserire il watermark all'interno del testo; in particolare non viene utilizzata l'intera lista di caratteri confusable, ma solo quelli che sono stati ritenuti più simili.

Il watermark da inserire nel testo non sarà costituito da un plaintext, bensì dall' hash di 64 bit risultante dall'applicazione di SipHash a quest'ultimo.
SipHash è una funzione hash crittografica di tipo MAC (Message Authentication Code) che prende in input un testo t di dimensione arbitraria ed una chiave segreta di 128-bit , e produce in output un hash di 64 bit(per maggiori dettagli [9]). Grazie all'utilizzo della chiave segreta sarà possibile verificare chi è il vero proprietario del testo, nel dettaglio, chiunque abbia accesso alle tabelle di conversione può estrarre il watermark, ma solo chi ha la chiave privata giusta può dimostrare che il watermark contenuto nel testo è lo stesso di quello generato applicando SipHash al testo originale ed utilizzando la sua chiave.

L'algoritmo lavora nel modo seguente: scandisce il testo originale, alla ricerca di un carattere confusable, una volta trovato, se questo appartiene alla Tabella 2.1 potrà inserire un bit di watermark, utilizzando la codifica originale del carattere per inserire il bit 0, mentre l'alternativa per il bit 1; se invece il carattere è un whitespace potrà codificare direttamente 3 bit di watermark (vedi Tabella 2.2).
In questo modo, non solo viene preservata la natura nel testo, ma non se ne modifica nemmeno il contenuto o la lunghezza.

La fase di estrazione del watermark avviene in modo analogo: il testo viene scandito ed ogni volta che si trova un carattere appartenente ad una delle due tabelle, in base alla codifica utilizzata viene ricostruito il bit giusto, o la giusta sequenza di tre bit, di watermark.

Possiamo quindi classificare l'algoritmo come invisibile, totalmente o parzialmente a seconda del font utilizzato; rilevabile, scandendo appunto la co-

difica dei caratteri; cieco, poiché non è necessario il testo originale ; fragile e verificabile.
La fragilità è data dalla non robustezza del watermark ad alcuni tipi di attacchi, come ad esempio la semplice riscrittura del testo; l'algoritmo risulterebbe però resistente ad un'operazione di tipo copia ed incolla, ben più comune nel contesto in cui lo stiamo utilizzando.

	$Bit\ 0$	$Bit\ 1$
$Simbolo$	$Codifica\ originale$	$Codifica\ alternativa$
,	$0x002c$	$0xa4f9$
–	$0x002d$	$0x2010$
.	$0x002e$	$0xa4f8$
;	$0x003b$	$0x037e$
C	$0x0043$	$0x216d$
D	$0x0044$	$0x216e$
K	$0x004b$	$0x212a$
L	$0x004c$	$0x216c$
M	$0x004d$	$0x216f$
V	$0x0056$	$0x2164$
X	$0x0058$	$0x2169$
c	$0x0063$	$0x217d$
d	$0x0064$	$0x217e$
i	$0x0069$	$0x2170$
j	$0x006a$	$0x0458$
l	$0x006c$	$0x217c$
v	$0x0076$	$0x2174$
x	$0x0078$	$0x2179$

Tabella 2.1: Subset simboli confusable utilizzati dall'algoritmo rappresentati in codifica hex

White space	*Bit*	*Unicode*
Space	000	$0x0020$
En Quad	001	$0x2000$
$Three - per - em\ Space$	010	$0x2004$
$Four - per - em\ Space$	011	$0x2005$
Punctation Space	100	$0x2008$
Thin Space	101	$0x202f$
$Narrow\ No - break\ Space$	110	$0x202f$
Medium Mathematical Space	111	$0x205f$

Tabella 2.2: Codifiche hex per i white space; al fine di scrivere 3 bit di watermark per ogni spazio vengono utilizzate 8 codifiche diverse

Capitolo 3

Descrizione dei test effettuati

La fase sperimentale è stata il cuore di questo studio, poiché è stata quella che effettivamente ci ha permesso di valutare le potenzialità dell'algoritmo utilizzato, nell'ambito dei social network; in particolare gli esperimenti sono stati svolti sulle seguenti 18 piattaforme: Facebook, Linkedin, Instagram, Twitter, Gmail, Google+, YouTube, Pinterest, Wordpress, Tumblr, Reddit, Vk, Flickr, Telegram, Whatsapp, Wechat, Viber e QQ.
A tal fine sono stati predisposti dei test per rispondere alle seguenti domande:

1. I social network fanno watermarking dei messaggi testuali?
2. Il metodo di watermarking con sostituzione degli omoglifi è robusto ai filtri dei social network?
3. Lo stesso è invisibile nei font e nell'encoding utilizzati dai Social Network?
4. È possibile veicolare messaggi tra piattaforme diverse preservando il watermark?

Per facilitare lo svolgimento dei test sono stati predisposti una serie di script che permettessero di postare e conseguentemente scaricare post attraverso un social, in modo da avere a disposizione il testo prima e dopo il passaggio dalla piattaforma.

Analizzando le due versioni della stringa è stato così possibile rispondere alle domande di cui sopra.

3.1 Esperimento 1: Verificare se social network fanno watermarking dei messaggi testuali

Per rispondere a questa domanda è stato necessario verificare se e in quale misura postando un testo su una delle piattaforme prese sotto esame, questo ne risultasse modificato, per esempio attraverso l'uso di una codifica diversa dei caratteri.
Tale evenienza, infatti, avrebbe reso impossibile l'utilizzo dell'algoritmo proposto, in quanto anch'esso si basa sulla sostituzione di alcuni omoglifi, e quindi le modifiche da esso attuate sarebbero potute venir sovrascritte, o comunque sommate a quelle effettuate dal social network, portando così ad errori nella fase di estrazione del watermark.
Il test è stato svolto sfruttando gli script citati in precedenza, per poter studiare in modo pratico stringhe di diversa lunghezza e natura, in modo da verificare anche limiti e politiche di gestione dei whitespace imposti da ciascuna piattaforma.
Le stringhe di ciascuna coppia, originale e scaricata, sono poi state processate con Sha1 e successivamente confrontate, se il risultato del confronto fosse stato falso allora almeno un carattere della stringa sarebbe stato modificato, altrimenti questa sarebbe passata intatta attraverso la piattaforma.
SHA-1 , Secure Hash Algorithm 1, infatti è una funzione di hashing one-way che produce un digest di 160 bit. Viene detta sicura appunto perché la ricerca di un messaggio che risulti in un determinato digest non è computazionalmente affrontabile, così come la ricerca di due messaggi diversi che risultino nello stesso digest, ragion per cui, pressoché ogni modifica al messaggio originale risulterebbe in un digest differente, come spiegato in [12, 1].

In particolare sono state utilizzate tre stringhe di base, che chiameremo short di 77 caratteri , medium di 352 caratteri e long di 718; e quando necessario ne sono state analizzate alcune aggiuntive, di lunghezze diverse, per verificare i limiti imposti dalle piattaforme.
La scelta di queste tre stringhe è stata dettata proprio dai limiti di lunghezza del testo imposti dalle piattaforme, ad esempio Twitter non accetta stringhe più lunghe di 140 caratteri, e Pinterest di 500. In aggiunta sono state analizzate alcune stringhe contenenti casi particolari di whitespace (quindi sia normali spazi, che tabulazioni), elencati di seguito, in modo da verificare se questi venissero o meno accettati senza alterazioni.

- tabulazioni;
- spazi multipli;
- tabulazioni multiple;
- newline all'interno del testo;
- spazi agli estremi del testo;
- tabulazioni agli estremi del testo.

Le modifiche sui whitespace non comporterebbero necessariamente la presenza di watermarking, in particolare perché quello che ci stiamo chiedendo noi è se questi vengano accettati così come sono o rifiutati e sostituiti con una spaziatura standard; tuttavia studiarle ci permette di dedurre importanti informazioni sulla struttura delle stringhe che possono essere utilizzate su ognuno dei social.

3.2 Esperimento 2: Testare la robustezza dell'algoritmo proposto ai filtri dei social network

Per rispondere a questa domanda è stato invece necessario capire quale fosse l'insieme di caratteri utilizzabile su ogni Social, è infatti possibile che questi non accettino tutte le codifiche per un determinato carattere, così, similmente all'esperimento precedente sono state postate stringhe contenenti la codifica unicode di tutti i 18 caratteri confusable e dei diversi whitespace utilizzati dall'algoritmo proposto, per verificare quali fossero accettati e quali invece venissero modificati o rifiutati dalle varie piattaforme. In questo modo si è definito un subset di caratteri utilizzabile per ogni social.

3.3 Esperimento 3: Testare l'invisibilità del metodo nei font e nell'encoding utilizzati dai Social Network?

A questo punto è stato necessario testare effettivamente l'algoritmo di text watermarking sulle piattaforme. In particolare ci siamo posti davanti a due nuove domande:

3.3.1 Prendendo dei post pubblicati da diversi utenti , a quanti di questi possiamo applicare l'algoritmo con successo?

Infatti per poter inserire i 64 bit di watermark sarà necessario un numero sufficiente di omoglifi sostituibili secondo il nostro metodo, e non ne sarà quindi sempre garantito il successo.

3.3.2 In che misura un post "originale" risulta diverso da uno a cui è stato applicato il watermark?

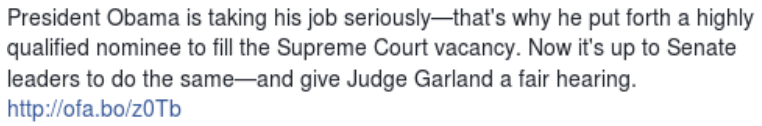
Modificando gli omoglifi dei caratteri, questi potranno risultare lievemente differenti, inoltre modificando i whitespace anche le spaziatu-

re varieranno leggermente, ragion per cui è necessario studiare questo cambiamento, e capire se risulta visibile ad un utente.

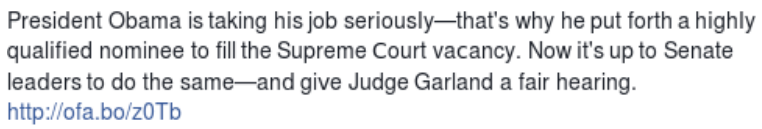
Per rispondervi è stata raccolta una grande quantità di post da utenti diversi, circa mille da ogni utente, in modo da garantire varietà nel tipo di linguaggio utilizzato; e sono state calcolate alcune statistiche sulla percentuale di post a cui si potesse applicare il watermark; inoltre , ognuno dei post a cui è stato possibile applicare il watermark è stato postato in versione originale e con watermark su un profilo appositamente creato, per poi ricavarne due screenshot e confrontare le immagini così ottenute, calcolandone la differenza.
Per ottenere una misura oggettiva di questa differenza è stato utilizzato MSE (mean squared error), scandendo pixel a pixel le immagini. [24]
Inoltre per avere un termine di confronto immediato l'immagine originale è stata confrontata, utilizzando lo stesso algoritmo, con due nuove versioni della stessa, in cui ogni riga appare spostata a destra o a sinistra (casualmente) di uno e di due pixel, vedi Figure 3.1(c) e 3.1(d). Il razionale di questa decisione è che spesso ciò che cambia con gli omoglifi è proprio la spaziatura, che quindi comporta un leggero spostamento complessivo delle righe.
A partire dalle tre serie di risultati derivanti rispettivamente dal confronto tra l'immagine originale e quella con watermark, con shift di un pixel e con shift di due pixel sono stati disegnati dei grafici, per mostrare i risultati in maniera immediata; inoltre vi è stato applicato il Test T [3, 6] , ovvero un test statistico atto a verificare se le differenze tra le serie di dati fossero semplicemente dovute al caso oppure fossero significative, in tal caso si è valutato quale fosse la migliore. Sono anche stati tracciati i grafici rappresentanti le funzioni di ripartizione, o funzione di probabilità cumulativa (CDF) ([4]) delle tre serie di dati; infine si è calcolato l'intervallo di confidenza al 95% relativo alla media dei risultati dell'algoritmo proposto ([5]).
La funzione di ripartizione F(x) ci dice con che probabilità i valori X di una serie di dati assumeranno valori minori a quello considerato, ovvero $P(X \leq x)$; ipotizziamo quindi che x sia 95, e F(x) sia 50%, questo vuol dire che nel 50%

dei casi otterremo risultati minori o uguali a 95.

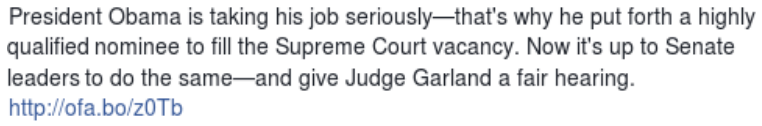
L'intervallo di confidenza invece ci da i limiti superiori ed inferiori rispetto alla media in cui troveremo il 95% dei nostri risultati, definendo così un intervallo che rappresenti meglio i risultati dei test, rispetto alla sola media.



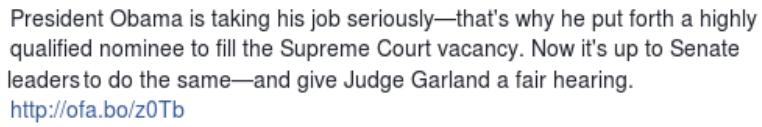
(a) Post originale



(b) Post con watermark



(c) Post originale a cui è stato applicato uno shift di12 px



(d) Post originale a cui è stato applicato uno shift di 2 px

Figura 3.1: Esempio di un post preso dal profilo Facebook di Obama, originale (a), con watermark (b) e con shift di 1 (c) e 2 (d) px.

3.4 Esperimento 4: Verificare la possibilità di veicolare messaggi tra piattaforme diverse preservando il watermark

Per rispondere a questa domanda, parallelamente all'esperimento precedente, sono state raccolte le percentuali relative alla quantità di post che rispettassero i vincoli imposti da ognuno dei social network. Nel dettaglio si è tenuto conto di quante stringhe provenienti da un social potessero essere pubblicate su di un altro, in base ai limiti di lunghezza imposti sul testo, e in tal caso se fosse possibile inserirvi il watermark.
Infatti i diversi social permettono testi di lunghezze diverse, e non tutti accettano lo stesso subset di caratteri confusable, per cui non è detto che una stringa a cui è stato applicato il watermark con successo per una piattaforma possa essere successivamente caricata anche su un'altra.

Capitolo 4

Risultati ottenuti

4.1 Risultati esperimento 1

Ricordiamo che il fine di questo esperimento è quello di determinare se le piattaforme facciano o meno watermarking su testo, il che comporterebbe una differenza tra la stringa postata sulla piattaforma e quello da essa recuperata.

Di seguito saranno esposti i risultati per ognuna delle piattaforme, riassunti in Tabella 4.1 .

Facebook: In questo caso le stringhe rappresentano post sulla bacheca di un utente, il che non ha imposto particolari vincoli sulla struttura delle stesse, se non una lunghezza massima di 60000 caratteri, difficilmente raggiungibili e sicuramente abbondanti per inserire il watermark.
Il test ha rivelato che nessuna delle tre stringhe testate ha subito modifiche, per cui si può dire che Facebook non fa watermarking testuale.
Facebook inoltre non comprime spaziature multiple, ma sostituisce le tabulazioni con dei normali spazi ed elimina i whitespace ad inizio e fine stringa, mentre accetta il carattere di newline.

Linkedin: In questo caso le stringhe rappresentano updates di un utente, la cui dimensione massima è di 700 caratteri, ragion per cui solamente le stringhe small e medium vengono accettate senza essere modificate, mentre la terza, eccedendo questo limite, viene rifiutata.
Inoltre spazi e tabulazioni vengono lasciati invariati, eccetto nel caso in cui questi si trovino all'inizio o alla fine del testo, in questo caso infatti vengono rimossi.
Il carattere di newline invece viene accettato.
Questo permette di concludere che Linkedin non faccia watermarking testuale.

Instagram: In questo terzo caso le stringhe rappresentano commenti ad un'immagine e non possono eccedere la dimensione di 300 caratteri.
Rispettando questo limite nella scelte dei testi non si notano modifiche, anche le tabulazioni vengono preservate, tuttavia whitespace multipli vengono compattati, e se presenti ad inizio o fine del testo vengono rimossi, mentre il carattere di newline viene sostituito con una normale spaziatura. Si conclude che nemmeno Instagram fa watermarking testuale.

Twitter: Quì le stringhe rappresentano tweet dell'utente, che però non possono eccedere la dimensione di 140 caratteri.
I tweet che rispettano questo limite vengono accettati senza essere modificai, ad eccezione delle tabulazioni , che vengono rimosse, così come i whitespace ad inizio e fine stringa, mentre il carattere di newline viene preservato. Possiamo quindi concludere che Twitter non fa watermaking testuale.

Gmail: In quest'altro caso, dato che le stringhe rappresentano il testo della mail, che può essere di lunghezza arbitraria, non ci sono limitazioni significative sul tipo di testo utilizzabile.
Inoltre non è stata rilevata alcuna modifica sui testi presi sotto esame, per cui possiamo dire che Gmail non fa watermarking testuale.

Google+: In questo caso il testo viene caricato come post, sul quale è imposto un limite di massimo 100 mila caratteri.
In un primo momento i confronti hanno portato a risultati negativi, mentre ad una più attenta analisi è risultato che alla fine di ogni stringa vengono aggiunti, apparentemente, 3 caratteri, che però rimangono invisibili nel font. Le codifiche ASCII di questi 3 caratteri sarebbero 239, 187 e 191, ma anche questa analisi non è completa, infatti a queste codifiche corrisponderebbero tre glifi visibili.
Se invece consideriamo la concatenazione dei tre caratteri come un'unica stringa e la decodifichiamo in UTF-8 otteniamo il carattere Unicode u'\ufeff', detto BOM (Byte Order Mark) , utilizzato in UTF-16 per definire l'utilizzo di big-endian piuttosto che little-endian [2].
Anche in questo caso possiamo dire che il social non fa watermarking testuale.

YouTube: Le stringhe rappresentano commenti ad un video, sui quali non è chiaro se ci sia o meno un limite di lunghezza, in particolare sembra che inizialmente fosse di 500 caratteri, poi esteso a 1000 ed infine rimosso completamente; per verificare questa eventualità sono stati effettuati test con testi anche superiori a 7100 caratteri, che sono stati accettati.
Ancora una volta nessuno dei testi utilizzati è stato modificato, se non per il fatto che i whitespace ad inizio e fine testo vengono rimossi, quindi Youtube non fa watermarking testuale.

Pinterest: Caricando il testo come commento ad un Pin si ha un limite massimo di 500 caratteri, ma rimanendo entro questa soglia, sia la stringa short che quella medium vengono accettate, mentre la stringa long viene troncata appunto a 500 caratteri. Si nota però che in alcuni casi questa piattaforma aggiunge uno spazio ad inizio e a fine stringa, questi spazi non hanno una codifica particolare, ma corrispondono allo spazio che si utilizza da tastiera, per cui limitandosi a rimuoverli si può verificare che le stringhe rimangono invariate.
Spaziature multiple, tab e new-line vengono sostituiti da un normale spazio

e i whitespace agli estremi vengono rimossi. Anche in questo caso possiamo dire che la pinterest non fa watermarking testuale.

Wordpress: In questo caso le stringhe vengono caricate come body di un articolo, sui quali non c'è alcuna limitazione di lunghezza.
Nessuno dei testi presi sotto esame è stato modificato, eccezion fatta per la rimozione di whitespace ad inizio e fine stringa.
Wordpress non fa watermarking testuale.

Tumblr: Su questa piattaforma il testo viene caricato come body di un post, senza limitazioni di lunghezza.
Anche in questo caso non è stata rilevata alcuna forma di modica dei testi, tuttavia tumblr non interpreta il carattere newline, e rimuove i whitespace ad inizio e fine stringa.
Tumblr non fa watermarking testuale.

Reddit: In questo caso il testo viene caricato sotto forma di commento ad un post, la cui lunghezza non può eccedere i 1000 caratteri.
Nessuna delle stringhe è stata modificata in alcun modo, quindi possiamo concludere che reddit non faccia watermarking testuale.

Vk: In questo caso le stringhe vengono caricate come post, su cui non ci sono limitazioni di lunghezza.
Questo social non ha modificato nessuna delle stringhe short, medium e large, tuttavia sostituisce le tabulazioni con dei normali spazi, non accetta i whitespace ad inizio e fine post e non interpreta il carattere newline.
Vk non fa watermarking testuale.

Flickr: Su questo Social il testo viene caricato come commento ad un'immagine, sul quale non sembrano esserci limitazioni di lunghezza, in particolare sono stati testati messaggi fino a 7700 caratteri e sono stati tutti accettati, incluse le stringhe di controllo short, medium e long.

Gli spazi multipli vengono compressi in uno solo, mentre le tabulazioni vengono rimosse, così come i whitespace agli estremi, viene invece accettato il carattere di newline.
Flickr non fa watermarking su testo.

Telegram: Su questa chat il testo rappresenta un messaggio, la cui lunghezza massima è di 4096 caratteri.
Per questo motivo tutte e tre le stringhe short, medium , long sono state accettate senza modifiche; vengono inoltre accettati spazi multipli e new-line, mentre le tabulazioni vengono sostituite con un normale spazio e i whitespace agli estremi vengono rimossi.
Possiamo dire che telegram non fa watermarking testuale.

Whatsapp: I messaggi di questa piattaforma non hanno limiti di lunghezza, ed anche in questo caso tutte e tre le stringhe sono state accettate senza modifiche.
Spazi multipli, tabulazioni e new-line vengono accettate, mentre i whitespace agli estremi vengono rimossi. Whatsapp non fa watermarking testuale.

Wechat: Questa chat non impone un limite al numero di caratteri, inoltre tutte le stringhe testate sono state accettate senza alcuna modifica, per cui anche in questo caso possiamo dire che Wechat non fa watermarking testuale.

Viber: Questa chat pone un limite di 7000 caratteri sulla lunghezza dei messaggi, per cui le tre stringhe short medium e long sono state accettate senza modifiche.
Gli spazi multipli e i new line vengono preservati, mentre le tabulazioni e i whitespace agli estremi vengono rimossi.
Viber non fa watermarking testuale.

QQ: Anche in questo caso non ci sono limiti sulla lunghezza del testo.
Le tre stringhe vengono accettate senza modifiche , mentre spazi multipli e

tabulazioni vengono sostituiti con un unico spazio. Il carattere di new-line viene accettato, ed anche gli spazi a fine stringa, ma non quelli all'inizio. QQ non fa watermarking testuale.

Come spiegato nel capitolo precedente in questo caso modifiche sui whitespace applicate dai diversi social non comportano la presenza di watermarking, ma evidenziano solo diverse politiche di gestione degli stessi.

Social	N. max caratteri	Short	Medium	Long	Spazi multipli	Tab	Whitespace ad inizio stringa	Whitespace a fine stringa	New-line
Facebook	60 000	✓	✓	✓	✓	✗	✗	✗	✓
Linkedin	700	✓	✓	✗	✓	✓	✗	✗	✓
Instagram	300	✓	✗	✗	✗	✓	✗	✗	✗
Twitter	140	✓	✗	✗	✓	✗	✗	✗	✓
Gmail	∞	✓	✓	✓	✓	✓	✓	✓	✓
Google+	100 000	✓	✓	✓	✓	✓	✓	✓	✓
YouTube	∞	✓	✓	✓	✓	✓	✗	✗	✓
Pinterest	500	✓	✓	✗	✗	✗	✗	✗	✗
Wordpress	∞	✓	✓	✓	✓	✓	✗	✗	✓
Tumblr	∞	✓	✓	✓	✓	✓	✗	✗	✗
Reddit	1 000	✓	✓	✓	✓	✓	✓	✓	✓
Vk	∞	✓	✓	✓	✓	✗	✗	✗	✗
Flickr	∞	✓	✓	✓	✗	✗	✗	✗	✓
Telegram	4 096	✓	✓	✓	✓	✗	✗	✗	✓
Whatsapp	∞	✓	✓	✓	✓	✓	✗	✗	✓
Wechat	∞	✓	✓	✓	✓	✓	✓	✓	✓
QQ	∞	✓	✓	✓	✗	✗	✗	✓	✓
Viber	7 000	✓	✓	✓	✓	✗	✗	✗	✓

Tabella 4.1: Risultati esperimento 1. ✓ indica l'accettazione del tipo di stringa, ✗ ne indica invece il rifiuto o la modifica; ∞ indica che non vi è un numero massimo di caratteri utilizzabili.

4.2 Risultati esperimento 2

Per rispondere a questa domanda è stato necessario capire quali dei simboli unicode utilizzati dal nostro metodo venissero accettati da ognuna delle piattaforme, a tal fine è stato necessario postare stringhe contenenti la codifica di questi caratteri e verificare se venissero o meno interpretati.
È infatti possibile che alcuni caratteri non vengano interpretati dalla piattaforma, che in tal caso ne visualizzerebbe la codifica e non il glifo, minando così all'invisibilità del metodo; oppure che vengano sostituiti dai loro corrispettivi standard, distruggendo così parti di watermark.
I risultati sono riportati nelle tabelle seguenti, rappresentanti rispettivamente i caratteri e i whitespace.

Carattere originale	Codifica Unicode	Face-book	Link-edin	Insta-gram	Twit-ter	Gmail	Goo-gle+	You-tube	Pint-erest	Word-press	Tum-blr	Red-dit	VK	Fli-ckr	Tele-gram	What-sapp	We-chat	QQ	Vi-ber
c	u'\u217D	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
C	u'\u216D	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
d	u'\u217E'	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
D	u'\u216E'	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
i	u'\u2170'	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
l	u'\u217C'	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
L	u'\u216C	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
M	u'\u216F'	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
v	u'\u2174'	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
V	u'\u2164'	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
x	u'\u2179'	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
X	u'\u2169	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
j	u'\u0458	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
K	u'\u212A	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
;	u'\u037E'	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
-	u'\u2010	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Tabella 4.2: Risultati esperimento 2, accettazione confusable, con glifo e codifica per python utilizzata dallo script

Codifica Unicode	Face-book	Link-edin	Insta-gram	Twit-ter	Gmail	Goo-gle+	You-tube	Pint-erest	Word-press	Tum-blr	Red-dit	VK	Fli-ckr	Tele-gram	What-sapp	We-chat	QQ	Vi-ber
u'\u2005'	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓
u'\u2007'	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓
u'\u202F'	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓
u'\u2009'	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓
u'\u205F'	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓
u'\u2004'	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓
u'\u2008'	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓

Tabella 4.3: Risultati esperimento 2, accettazione whitespace Unicode

I dati riportati in queste tabelle sono fondamentali, perché ci indicano quali caratteri è possibile utilizzare su ogni piattaforma, e conseguentemente, se si volesse veicolare tra queste un messaggio, qual è il subset di caratteri utilizzabile perché il watermark non vada perso; ovvero l'intersezione degli insiemi di caratteri utilizzabili su ognuna delle piattaforme considerate.
In particolare si nota che la maggior parte delle piattaforme accetta tutti i caratteri; su queste quindi non ci saranno problemi nel trasmettere lo stesso messaggio contenente il watermark, se non per i diversi vincoli di lunghezza del testo.
Facebook invece accetta una sola codifica alternativa dello spazio, mentre le altre vengono sostituite con lo spazio normalmente utilizzabile da tastiera. Similmente si comporta Telegram, che però non accetta nessuna codifica alternativa per gli spazi.
Questa limitazione può compromettere gravemente la possibilità di inserire il watermark, poiché gran parte di esso viene inserito proprio grazie agli spazi, inquanto per ognuno di questi sarebbe possibile inserire 3 bit di watermark.
Twitter invece non accetta la codifica alternativa per 'K' e ';', che quindi non potranno essere usati su questa piattaforma per inserire il watermark.
Questa limitazione è però meno influente della precedente, poiché questi due caratteri sono utilizzati molto meno frequentemente degli spazi, e permettono l'inserimento di un solo bit per volta.

4.3 Risultati esperimenti 3 e 4

Queste due prove sono state svolte parallelamente , poiché dipendono entrambe da risultati ottenuti a partire da un set di post pubblici.
Per questo motivo e per garantire eterogeneità nei post e in particolare nel linguaggio in essi utilizzato, sono stati scelti tre profili in lingua inglese e tre in italiano , di persone con profilo pubblico, appartenenti a categorie professionali diverse; in particolare tra politici, cantanti e giornalisti; questi sono Obama, la giornalista Christiane Amanpour, il rapper Macklemore, Renzi,

Vasco Rossi e Travaglio.
Ovviamente i profili pubblici sono stati scelti per la facilità nel reperire i post, mentre il razionale della scelta di personaggi noti è stato il fatto che i loro post potrebbero essere usati più facilmente per fare broadcast di contenuti.

I risultati delle prove sono stati divisi per provenienza e social network, ponendo attenzione alla percentuale di post a cui si potesse applicare il watermark e alla lunghezza dei testi presi in esame, che in alcuni casi è soggetta a limitazioni imposte dalle piattaforme.
La divisione per social network è stata fatta in quattro "gruppi", in base alle limitazioni scoperte nella fase precedente, in particolare Facebook, Twitter e Telegram sono stati considerati come casi a parte, mentre tutti gli altri social sono stati trattati assieme.

Per prima cosa è stato necessario scrivere altre tre versioni dell'algoritmo di watermarking , conseguentemente ai risultati dell'esperimento 2.
In particolare ne è stata fatta una versione per Facebook, che trattasse lo spazio come un normale carattere confusable, inserendo quindi un solo bit di watermark per ogni spazio, poiché questo social accetta solo due codifiche per le spaziature; una versione per Twitter, che semplicemente non utilizzasse i due caratteri confusable che questa piattafoma non accetta; ed una versione per Telegram, che utilizzasse solamente i caratteri confusable in tabella 4.2 .

Fatto questo sono stati raccolti i post, ottenendone la percentuali appena descritte, e i due screenshot con cui calcolare la similarità.
A partire dalle tre serie di risultati, ottenute dal confronto tra l'immagine con watermark e le tre immagini originale e con shift di uno e due pixel, è stato costruito un grafico che permettesse una valutazione immediata del valore di similarità tra le immagini.
Infine si è passati alle analisi statistiche, spiegate in dettaglio nel capitolo precedente.

Questa fase di analisi di similarità è stata però impossibile per Telegram, poiché non è stato possibile reperire gli screenshot del testo in modo automatico, possiamo però aspettarci dei risultati simili a quelli di Facebook poiché l'algoritmo di watermark differisce solamente per uno spazio.

Riporto inoltre che in alcuni casi le modifiche al testo sono tali da farlo andare a capo una volta in più, ottenendo così una nuova riga, questa differenza appare solo nella rappresentazione del testo, a cui non vengono effettivamente aggiunti dei new-line.
Bisogna però anche tener conto che questo fenomeno si può presentare più o meno frequentemente al variare delle dimensioni della finestra in cui il testo viene visualizzato, in particolare più questa sarà piccola, più è probabile che si presenti; inoltre difficilmente un utente noterebbe che il testo anzi che essere rappresentato in n righe lo sia in n+1 se non avendo a disposizione entrambe le versioni dello stesso.
Siccome quindi questo fenomeno non dipende dall'algoritmo di watermarking i casi in cui si è presentato sono stati esclusi dal confronto tra immagini per non falsare le valutazioni sulla similarità visiva; ma è stata comunque misurata la percentuale con cui si sono presentati.

Di seguito sono riportati i risultati delle analisi, suddivisi per gruppo di appartenenza, come spiegato in precedenza; mentre in Tabella 4.8 sono riportati gli intervalli di confidenza per ogni Social e per ogni profilo, in modo da poter avere una visione complessiva dei risultai.

4.3.1 Facebook

In questo gruppo appare solamente questa piattaforma, in quanto è l'unica a non accettare alcuna codifica alternativa per lo spazio, se non per u'\u202F'.
Nella tabella seguente sono riportati i risultati raccolti durante questo esperimento.

	Totale campioni raccolti - Percentuale di successo	Numero post <1000 caratteri - Percentuale di successo sul totale	Numero post <700 caratteri - Percentuale di successo sul totale	Numero post <500 caratteri - Percentuale di successo sul totale	Numero post <300 caratteri - Percentuale di successo sul totale	Numero post <140 caratteri - Percentuale di successo sul totale	Percentuale post con una riga in più
Obama	1036 - 11%	1036 - 11%	1036 - 11%	1035 - 10.9%	1006 - 8%	719 - 0%	0%
Renzi	1084 - 63%	967 - 53%	866 - 43.7%	727 - 30.9%	563 - 15.8%	294 - 0.2%	2%
Amanpour	1075 - 28.8%	1073 - 28.7%	1072 - 28.6%	1061 - 27.5%	960 - 18.1%	495 - 0%	6.1%
Travaglio	1021 - 75.4%	289 - 3.7%	277 - 2.5%	273 - 2.1%	269 - 1.5%	243 - 0%	5.1%
Macklemore	1074 - 22.4%	1057 - 20.9%	1039 - 19.2%	1010 - 16.5%	974 - 8.8%	640 - 0%	6.6%
Vasco	1065 - 30%	968 - 20.7%	931 - 17.2%	900 - 13.7%	835 - 8.2%	705 - 0.1%	3%

Tabella 4.4: Risultati esperimento 3, percentuali di successo su Facebook

Con percentuale di successo si intende la percentuale di post raccolti con un numero sufficiente di caratteri per inserire i 64 bit di watermark.
Ad ogni post infatti è stato applicato l'algoritmo di watermarking, ed in caso di successo questo è stato ripubblicato, su di un account appositamente creato, con e senza watermark.
Si è tenuto conto inoltre delle limitazioni di lunghezza più significative imposte da diversi social, verificando la percentuale di post che potessero contenere il watermark nonostante tali limiti.
Questo ci permette di verificare in che percentuale un post proveniente da una piattaforma possa essere veicolato su di un'altra mantenendo il watermark; ad esempio dei 1036 post di Obama analizzati solo l'11% potrebbe essere caricato anche su Reddit e Linkedin, che impongono rispettivamente un limite di lunghezza dei post ai 1000 e 700 caratteri; mentre solamente l'8% su Instagram che impone un massimo di 300 caratteri ed infine lo 0% su Twitter, che impone il limite di 140 caratteri, oltre ai limiti aggiuntivi sui confusable utilizzabili per inserire il watermark.
La causa dei pessimi risultati nel watermark di testi di lunghezza inferiore ai 140 caratteri è data dalla grossa limitazione che impone facebook sull'utilizzo degli spazi nel watermarking stesso, combinata al piccolo numero di caratteri disponibili per inserire il watermark; infatti l'algoritmo di base ci permetterebbe di scrivere 3 bit di watermark per ogni spazio; mentre in questo caso ne possiamo scrivere solamente 1.

Nonostante questo però i grafici che seguono mostrano che la differenza tra il post originale e quello con watermark è tendenzialmente minore di quella a cui è stato applicato lo shift di un pixel ; e sempre migliore di quella a cui ne è stato applicato uno di due pixel. Questo ci dice che sarebbe quasi impossibile notare la presenza di watermarking senza avere a disposizione le due immagini da confrontare, ed anche in questo caso risulterebbe molto arduo ad occhio umano, come si può notare in Figura 3.1.
Di seguito sono illustrati nel dettaglio i risultati del confronto di immagini

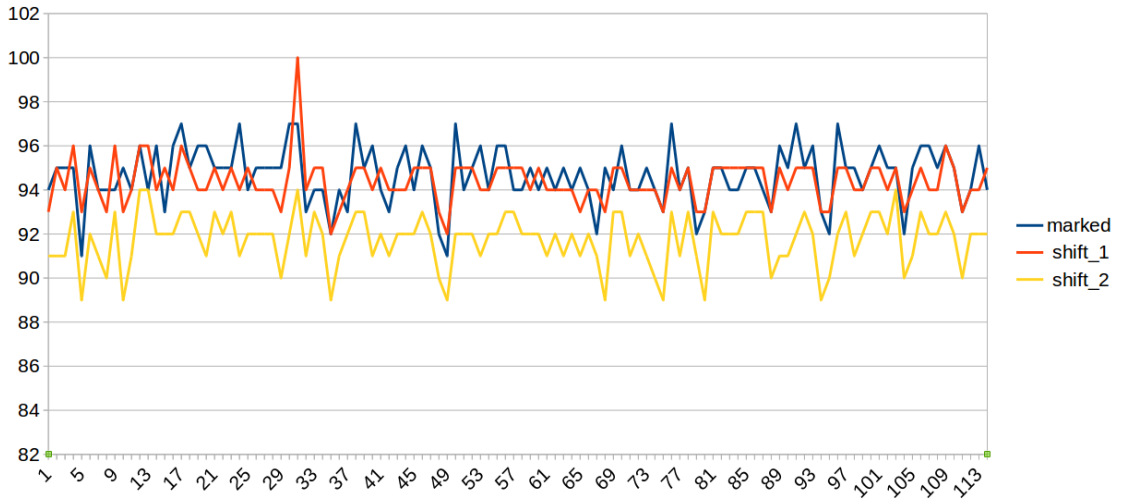
per ognuno dei soggetti presi in esame, dove con serie di dati marked, shifted 1 e shifted 2 si intendono rispettivamente i risultati del confronto con MSE tra il post originale e quello con watermark, quello avente le righe spostate di un pixel e quello avente le righe spostate di due pixel; inoltre dicendo che una serie di dati è migliore di un'altra si intende che le immagini da cui deriva la prima siano più simili agli originali, rispetto a quelle da cui deriva la seconda.

Obama: La media dei risultati della serie marked è di 94.68, di quella shifted 1 è di 94.39 e di quella shifted 2 di 91.74. Questo già ci da un'idea del fatto che i cambiamenti inseriti dall'algoritmo di watermarking sono meno significativi di un semplice shift, inoltre il test T ha determinato che le differenze tra le serie non siano dovute al caso confermando l'ipotesi appena formulata.
L'intervallo di confidenza del 95% delle tre serie è rispettivamente:

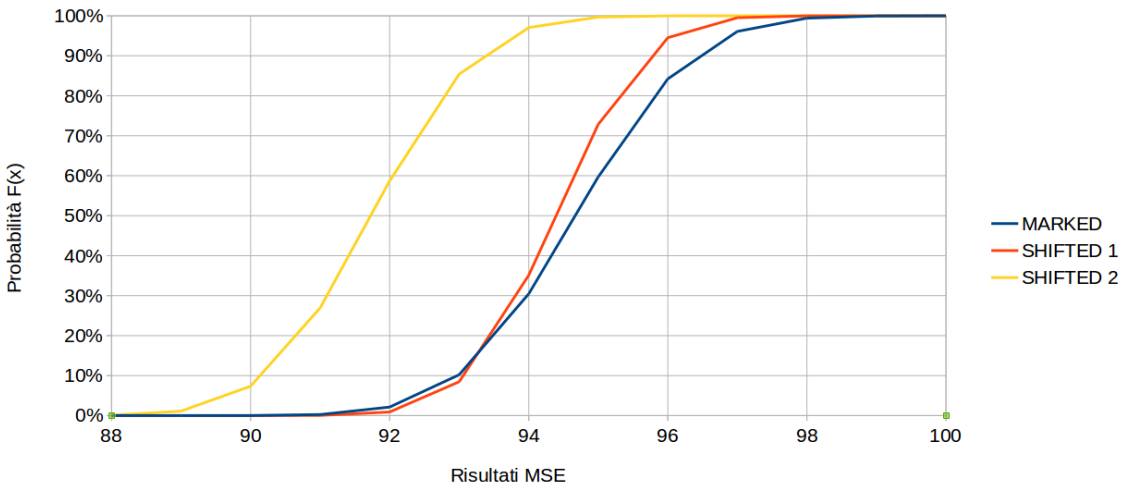
- 94.68 ± 0.24
- 94.39 ± 0.19
- 91.74 ± 0.22

Questi risultati ci mostrano che il limite superiore della serie di dati shifted 1 (94.58) è leggermente superiore al limite inferiore di quella marked (94.44), dove con limite superiore si intende la media sommata al valore di confidenza e analogamente per il limite inferiore, dove il valore di confidenza va invece sottratto. Questo ci dice che prendendo un valore appartenente all'intervallo [94.44 ; 94.58] non sapremo dire a quale delle due serie questo appartenga, tuttavia l'intervallo è estremamente piccolo, ragion per cui possiamo ancora considerare le due serie distinte.
La differenza con la serie shifted 2 invece è netta.
L'Immagine 4.1 (a) mostra graficamente quanto detto fin'ora, ovvero che la serie di dati marked è mediamente migliore di quella shifted 1 , e sempre migliore di quella shifted 2.

Il grafico in Figura 4.1 (b) mostra i dati relativi alla funzione di ripartizione delle tre serie di dati, in cui si nota che nelle due serie di dati shifted la probabilità di trovare risultati minori di un determinato valore è più alta rispetto a quella marked, il che conferma ancora una volta la positività di questi risultati.



(a) Valori di similarità ottenuti con MSE in scala 1-100.



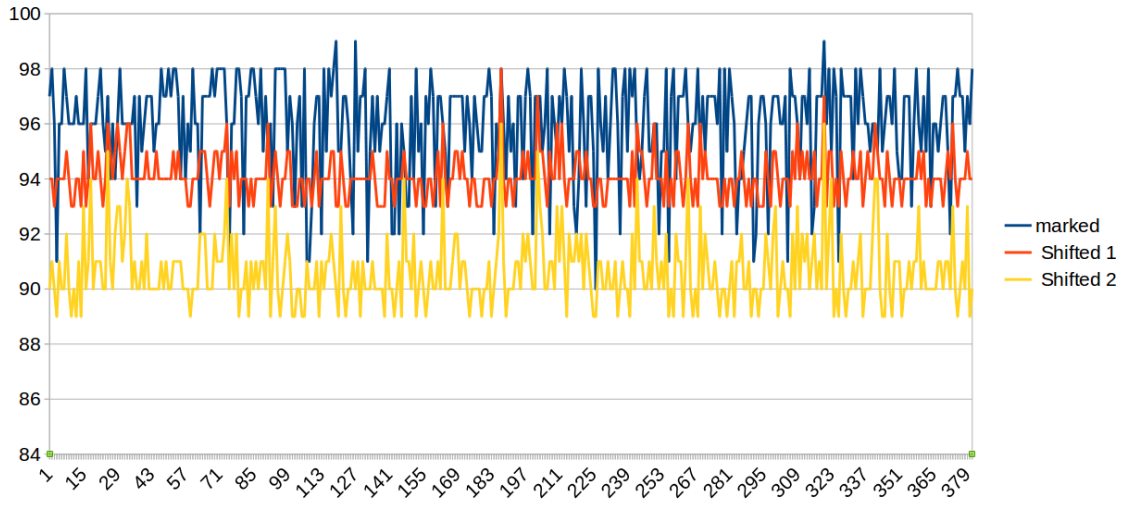
(b) Funzione di probabilità cumulativa CDF

Figura 4.1: Grafici di similarità (a) e CDF (b) relativi ai post di Obama su Facebook.

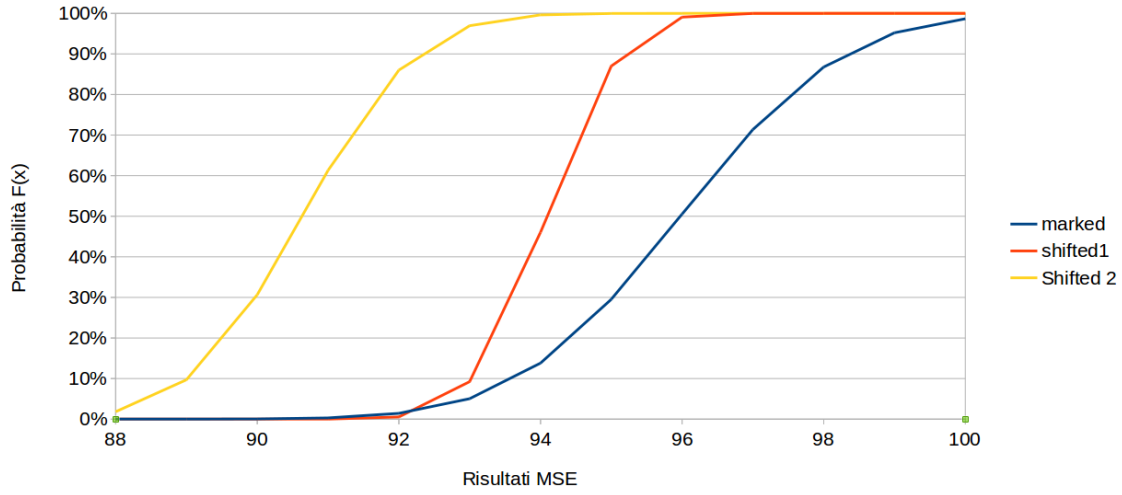
Renzi: La media dei risultati della serie marked è di 95.98, di quella shifted 1 è di 94.08 e di quella shifted 2 di 90.64. Il test T ha inoltre determinato che le differenze tra le serie non sono dovute al caso, possiamo quindi concludere che le immagini con watermark sono più simili alle originali rispetto a quelle a cui è stato applicato lo shift.
Per avere un'idea di quanto detto i risultati delle tre serie sono rappresentati graficamente in Figura 4.2 (a).
L'intervallo di confidenza del 95% delle tre serie è rispettivamente:

- 95.98 ± 0.18
- 94.08 ± 0.08
- 90.64 ± 0.13

Questi risultati ci mostrano che il limite inferiore della serie di dati marked (95.80) è comunque maggiore del limite superiore della serie shifted 1 (94.16) e quindi ovviamente maggiore al limite superiore della serie shifted 2 (90.77). Questo ci dice che dato un valore, nel 95% dei casi saremmo in grado di dire a che serie di dati appartiene, il che dimostra ancora una volta la differenza tra le tre serie ed in particolare la superiorità delle prima.
Il grafico in Figura 4.2 (b) mostra i dati relativi alla funzione di ripartizione delle tre serie, si nota che nelle due serie di dati shifted la probabilità di trovare risultati minori di un determinato valore è più alta rispetto a quella marked, a conferma della maggiore similarità dei i post con watermark.



(a) Valori di similarità ottenuti con MSE in scala 1-100.



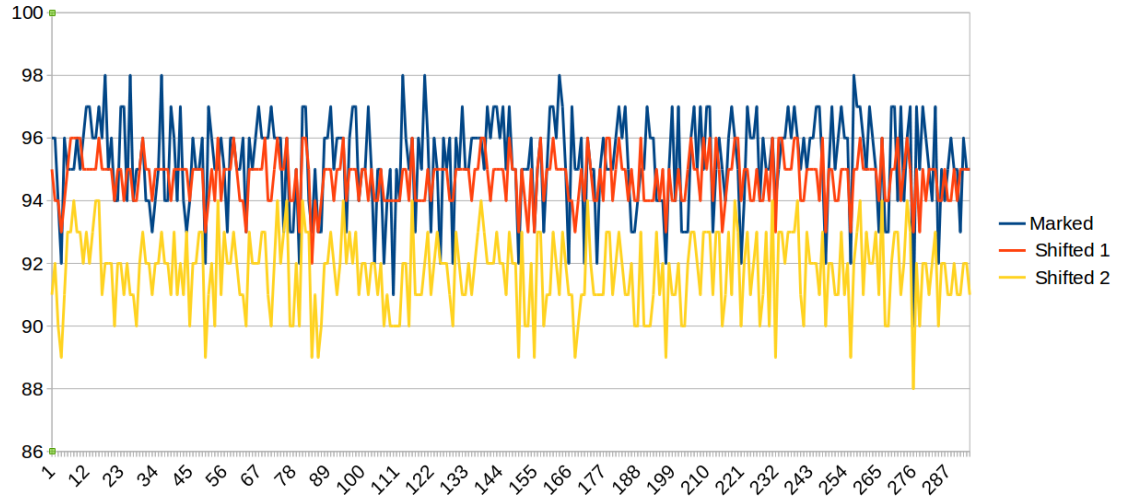
(b) Funzione di probabilità cumulativa CDF

Figura 4.2: Grafici di similarità (a) e CDF (b) relativi ai post di Renzi su Facebook.

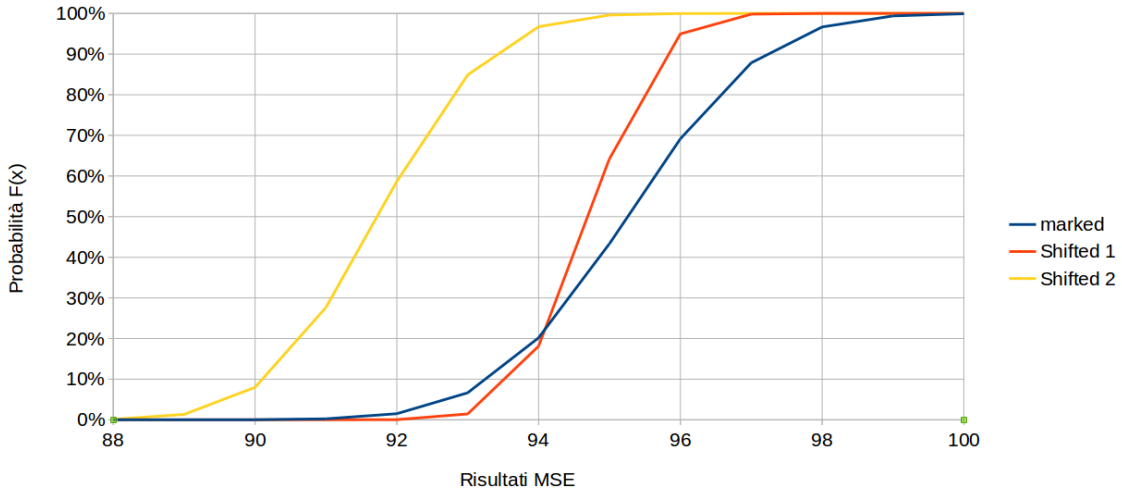
C. Amanpour: La media dei risultati della serie marked è di 95.25, di quella shifted 1 è di 94.71 e di quella shifted 2 di 91.73. Il test T ha inoltre determinato che le differenze tra le serie non siano dovute al caso, per cui le immagini con watermark sono più simili alle originali rispetto a quelle a cui è stato applicato lo shift. Per avere un'idea di quanto detto, i risultati delle tre serie sono rappresentati graficamente in Figura 4.3 (a).
L'intervallo di confidenza del 95% delle tre serie è rispettivamente:

- $95.25 \pm 0,17$
- $94.71 \pm 0,09$
- $91.73 \pm 0,14$

Similmente al caso precedente il limite inferiore della serie di dati marked (95.08) è maggiore del limite superiore della serie shifted 1 (94.80) e quindi ovviamente anche a quello della serie shifted 2 (91.87).
Il grafico in Figura 4.3 (b) mostra i dati relativi alla funzione di ripartizione delle tre serie, anche in questo caso possiamo notare che nelle due serie di dati shifted la probabilità di trovare risultati minori di un determinato valore è più alta rispetto a quella marked, eccetto nel primo tratto di grafico, in cui risulta leggermente migliore la serie shifted 1, in questo tratto comunque le probabilità rimangono molto basse e risultano quindi più significative le differenze nel tratto successivo.
Per questi motivi, anche in questo caso posiamo concludere che le immagini a cui è stato applicato il watermark siano comunque più simili alle originali rispetto a quelle in cui le righe sono state spostate di uno o due pixel.



(a) Valori di similarità ottenuti con MSE in scala 1-100.



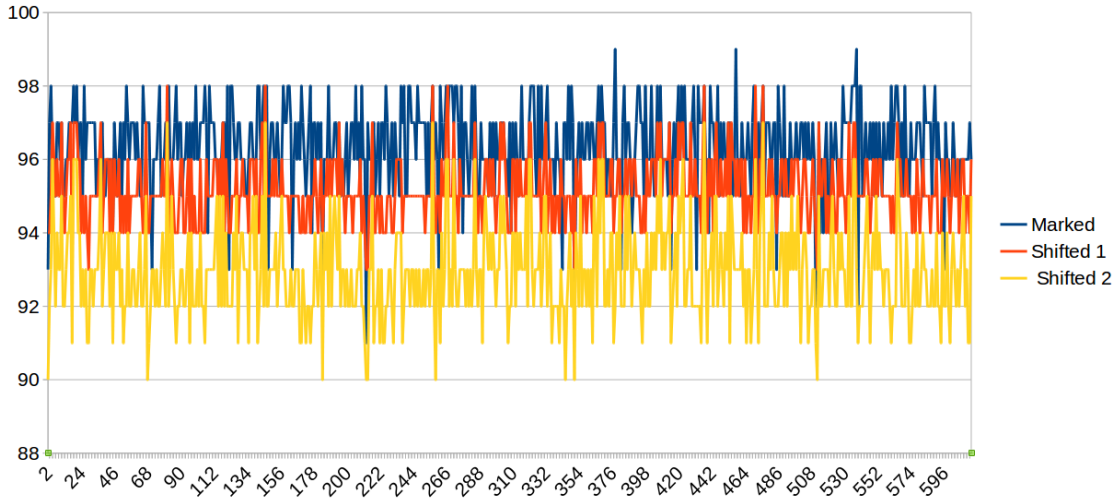
(b) Funzione di probabilità cumulativa CDF

Figura 4.3: Grafici di similarità (a) e CDF (b) relativi ai post di C. Amanpour su Facebook.

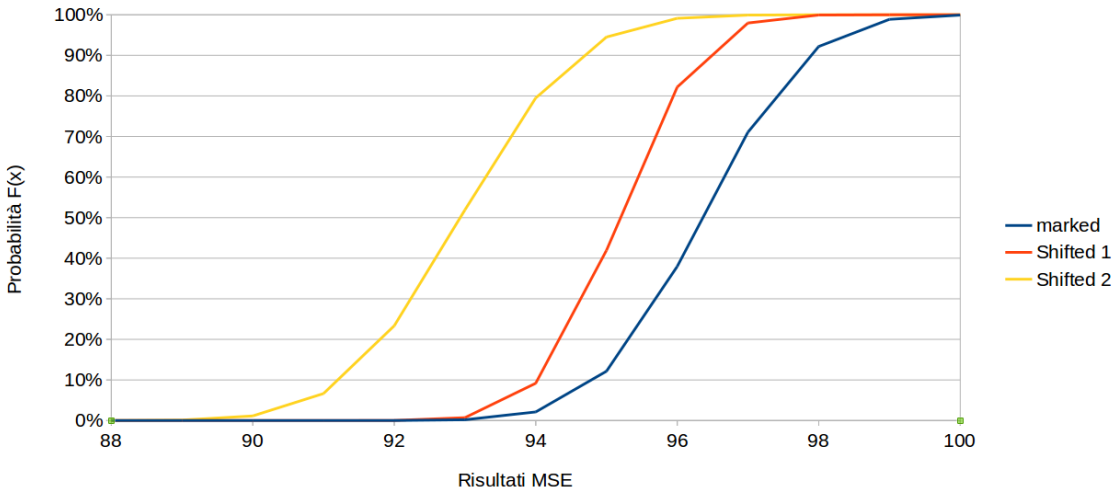
Travaglio: La media dei risultati della serie marked è di 96.36, di quella shifted 1 è di 95.18 e di quella shifted 2 di 92.94. Il test T ha inoltre determinato che le differenze tra le serie non siano dovute al caso, per cui anche in questo caso le immagini con watermark appaiono più simili alle originali rispetto a quelle a cui è stato applicato lo shift. Per avere un'idea di quanto detto, i risultati delle tre serie sono rappresentati graficamente in Figura 4.4 (a).
L'intervallo di confidenza del 95% delle tre serie è rispettivamente:

- 96.36 ± 0.09
- 95.18 ± 0.07
- 92.94 ± 0.10

Anche questa volta il limite inferiore della serie di dati marked (96.27) è maggiore del limite superiore della serie shifted 1 (95.25) e quindi ovviamente anche a quello della serie shifted 2 (93.04).
Il grafico in Figura 4.4 (b) mostra i dati relativi alla funzione di ripartizione delle tre serie, anche in questo caso possiamo notare che nelle due serie di dati shifted la probabilità di trovare risultati minori di un determinato valore è più alta rispetto a quella marked.
Per questi motivi, posiamo nuovamente concludere che le immagini a cui è stato applicato il watermark siano comunque più simili alle originali rispetto a quelle in cui le righe sono state spostate di uno o due pixel.



(a) Valori di similarità ottenuti con MSE in scala 1-100.



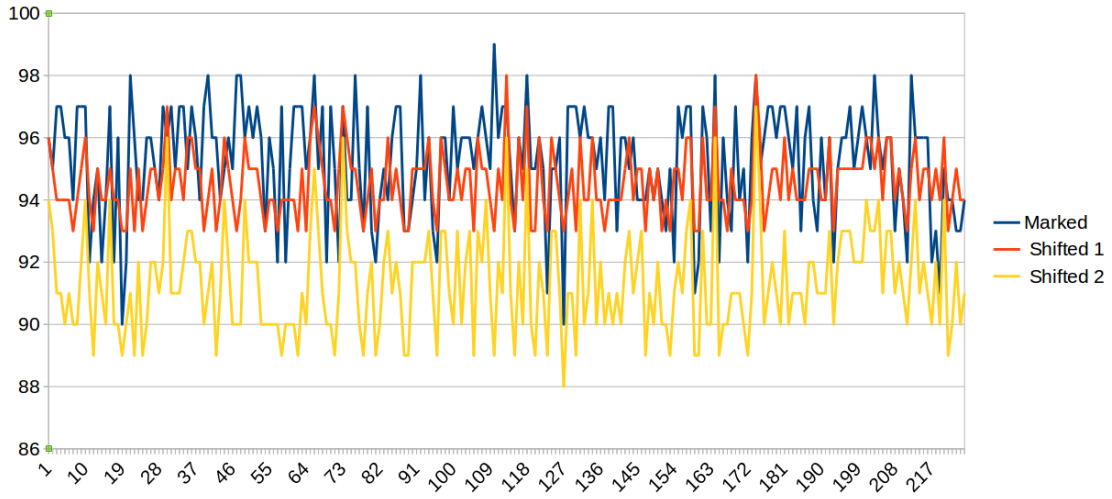
(b) Funzione di probabilità cumulativa CDF

Figura 4.4: Grafici di similarità (a) e CDF (b) relativi ai post di Travaglio su Facebook.

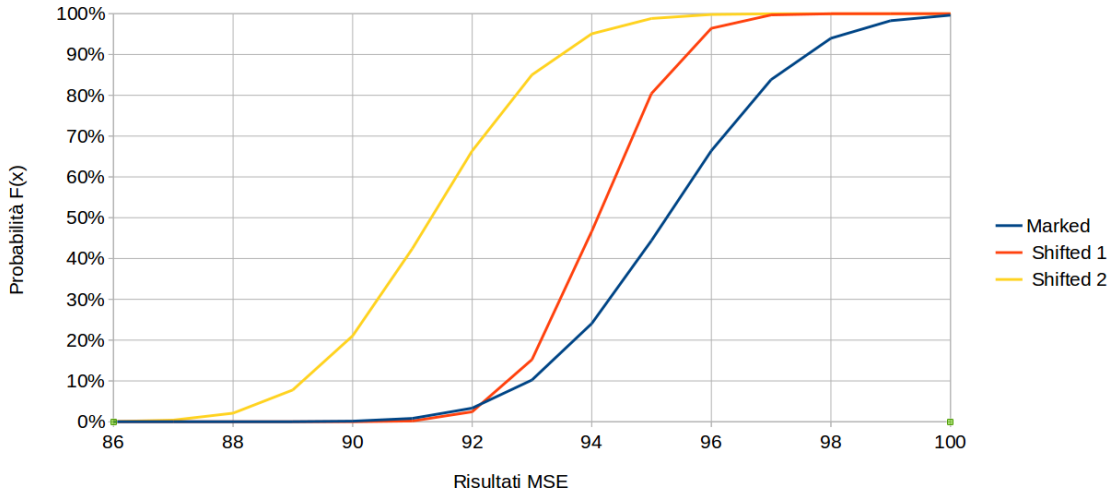
Macklemore: La media dei risultati della serie marked è di 95.25, di quella shifted 1 è di 94.09 e di quella shifted 2 di 91.31. Come nei casi precedenti il test T ha inoltre determinato che le differenze tra le serie non sono dovute al caso.
L'intervallo di confidenza del 95% delle tre serie è rispettivamente:

- 95.25 ± 0.23
- 94.09 ± 0.14
- 91.31 ± 0.21

Anche questa volta il limite inferiore della serie di dati marked (95.02) è maggiore del limite superiore della serie shifted 1 (94.23) e quindi ovviamente anche a quello della serie shifted 2 (91.52).
Il grafico in Figura 4.5 (a) mostra l'andamento delle tre serie di dati.
Il grafico in Figura 4.5 (b) mostra i dati relativi alla funzione di ripartizione delle tre serie, anche in questo caso possiamo notare che nelle due serie di dati shifted la probabilità di trovare risultati minori di un determinato valore è più alta rispetto a quella marked, eccetto nel primo tratto di grafico, fino a poco dopo 92, in cui risulta leggermente migliore la serie shifted 1 , in questo tratto comunque le probabilità rimangono molto basse, inferiori al 5% e risultano quindi più significative le differenze nel tratto successivo.
Tutti questi risultati mostrano la maggiore similarità tra le immagini con watermark rispetto a quelle a cui è stato applicato lo shift.



(a) Valori di similarità ottenuti con MSE in scala 1-100.



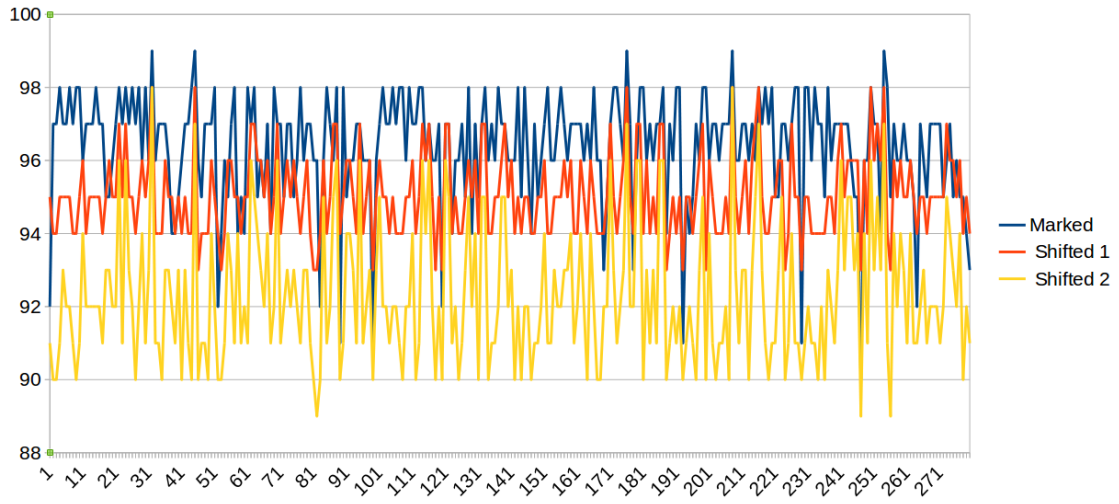
(b) Funzione di probabilità cumulativa CDF

Figura 4.5: Grafici di similarità (a) e CDF (b) relativi ai post di Macklemore su Facebook.

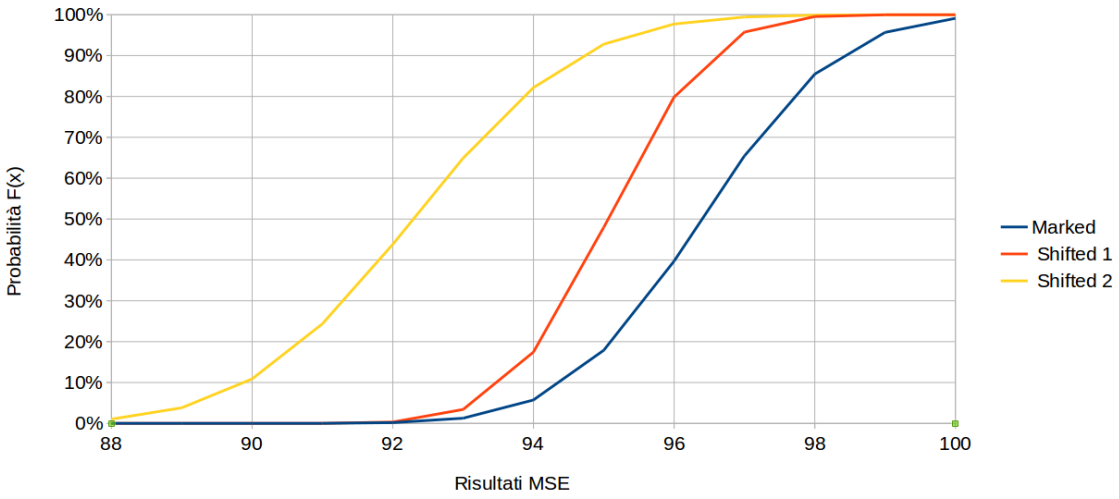
Vasco: La media dei risultati della serie marked è di 96.40, di quella shifted 1 è di 95.06 e di quella shifted 2 di 92.29. Come nei casi precedenti il test T ha inoltre determinato che le differenze tra le serie non siano dovute al caso, ed essendo la media della serie marked maggiore delle altre due possiamo dire che le immagini con watermark sono più simili alle originali rispetto a quelle a cui è stato applicato lo shift.
L'intervallo di confidenza del 95% delle tre serie è rispettivamente:

- 96.40 ± 0.18
- 95.06 ± 0.13
- 92.29 ± 0.22

Anche questa volta il limite inferiore della serie di dati marked (96.22) è maggiore del limite superiore della serie shifted 1 (95.19) e quindi ovviamente anche a quello della serie shifted 2 (92.51).
Il grafico in Figura 4.6 (a) mostra l'andamento delle tre serie di dati.
Il grafico in Figura 4.6 (b) mostra i dati relativi alla funzione di ripartizione delle tre serie, anche in questo caso possiamo notare che nelle due serie di dati shifted la probabilità di trovare risultati minori di un determinato valore è più alta rispetto a quella marked.
I risultati di questi tre esperimenti permettono di concludere che anche in questo caso i post a cui è stato applicato il watermark risultano più simili ai post originali di quelli a cui è stato applicato lo shift delle righe.



(a) Valori di similarità ottenuti con MSE in scala 1-100.



(b) Funzione di probabilità cumulativa CDF

Figura 4.6: Grafici di similarità (a) e CDF (b) relativi ai post di Vasco su Facebook.

4.3.2 Twitter

Similmente al gruppo precedente, abbiamo solo Twitter, poiché è l'unica piattaforma a non accettare le codifiche alternative per i caratteri 'K' e ';' . Si ricordi inoltre che questo social impone il limite di 140 caratteri, ovvero il limite più restrittivo sulla lunghezza del testo trovato durante questo studio, conseguentemente tutte le stringhe caricate su questa piattaforma potrebbero anche essere caricate sulle altre, a patto che rispettino anche i limiti imposti sull'utilizzo dei caratteri confusable.
Come nel caso precedente è stato raccolto un insieme di tweet di partenza, a cui si è cercato di applicare l'algoritmo di watermark specifico per twitter, ottenendo così le seguenti statistiche:

	Totale campioni raccolti	Percenduale di successo	Percentuale tweet con una riga in più
Obama	1 000	47.1%	2.4%
Renzi	1 000	56.5%	2.4%
Amanpour	1 000	68.4%	3%
Travaglio	895	62.8%	2%
Macklemore	1 000	25%	15%
Vasco	426	34.1%	17.6%

Tabella 4.5: Risultati esperimento 3, percentuali di successo su Twitter

Di questi post una percentuale pressoché nulla potrebbe essere caricata anche su Facebook o Telegram, poiché, come spiegato nella sezione precedente, la forte restrizione sulla lunghezza del testo, combinata all'impossibilità di scrivere 3 bit di watermark per ogni spazio rende davvero improbabile l'evenienza di avere a disposizione un numero sufficiente di caratteri per inserire il watermark.
Invece tutti i messaggi a cui è stato applicato il watermark con successo in questo gruppo potrebbero essere ripubblicati senza alcuna modifica sulle piattaforme appartenenti all'ultimo gruppo.

Potrebbe sembrare preoccupante la percentuale di post in cui si ritrova una riga in eccesso, ma bisogna tener conto che questa non è una misura effettiva, nel senso che dipende fortemente dalla dimensione della finestra con cui si è aperto il Social Network, e dalla dimensione del font utilizzato, che nel caso di twitter è molto grande; inoltre, non avendo a disposizione il post originale non è possibile notare il cambiamento e quindi avere informazioni sulla presenza del watermark.

Di seguito sono riportati nel dettaglio i risultati riguardanti lo studio di similarità per ognuno dei soggetti da cui sono stati raccolti i post.

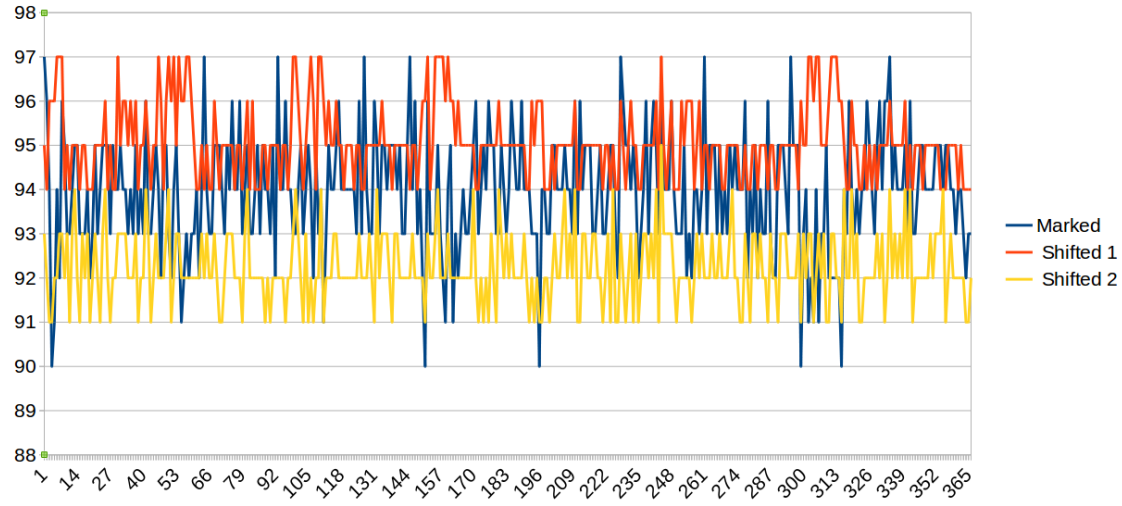
Obama: La media dei risultati della serie marked è di 93.94, di quella shifted 1 è di 95.05 e di quella shifted 2 di 92.22.
In questo caso si vede che la serie di dati marked è in media peggiore di quella shifted 1, ma rimane superiore di quella shifted 2.
Il test T ha inoltre determinato che le differenze tra le serie non siano dovute al caso, confermando quanto appena detto.
Per avere un'idea immediata di quanto detto i risultati delle tre serie sono rappresentati graficamente in Figura 4.7 (a).

L'intervallo di confidenza del 95% delle tre serie è rispettivamente:

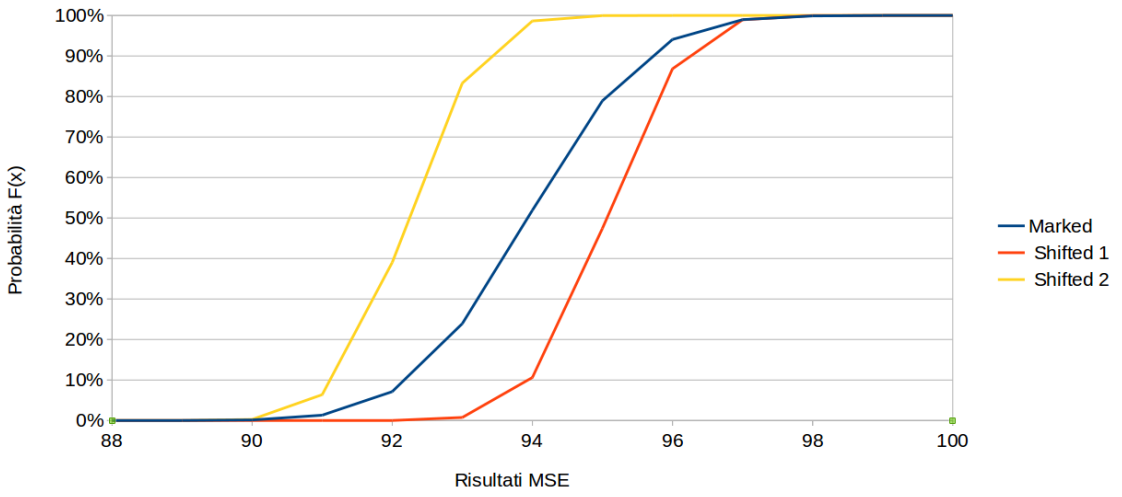
- 93.94 ± 0.14
- 95.05 ± 0.09
- 92.22 ± 0.08

Osservando i limiti superiori ed inferiori relativi alle tre classi notiamo che il limite inferiore della serie di dati shifted1 (94.96) è maggiore del limite superiore della serie marked (94.08); queste sono quindi disgiunte, ed in particolare i risultati di quest'ultima risultano peggiori, tuttavia rimangono migliori della serie shifted 2, il cui limite superiore (92.30) è ancora minore del limite inferiore (93.80) della serie di dati marked.

Il grafico in Figura 4.7 (b) mostra i dati relativi alla funzione di ripartizione delle tre serie, confermando quanto detto fin'ora, la funzione di ripartizione della serie marked è compresa tra le altre due, sarà quindi più probabile avere buoni risultati con questa rispetto alla serie shifted 2 , ma meno probabile rispetto alla serie shifted1.



(a) Valori di similarità ottenuti con MSE in scala 1-100.



(b) Funzione di probabilità cumulativa CDF

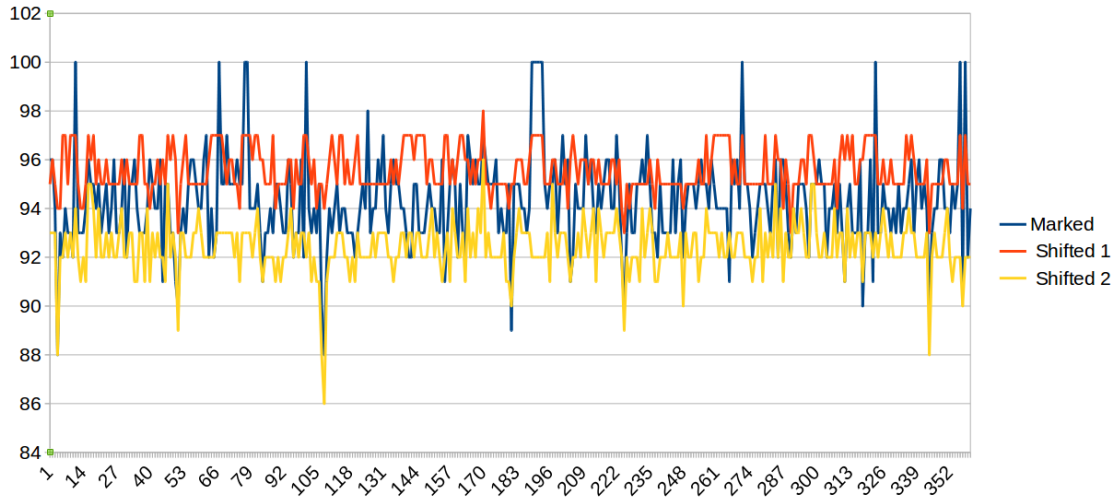
Figura 4.7: Grafici di similarità (a) e CDF (b) relativi ai post di Obama su Twitter.

Renzi: La media dei risultati della serie marked è di 94.30, di quella shifted 1 è di 95.56 e di quella shifted 2 di 92.39.
Anche questa volta si nota che la serie di dati marked è in media peggiore di quella shifted 1, ma rimane superiore di quella shifted 2.
Il test T ha inoltre determinato che le differenze tra le serie non siano dovute al caso, confermando quanto appena detto. Il grafico in Figura 4.8 (a) rappresenta le serie, mostrando in modo immediato i risultati appena esposti.
L'intervallo di confidenza del 95% delle tre serie è rispettivamente:

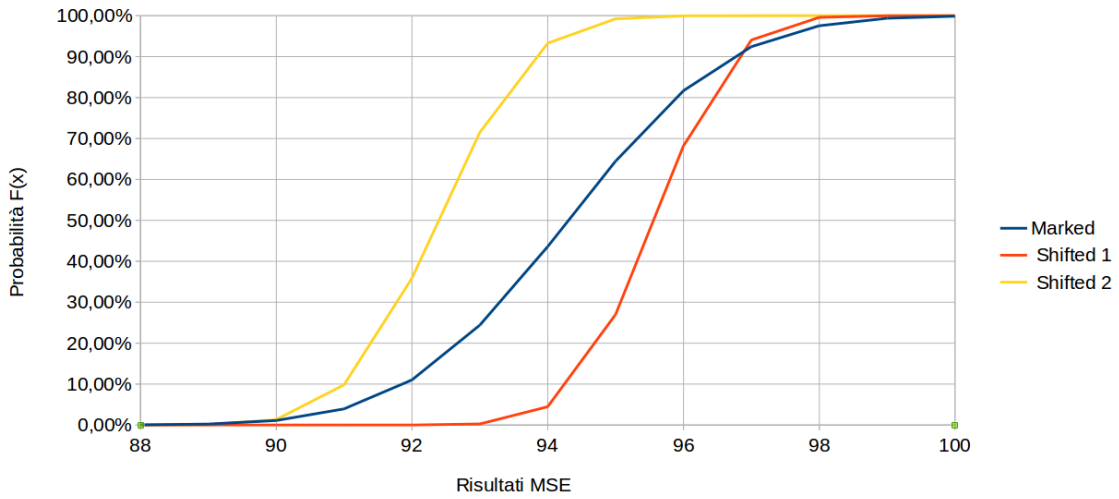
- 94.30 ± 0.19
- 95.56 ± 0.10
- 92.39 ± 0.11

Come nel caso precedente il limite inferiore della serie di dati shifted1 (95.46) è maggiore del limite superiore della serie marked (94.49); che risulta quindi peggiore, tuttavia rimane migliore della serie shifted 2, il cui limite superiore (92.50) è minore del limite inferiore (94.11) della serie di dati marked.

Il grafico in Figura 4.8 (b) mostra i dati relativi alla funzione di ripartizione delle tre serie, confermando quanto detto fin'ora, la funzione di ripartizione della serie marked è compresa tra le altre due, sarà quindi più probabile avere buoni risultati con questa rispetto alla serie shifted 2 , ma meno probabile rispetto alla serie shifted1, fa però eccezione il tratto di grafico successivo a 97, dove la serie di dati marked risulta leggermente migliore anche di quella shifted1.



(a) Valori di similarità ottenuti con MSE in scala 1-100.



(b) Funzione di probabilità cumulativa CDF

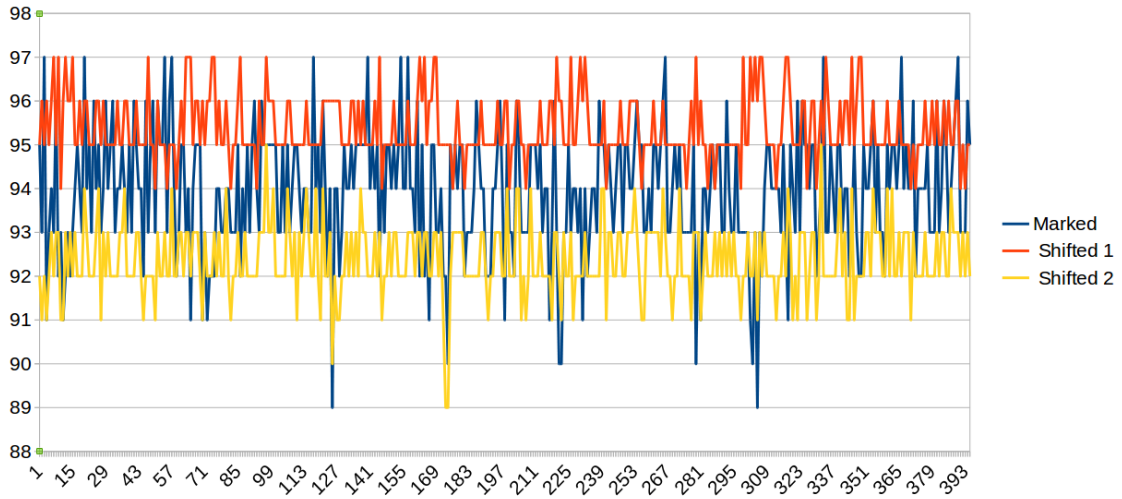
Figura 4.8: Grafici di similarità (a) e CDF (b) relativi ai post di Renzi su Twitter.

C. Amanpour: La media dei risultati della serie marked è di 93.92, di quella shifted 1 è di 95.36 e di quella shifted 2 di 92.37.
Anche questa volta si nota che la serie di dati marked è in media peggiore di quella shifted 1, ma rimane superiore di quella shifted 2.
Il test T ha inoltre determinato che le differenze tra le serie non siano dovute al caso, confermando quanto appena detto. Il grafico in Figura 4.9 (a) rappresenta le serie, mostrando in modo immediato questi stessi risultati.
L'intervallo di confidenza del 95% delle tre serie è rispettivamente:

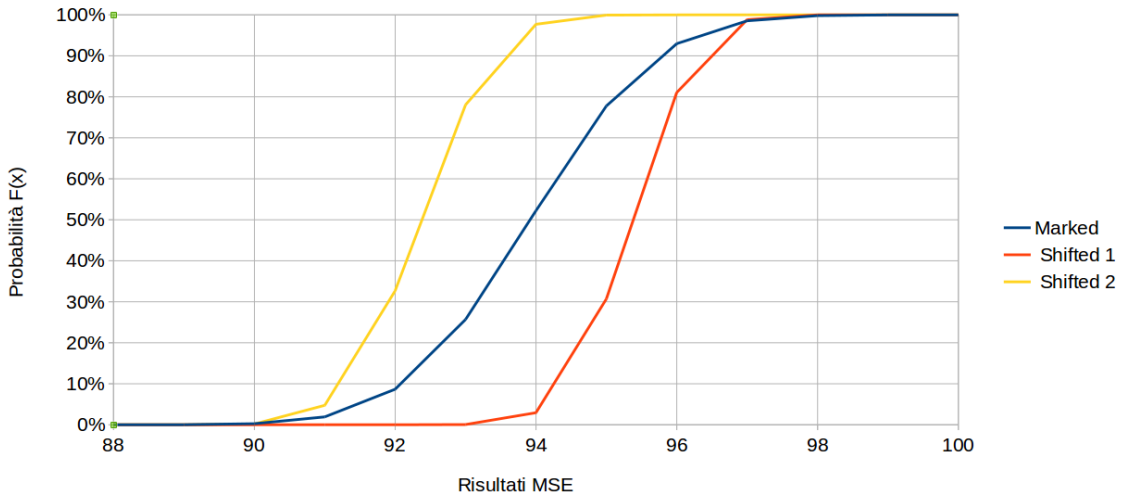
- 93.92 ± 0.14
- 95.36 ± 0.07
- 92.37 ± 0.08

Anche questa volta il limite inferiore della serie di dati shifted1 (95.29) è maggiore del limite superiore della serie marked (94.06); i risultati di quest'ultima sono quindi peggiori, tuttavia rimangono migliori della serie shifted 2, il cui limite superiore (92.45) è minore del limite inferiore (93.78) della serie di dati marked.

Il grafico in Figura 4.9 (b) mostra i dati relativi alla funzione di ripartizione delle tre serie, confermando quanto detto fin'ora, la funzione di ripartizione della serie marked è compresa tra le altre due, sarà quindi più probabile avere buoni risultati con questa rispetto alla serie shifted 2 , ma meno probabile rispetto alla serie shifted1.



(a) Valori di similarità ottenuti con MSE in scala 1-100.



(b) Funzione di probabilità cumulativa CDF

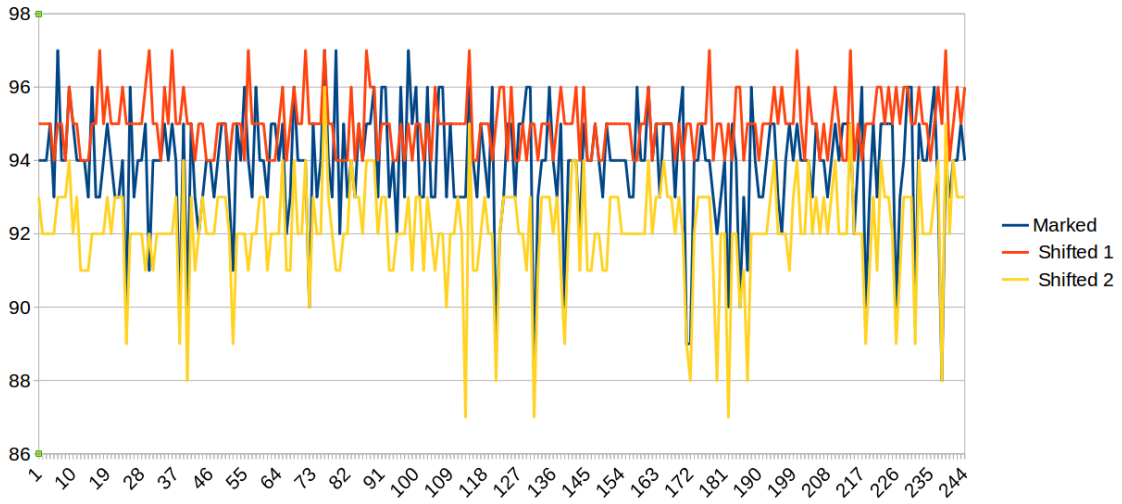
Figura 4.9: Grafici di similarità (a) e CDF (b) relativi ai post di C. Amanpour su Twitter.

Travaglio: La media dei risultati della serie marked è di 93.89, di quella shifted 1 è di 94.98 e di quella shifted 2 di 92.08.
Come nei casi precedenti si nota che la serie di dati marked è in media peggiore di quella shifted 1, ma superiore a quella shifted 2.
Questa ipotesi viene confermata dal test T, che ha rilevato che le differenze tra le serie di dati non sono dovute al caso. In Figura 4.10 (a) sono rappresentate graficamente le tre serie di dati.
L'intervallo di confidenza del 95% delle tre serie è rispettivamente:

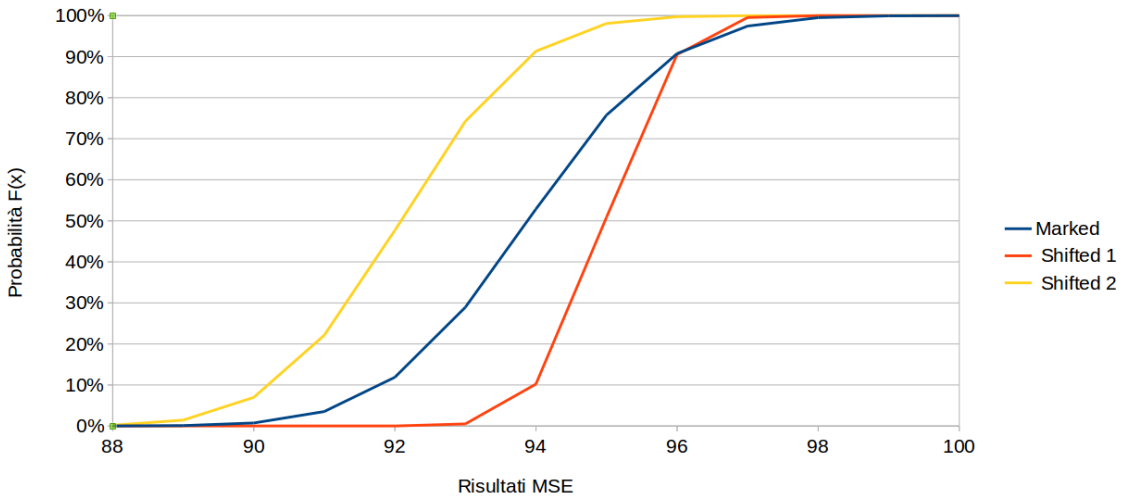
- 93.89 ± 0.20
- 94.98 ± 0.10
- 92.08 ± 0.18

Il limite inferiore della serie di dati shifted1 (94.88) è maggiore del limite superiore della serie marked (94.09); i risultati di quest'ultima appaiono quindi peggiori, tuttavia rimangono migliori della serie shifted 2, il cui limite superiore (92.26) è minore del limite inferiore (93.69) della serie di dati marked.

Il grafico in Figura 4.10 (b) mostra i dati relativi alla funzione di ripartizione delle tre serie, confermando quanto detto fin'ora, la funzione di ripartizione della serie marked è compresa tra le altre due, sarà quindi più probabile avere buoni risultati con questa rispetto alla serie shifted 2 , ma meno probabile rispetto alla serie shifted1, eccezion fatta per un breve tratto tra 96 e 98 in cui la serie marked risulta leggermente migliore anche di shifted1.



(a) Valori di similarità ottenuti con MSE in scala 1-100.



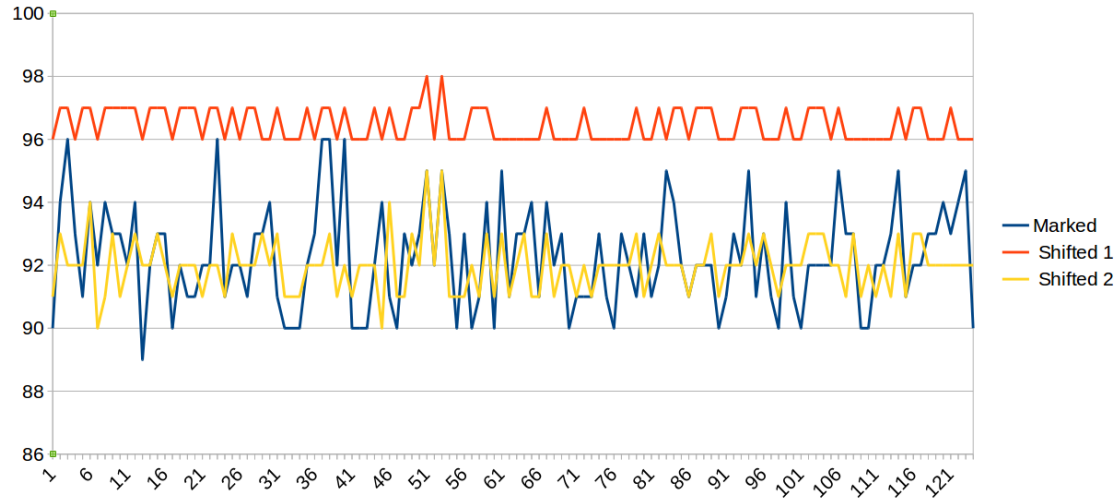
(b) Funzione di probabilità cumulativa CDF

Figura 4.10: Grafici di similarità (a) e CDF (b) relativi ai post di Travaglio su Twitter.

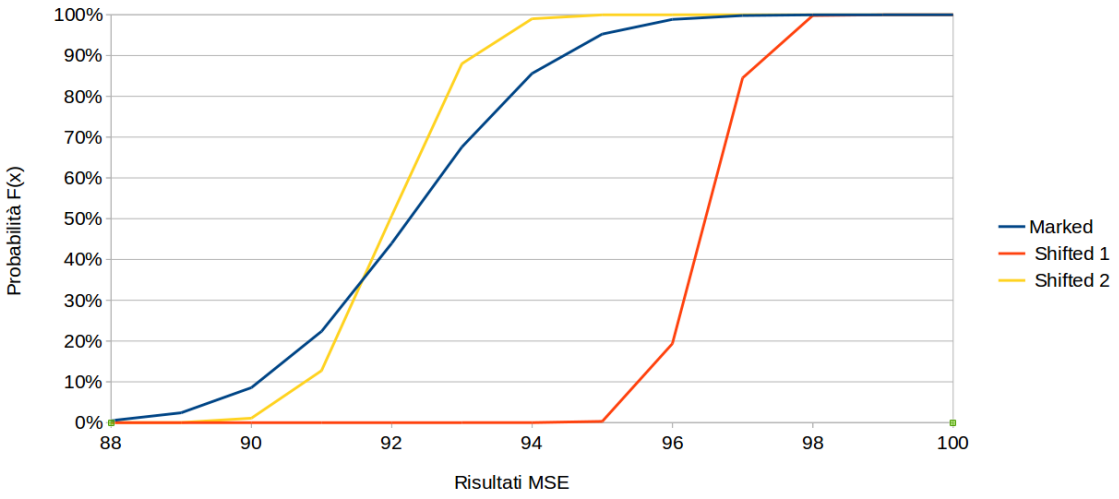
Macklemore: La media dei risultati della serie marked è di 92.25, di quella shifted 1 è di 96.46 e di quella shifted 2 di 91.98.
Il test T ha confermato che queste differenze non sono dovute al caso.
L'intervallo di confidenza del 95% delle tre serie è rispettivamente:

- 92.25 ± 0.29
- 96.46 ± 0.09
- 91.98 ± 0.15

Il limite inferiore della serie di dati shifted1 (96.37) è maggiore del limite superiore della serie marked(92.54); i risultati di quest'ultima risultano quindi peggiori, tuttavia rimangono migliori della serie shifted 2, nonostante il suo limite superiore (92.13) sia maggiore del limite inferiore (91,96) della serie di dati marked, questo ci dice che prendendo un valore appartenente all'intervallo [91,96 ; 92.13] non sapremo dire a quale delle due serie appartenga, tuttavia l'intervallo è estremamente piccolo, e possiamo quindi considerare le due serie distinte.
In Figura 4.11 (a) sono rappresentate graficamente le tre serie di dati.
Il grafico in Figura 4.11(b) mostra che per valori leggermente inferiori a 92 la serie con cui è più probabile avere risultati minori del valore preso sotto esame è proprio marked, tuttavia in questo intevallo le probabilità sono molto basse e risulterà quindi più significativo l'intervallo successivo, in cui la serie marked si colloca nuovamente tra le due shifted.
I risultati di questi tre esperimenti confermano ancora una volta che su questa piattaforma i post con watermark risultano più simili agli originali di quelli con shift di due pixel, ma meno di quelli con shift di un pixel.



(a) Valori di similarità ottenuti con MSE in scala 1-100.



(b) Funzione di probabilità cumulativa CDF

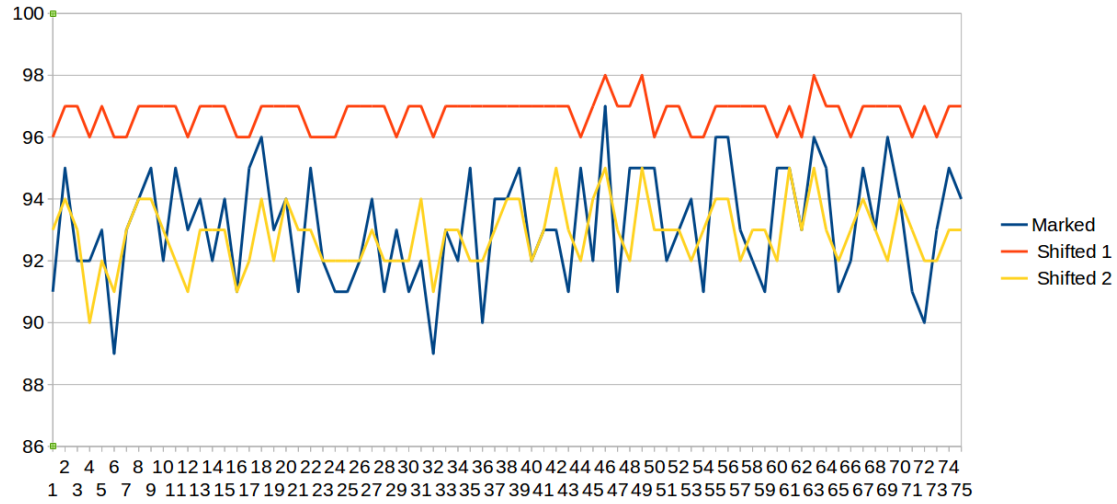
Figura 4.11: Grafici di similarità (a) e CDF (b) relativi ai post di Macklemore su Twitter.

Vasco: La media dei risultati della serie marked è di 93.11, di quella shifted 1 è di 96.76 e di quella shifted 2 di 92,85.
Il test T ha rivelato che le differenze tra marked e shifted 1 non siano dovute al caso, invece c'è una probabilità pari quasi il 9% che quelle tra marked e shifted 2 lo siano.
Le tre serie di dati sono rappresentate graficamente in Figura 4.12 (a).
L'intervallo di confidenza del 95% delle tre serie è rispettivamente:

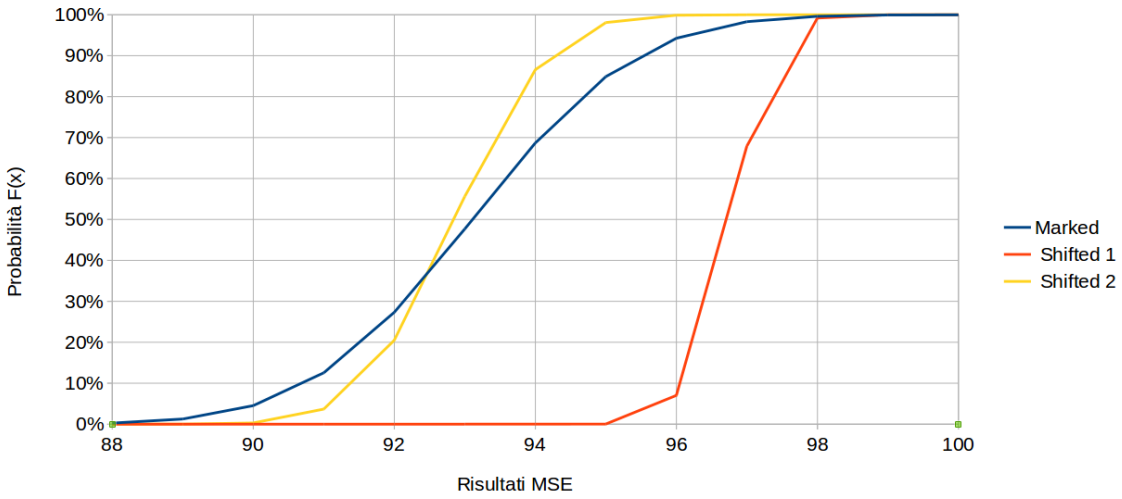
- $93.11 \pm 0,42$
- $96.76 \pm 0,12$
- $92,85 \pm 0,23$

Il limite inferiore della serie di dati shifted1 (96.64) è maggiore del limite superiore della serie marked(93.53); i risultati di quest'ultima risultano quindi peggiori, tuttavia rimangono migliori della serie shifted 2, nonostante il suo limite superiore (93.08) sia maggiore del limite inferiore (92.69) della serie di dati marked, questo ci dice che prendendo un valore appartenente all'intervallo [92.69 ; 93.08] non sapremo dire a quale delle due serie appartenga.

Il grafico in Figura 4.12(b) mostra che per valori leggermente inferiori a 93 la serie con cui è più probabile avere risultati minori del valore preso sotto esame è proprio marked, nel tratto successivo invece la serie marked si colloca nuovamente tra le due shifted.
Questa volta i risultati sono leggermente peggiori, ma complessivamente possiamo concludere che la qualità dei risultati della serie marked sia inferiore di quella shifted1 e superiore di quella shifted2.



(a) Valori di similarità ottenuti con MSE in scala 1-100.



(b) Funzione di probabilità cumulativa CDF

Figura 4.12: Grafici di similarità (a) e CDF (b) relativi ai post di Vasco su Twitter.

Concludendo, i risultati raccolti su questa piattaforma risultano peggiori di quelli raccolti su Facebook, tuttavia rimangono accettabili in quanto anche uno shift di due pixel risulta pressoché invisibile ad occhio nudo (si veda la Figura 3.1), e gli esperimenti permettono di concludere che la differenza causata dall'algoritmo di watermarking sia meno rilevabile di quest'ultimo. Il peggioramento è probabilmente dovuto alla brevità dei tweet; poiché per inserire i 64 bit di watermark sono spesso necessari tutti, o quasi, i 140 caratteri del testo; il che porta ad avere modifiche lungo l'intera stringa, mentre in post più lunghi le modifiche coprirebbero solo una parte del testo, risultando quindi meno significative, così come risulta meno evidente una goccia d'inchiostro in un lago, piuttosto che in un bicchiere d'acqua.

4.3.3 Telegram

Come nei due casi precedenti in questo gruppo abbiamo solo Telegram, la ragione di questa scelta sta nel fatto che questa piattaforma è l'unica a non accettare alcuna codifica alternativa per gli spazi, per cui per inserire il watermark potranno essere utilizzati solamente i caratteri confusable in tabella 2,1.
Per eseguire l'esperimento sono stati riutilizzati i post di partenza raccolti per il primo gruppo, quello di Facebook.

Nella tabella seguente sono riportati i risultati del test.

	Totale campioni raccolti - Percentuale di successo	Numero post <1000 caratteri - Percentuale di successo sul totale	Numero post <700 caratteri - Percentuale di successo sul totale	Numero post <500 caratteri - Percentuale di successo sul totale	Numero post <300 caratteri - Percentuale di successo sul totale	Numero post <140 caratteri - Percentuale di successo sul totale
Obama	1036 - 0.3%	1036 - 0.3%	1036 - 0.3%	1036 - 0.2%	1006 - 0%	719 - 0%
Renzi	1084 - 48.7%	967 - 28.6%	866 - 28.6%	727 - 15.8%	563 - 1.2%	294 - 0%
Amanpour	1075 - 4%	1073 - 3.7%	1072 - 3.7%	1061 - 2.7%	960 - 0%	495 - 0%
Travaglio	1021 - 73.9%	289 - 1.1%	277 - 1.1%	273 - 0.7%	269 - 0.2%	243 - 0%
Macklemore	1074 - 6.9%	1057 - 3.6%	1039 - 3.6%	1010 - 1.3%	974 - 0%	640 - 0%
Vasco	1065 - 21.5%	968 - 8.9%	931 - 8.9%	900 - 5.6%	835 - 0.7%	705 - 0%

Tabella 4.6: Risultati esperimento 3, percentuali di successo su Telegram

I pessimi risultati su questa piattaforma sono dovuti dalla totale impossibilità di utilizzare gli spazi per includere il watermark, infatti pressoché in ogni testo sono presenti degli spazi, mentre non è detto che lo siano i caratteri confusable utilizzati dall'algoritmo proposto, soprattutto non in sufficiente quantità.
Questi ipotesi è dimostrata dal fatto che solo nei post più lunghi le percentuali di successo sono buone, in particolare nei post più lunghi di 1000 caratteri di Travaglio e Renzi, i quali pubblicano spesso post molto lunghi, anche superiori ai 2000 caratteri.
Questo rende evidente l'impossibilità di veicolare messaggi con watermark tra Telegram e la maggior parte delle altre piattaforme, soprattutto quelle che non accettano testi di lunghezza superiore ai 1000 caratteri.
Come già detto su questa piattaforma non è stato possibile reperire gli screenshot, per cui non è stato effettuato lo studio sulla similarità dei post, tuttavia possiamo aspettarci dei risultati simili a Facebook.
Infine nella tabella non è stata inserita la percentuale di post con una riga in più, poiché anche questa veniva calcolata in fase di analisi degli screenshot.

4.3.4 Tutte le altre piattaforme

In questo gruppo appaiono invece tutte le piattaforme rimanenti, la ragione di questa scelta è che su ognuna di queste possiamo usare l'intera lista di caratteri che utilizza l'algoritmo di watermarking proposto, per cui sarà sufficiente porre attenzione ai limiti di lunghezza dei post.
Per eseguire l'esperimento sono stati riutilizzati i post di partenza raccolti per il primo gruppo, quello su Facebook, questa volta però il watermarking è stato inserito utilizzando tutti i caratteri possibili, data l'assenza di limitazioni di questo tipo, ed entrambi i post, originale e con watermark, sono stati ripubblicati su alcune delle piattaforme rimanenti, principalmente Reddit e Wordpress per motivi di praticità.

	Totale campioni raccolti - Percentuale di successo	Numero post <1000 caratteri - Percentuale di successo sul totale	Numero post <700 caratteri - Percentuale di successo sul totale	Numero post <500 caratteri - Percentuale di successo sul totale	Numero post <300 caratteri - Percentuale di successo sul totale	Numero post <140 caratteri - Percentuale di successo sul totale	Percentuale post con una riga in più
Obama	1036 - 46.2%	1036 - 46.2%	1036 - 46.2%	1036 - 46.1%	1006 - 43.4%	719 - 15.6%	0%
Renzi	1084 - 85.1%	967 - 74.3%	866 - 65%	727 - 52.1%	563 - 37%	294 - 12.5%	3.6%
Amanpour	1075 - 74.6%	1073 - 74.4%	1072 - 74.3%	1061 - 73.3%	960 - 63.9%	495 - 20.8%	4.1%
Travaglio	1021 - 77.1%	289 - 5.5%	277 - 4.4%	273 - 3.2%	269 - 2.9%	243 - 1.1%	1.9%
Macklemore	1074 - 54.8%	1057 - 53.3%	1039 - 51.6%	1010 - 48.8%	974 - 41.2%	640 - 13.8%	2.5%
Vasco	1065 - 40.1%	968 - 31%	931 - 27.5%	900 - 13.7%	835 - 18.5%	705 - 6.4%	4.2%

Tabella 4.7: Risultati esperimento 3, percentuali di successo sulle piattaforme che non escludono alcuna codifica utilizzata dall'algoritmo proposto

In Tabella 4.7 sono riportati i risultati del test. La differenza, ed in particolare il miglioramento, tra questi risultati e quelli ottenuti da Facebook a partire dagli stessi post è data dal fatto che su queste piattaforme non ci sono vincoli sull'utilizzo di confusable; quindi è molto più probabile che ci siano un numero di caratteri sufficienti per nascondere i 64 bit di watermark. All'interno di questo gruppo di piattaforme è anche notevolmente più concreta la possibilità di veicolare un messaggio, ad esempio tutti i messaggi a cui è stato applicato un watermark con successo su Instagram potranno essere ripubblicati anche su Linkedin o Whatsapp e naturalmente anche viceversa, a patto che il messaggio in questione rispetti i limiti di lunghezza delle diverse piattaforme.

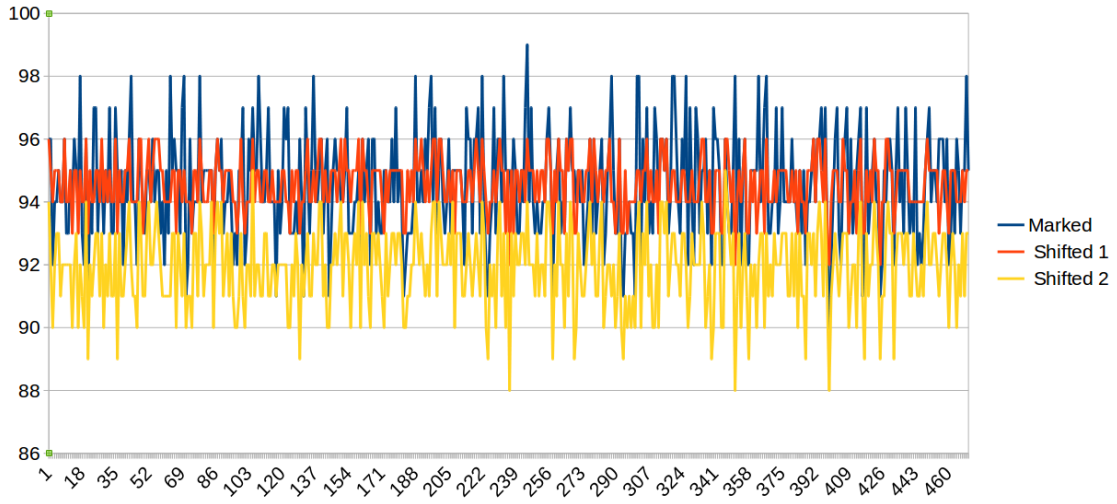
Di seguito sono riportati nel dettaglio i risultati degli esperimenti sulla similarità, anche questa volta con serie di dati marked, shifted 1 e shifted 2 si intendono rispettivamente i risultati del confronto con MSE tra il post originale e quello con watermark, quello avente le righe spostate di un pixel e quello avente le righe spostate di due pixel ; inoltre dicendo che una serie di dati è migliore di un'altra si intende che le immagini da cui deriva la prima siano più simili agli originali, rispetto a quelle da cui deriva la seconda.

Obama: La media dei risultati della serie marked è la stessa di quella shifted 1, ovvero 94.56, mentre di quella shifted 2 di 91.91.
Il test T ha determinato che le differenze tra la serie marked e quella shifted 1 potrebbero essere dovute al caso, con probabilità del 47%, questo, in altre parole, ci dice che possiamo considerare le due serie di dati come equivalenti; per quanto riguarda invece le differenze tra la serie marked e quella shifted 2 sono risultate non essere dovute al caso, e valutandone le medie possiamo concludere che la serie marked sia migliore di quella shifted2
L'intervallo di confidenza del 95% delle tre serie è rispettivamente:

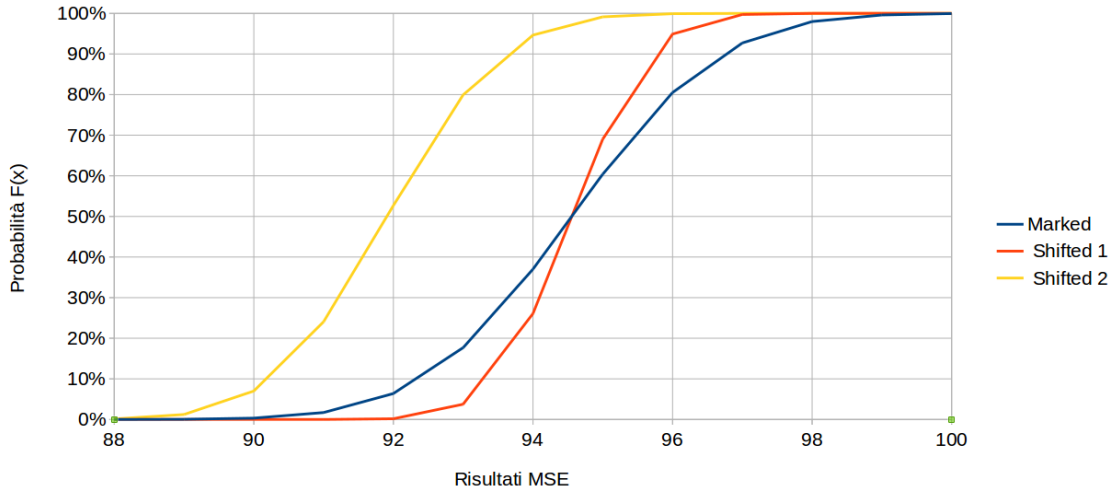
- 94.56 ± 0.15

- 94.56 ± 0.08
- 91.91 ± 0.12

Anche questi esperimenti ci mostrano che le prime due serie di dati hanno un comportamento molto simile, mentre il limite inferiore della serie marked(94.41) è nettamente maggiore del limite superiore della serie shifted 2(92.03), e quindi avendo un valore saremmo in grado di dire da quale delle due serie proviene L'Immagine 4.13 (a) mostra graficamente quanto detto fin'ora.
Il grafico in Figura 4.13 (b) mostra i dati relativi alla funzione di ripartizione delle tre serie di dati, dandoci forse le informazioni più dettagliate rispetto ai due test precedenti.
In particolare ci da conferma del fatto che l'andamento della serie marked sia migliore di shifted 2; invece, per quanto riguarda la serie marked e quella shifted1 possiamo notare che si incontrato in punto di poco inferiore a 95, nel tratto che precede questo punto risulta più probabile trovare risultati migliori nella serie shifted 1, e viceversa nel tratto che segue questo punto.
Ancora una volta abbiamo conferma di quanto visto negli esperimenti precedenti, le prime due serie hanno un comportamento paragonabile, mentre la serie marked risulta migliore della serie shifted2.



(a) Valori di similarità ottenuti con MSE in scala 1-100.



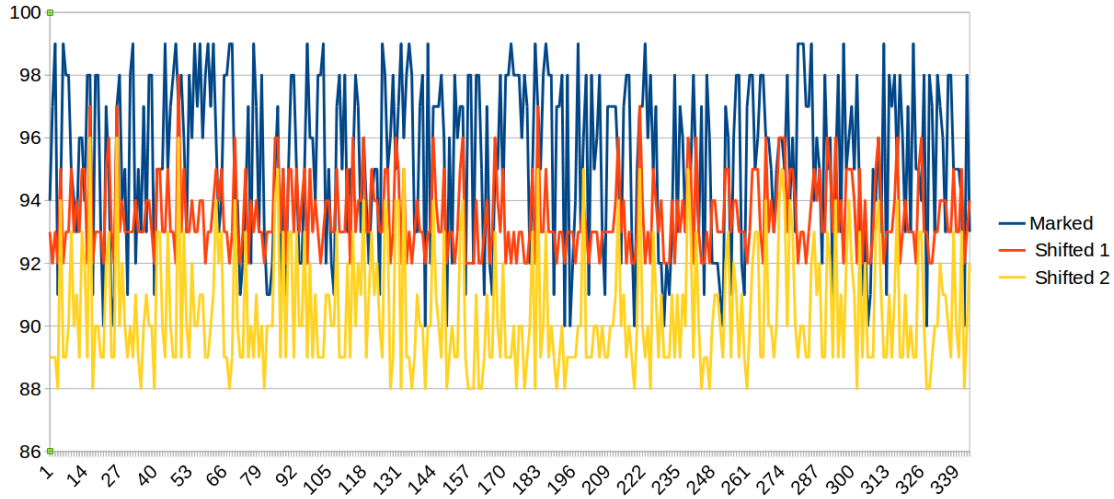
(b) Funzione di probabilità cumulativa CDF

Figura 4.13: Grafici di similarità (a) e CDF (b) relativi ai post di Obama sulle piattaforme che non impongono limiti all'insieme di caratteri utilizzati dall'algoritmo proposto.

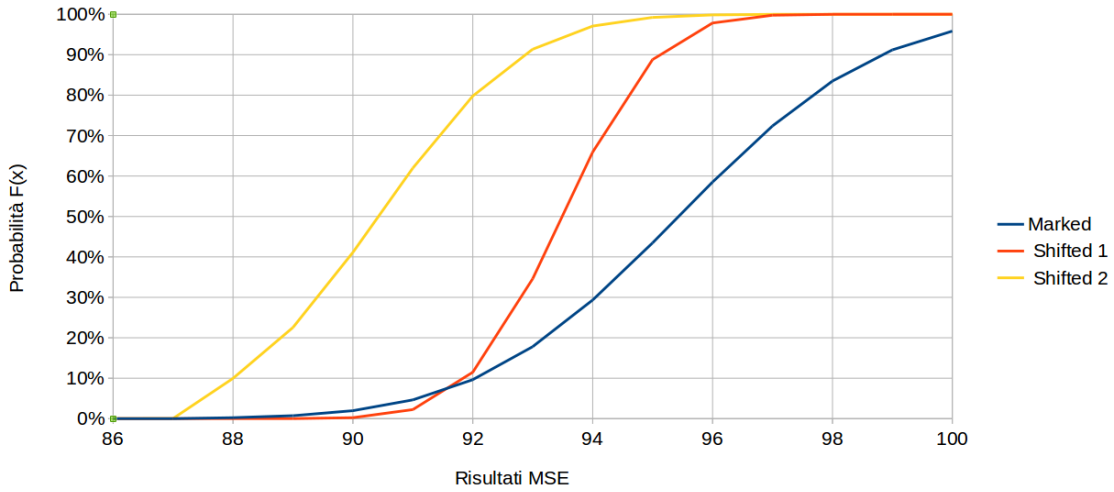
Renzi: La media dei risultati della serie marked è di 95.43, di quella shifted 1 è di 93.49 e di quella shifted 2 di 90.42.
Il test T ha determinato che le differenze tra le tre serie non sono casuali, possiamo quindi guardare alle medie per valutare quale sia la migliore.
L'intervallo di confidenza del 95% delle tre serie è rispettivamente:

- 95.43 ± 0.28
- 93.49 ± 0.13
- 90.42 ± 0.20

In questo caso osservando i limiti superiori ed inferiori dei tre intervalli di confidenza si nota che sono disgiunti, e quindi, avendo un valore saremo in grado di stimare con probabilità del 95% a quale serie questo appartenga. Questa è un'ulteriore conferma del fatto che le differenze tra le tre serie siano significative.
In Figura 4.14 (a) sono mostrati gli andamenti delle tre serie.
Il grafico in Figura 4.14 (b) mostra i dati relativi alla funzione di ripartizione delle tre serie di dati, dove possiamo notare che, eccetto in un primo breve tratto, dove comunque le probabilità sono molto basse, la serie di dati con cui è più probabile avere risultati positivi è shifted1.
Da questi tre esperimenti possiamo concludere che in questo caso le immagini con watermarking risultino più simili alle originali rispetto a quelle a cui è stato applicato lo shift.



(a) Valori di similarità ottenuti con MSE in scala 1-100.



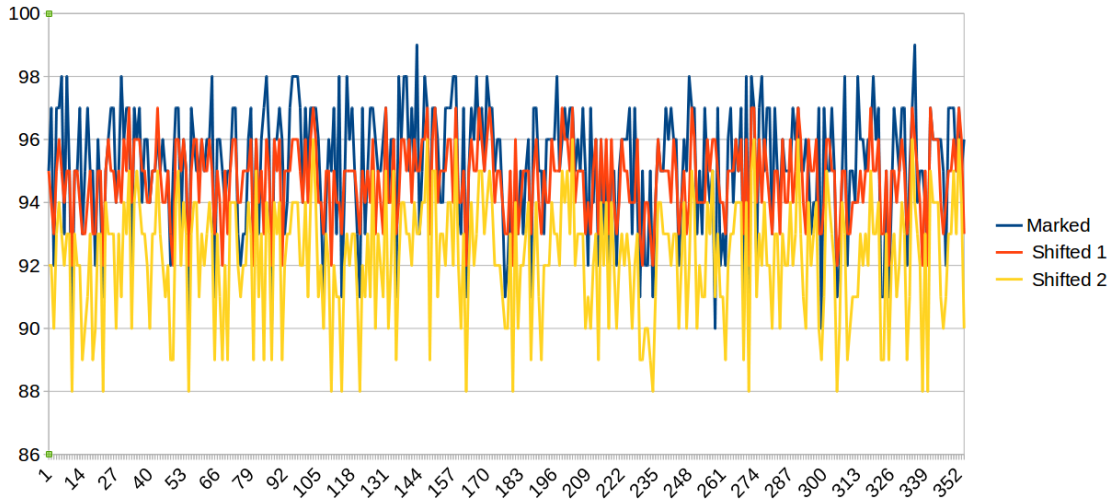
(b) Funzione di probabilità cumulativa CDF

Figura 4.14: Grafici di similarità (a) e CDF (b) relativi ai post di Renzi sulle piattaforme che non impongono limiti all'insieme di caratteri utilizzati dall'algoritmo proposto.

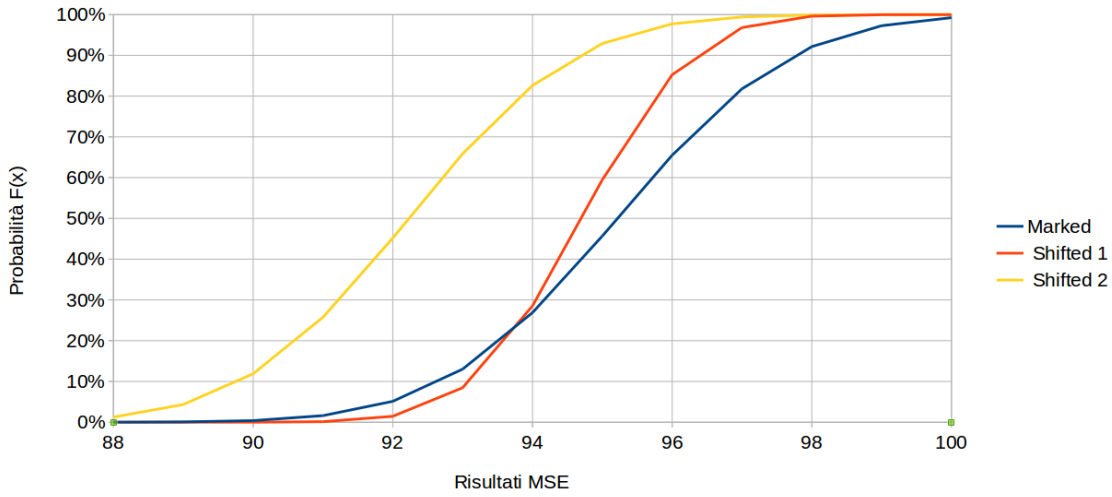
C. Amanpour: La media dei risultati della serie marked è di 95.21, di quella shifted 1 è di 94.70 e di quella shifted 2 di 92.23.
Anche in questo caso il test T ha determinato che le differenze tra le tre serie non sono casuali, possiamo quindi guardare alle medie per valutare quale sia la migliore.
L'intervallo di confidenza del 95% delle tre serie è rispettivamente:

- 95.21 ± 0.21
- 94.70 ± 0.13
- 92.23 ± 0.20

Similmente al caso precedente osservando i limiti superiori ed inferiori dei tre intervalli di confidenza si nota che sono disgiunti, e quindi, avendo un valore saremo in grado di stimare con probabilità del 95% a quale serie questo appartenga. Questa è un'ulteriore conferma del fatto che le differenze tra le tre serie siano significative.
In Figura 4.15 (a) si può avere un'idea grafica di quanto detto fin'ora
Il grafico in Figura 4.15 (b) mostra i dati relativi alla funzione di ripartizione delle tre serie di dati, dove possiamo notare che in un primo breve tratto la serie shifted 1 si comporta meglio della serie marked, tuttavia in questo intervallo le probabilità rimangono molto basse, per cui ritengo più significativo l'intervallo successivo, in cui la serie marked è quella per cui è meno probabile ottenere risultati minori del valore preso sotto esame.
Da questi tre esperimenti possiamo quindi concludere che anche in questo caso le immagini con watermarking risultino più simili alle originali rispetto a quelle a cui è stato applicato lo shift.



(a) Valori di similarità ottenuti con MSE in scala 1-100.



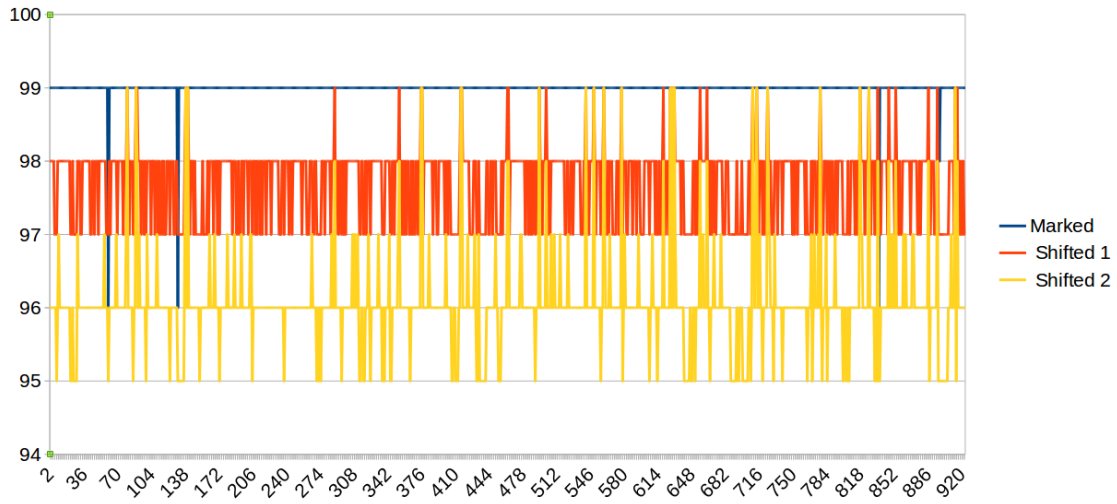
(b) Funzione di probabilità cumulativa CDF

Figura 4.15: Grafici di similarità (a) e CDF (b) relativi ai post di C. Amanpour sulle piattaforme che non impongono limiti all'insieme di caratteri utilizzati dall'algoritmo proposto.

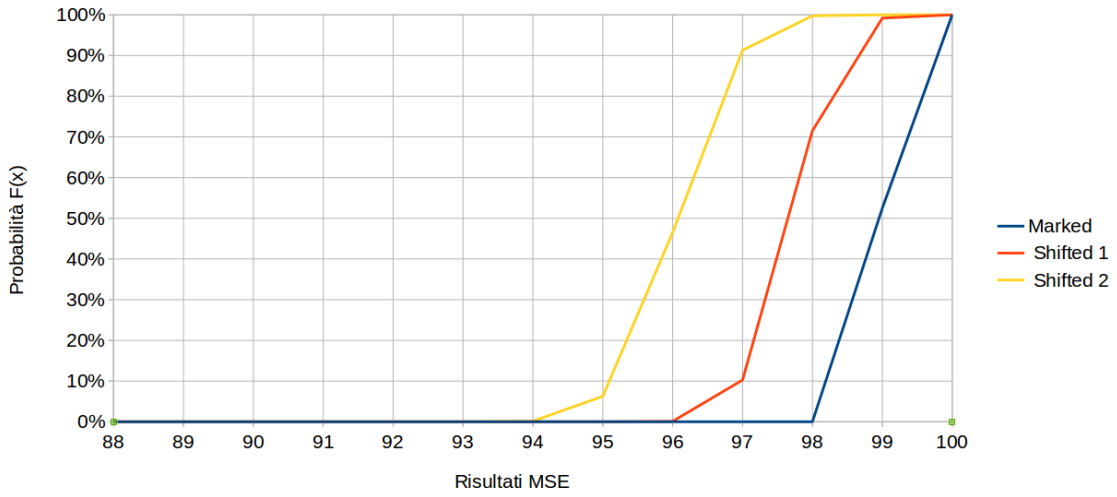
Travaglio: La media dei risultati della serie marked è di 98.99, di quella shifted 1 è di 97.69 e di quella shifted 2 di 96.06.
Anche in questo caso il test T ha determinato che le differenze tra le tre serie non sono casuali, possiamo quindi guardare alle medie per valutare quale sia la migliore.
L'intervallo di confidenza del 95% delle tre serie è rispettivamente:

- 98.99 ± 0.01
- 97.69 ± 0.04
- 96.06 ± 0.04

Similmente ai casi precedenti osservando i limiti superiori ed inferiori dei tre intervalli di confidenza si nota che sono disgiunti, e quindi, avendo un valore saremo in grado di stimare con probabilità del 95% a quale serie questo appartenga, avendo così un'ulteriore conferma del fatto che le differenze tra le tre serie siano significative.
In Figura 4.16 (a) si può avere un'idea grafica di quanto detto fin'ora, il motivo per cui la serie marked assume quasi sempre il valore di 99 è che i post di Travaglio superano spesso i due mila caratteri, e in un post di tale dimensione il watermark occupa solo una minima percentuale del testo, mentre la restante parte rimane così inalterata.
Il grafico in Figura 4.16 (b) mostra i dati relativi alla funzione di ripartizione delle tre serie di dati, dove possiamo notare la serie marked è quella per cui è meno probabile ottenere risultati minori del valore preso sotto esame.
Anche in questo caso dagli esperimenti possiamo concludere che le immagini con watermarking risultino più simili alle originali rispetto a quelle a cui è stato applicato lo shift.



(a) Valori di similarità ottenuti con MSE in scala 1-100.



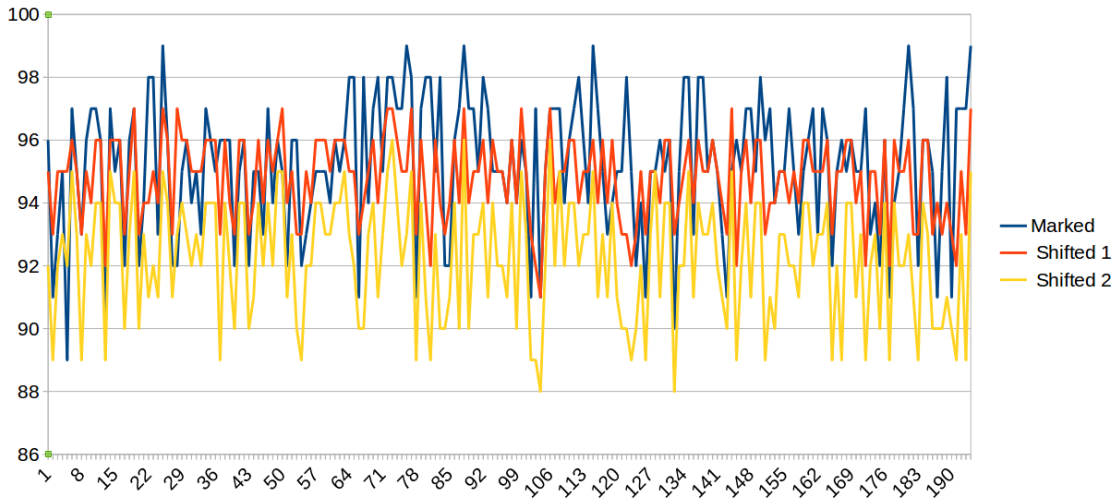
(b) Funzione di probabilità cumulativa CDF

Figura 4.16: Grafici di similarità (a) e CDF (b) relativi ai post di Travaglio sulle piattaforme che non impongono limiti all'insieme di caratteri utilizzati dall'algoritmo proposto.

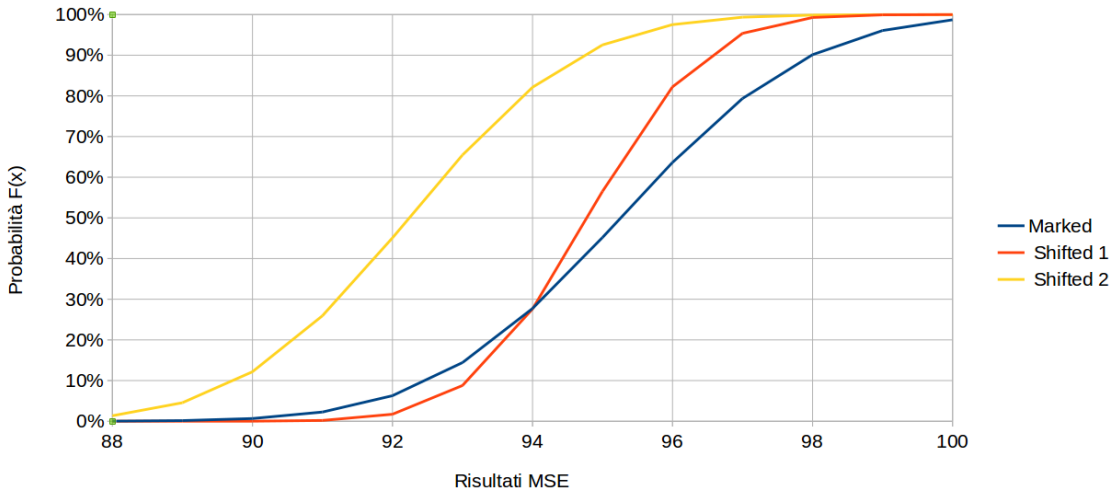
Macklemore: La media dei risultati della serie marked è di 95.26, di quella shifted 1 è di 94.78 e di quella shifted 2 di 92.24.
Anche in questo caso il test T ha determinato che le differenze tra le tre serie non sono casuali, possiamo quindi guardare alle medie per valutare quale sia la migliore.
L'intervallo di confidenza del 95% delle tre serie è rispettivamente:

- 95.26 ± 0.30
- 94.78 ± 0.19
- 92.24 ± 0.27

Similmente ai casi precedenti osservando i limiti superiori ed inferiori dei tre intervalli di confidenza si nota che sono disgiunti, e quindi, avendo un valore saremo in grado di stimare con probabilità del 95% a quale serie questo appartenga, avendo così un'ulteriore conferma del fatto che le differenze tra le tre serie siano significative.
In Figura 4.17 (a) si può avere un'idea grafica di quanto detto fin'ora.
Il grafico in Figura 4.17 (b) mostra i dati relativi alla funzione di ripartizione delle tre serie di dati, dove possiamo notare la serie shifted2 è la peggiore, come ci si aspettava, mentre tra la serie marked e la serie shifted1 risulta migliore la seconda in un primo tratto, fino a 94, tuttavia le differenze più significative si hanno nel secondo tratto, dove risulta migliore la serie marked.
Anche in questo caso quindi possiamo concludere che le immagini con watermarking risultino più simili alle originali rispetto a quelle a cui è stato applicato lo shift.



(a) Valori di similarità ottenuti con MSE in scala 1-100.



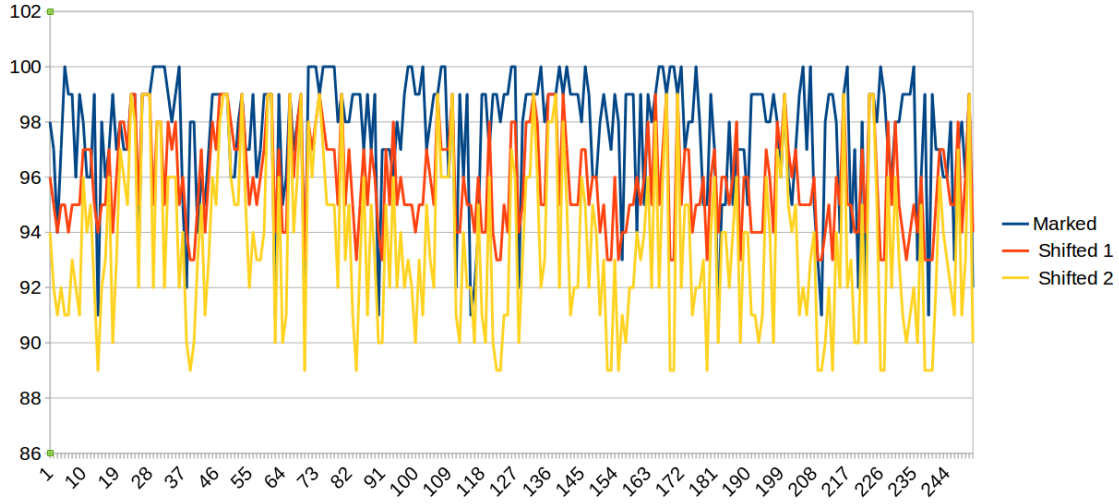
(b) Funzione di probabilità cumulativa CDF

Figura 4.17: Grafici di similarità (a) e CDF (b) relativi ai post di Macklemore sulle piattaforme che non impongono limiti all'insieme di caratteri utilizzati dall'algoritmo proposto.

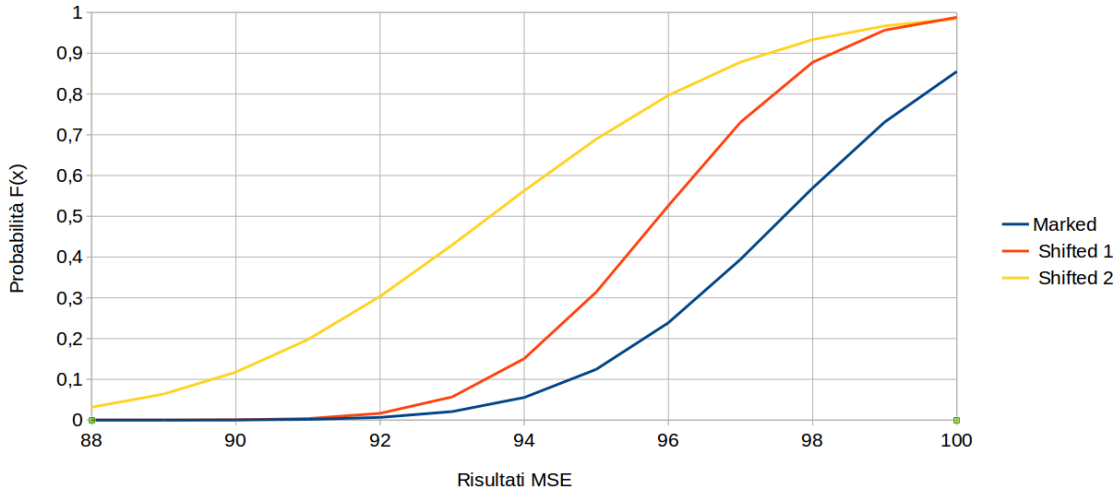
Vasco: La media dei risultati della serie marked è di 97.60, di quella shifted 1 è di 95.88 e di quella shifted 2 di 93.53.
Anche in questo caso il test T ha determinato che le differenze tra le tre serie non sono casuali, possiamo quindi guardare alle medie per valutare quale sia la migliore.
L'intervallo di confidenza del 95% delle tre serie è rispettivamente:

- 97.60 ± 0.28
- 95.88 ± 0.23
- 93.53 ± 0.37

Similmente ai casi precedenti osservando i limiti superiori ed inferiori dei tre intervalli di confidenza si nota che sono disgiunti, e quindi, avendo un valore saremo in grado di stimare con probabilità del 95% a quale serie questo appartenga, avendo così un'ulteriore conferma del fatto che le differenze tra le tre serie siano significative.
In Figura 4.18 (a) si può avere un'idea grafica di quanto detto fin'ora.
Il grafico in figura 4.18 (b) mostra i dati relativi alla funzione di ripartizione delle tre serie di dati, mostrando che preso un cerco valore, la serie che con minore probabilità otterrà risultati minori a questo è la serie marked.
Anche in questo caso quindi possiamo concludere che le immagini con watermarking risultino più simili alle originali rispetto a quelle a cui è stato applicato lo shift.



(a) Valori di similarità ottenuti con MSE in scala 1-100.



(b) Funzione di probabilità cumulativa CDF

Figura 4.18: Grafici di similarità (a) e CDF (b) relativi ai post di Vasco sulle piattaforme che non impongono limiti all'insieme di caratteri utilizzati dall'algoritmo proposto.

Similmente ai post caricati su Facebook, anche questa volta l'andamento della serie relativa alle immagini con Watermark è paragonabile o migliore a quella relativa all'immagine con shift di 1 pixel e migliore di quella relativa allo shift di 2 pixel. Questo comportamento era effettivamente prevedibile, poiché in questo caso i post considerati assumono lunghezze decisamente maggiori rispetto a quelli caricati su Twitter, e quindi il watermak occupa solo una parte dell'immagine totale, così come accade per Facebook; la differenza da quest'ultima piattaforma consiste solamente nel tipo di caratteri che vengono modificati, in particolare su Facebook è presente una sola codifica alternativa per lo spazio, mentre in quest'ultimo esperimento sono state utilizzate tutte quelle in tabella 2.2, producendo però risultati simili.

Tabella 4.8: Tabella riassuntiva di medie e intervalli di confidenza

	Profilo	Marked	Shifted1	Shifted2
Facebook	Obama	94.68 ± 0.24	94.39 ± 0.19	91.74 ± 0.22
	Renzi	95.98 ± 0.18	94.08 ± 0.08	90.64 ± 0.13
	Amanpour	$95.25 \pm 0,17$	$94.71 \pm 0,09$	$91.73 \pm 0,14$
	Travaglio	96.36 ± 0.09	95.18 ± 0.07	92.94 ± 0.10
	Macklemore	95.25 ± 0.23	94.09 ± 0.14	91.31 ± 0.21
	Vasco	96.40 ± 0.18	95.06 ± 0.13	92.29 ± 0.22
Twitter	Obama	93.94 ± 0.14	95.05 ± 0.09	92.22 ± 0.08
	Renzi	94.30 ± 0.19	95.56 ± 0.10	92.39 ± 0.11
	Amanpour	93.92 ± 0.14	95.36 ± 0.07	92.37 ± 0.08
	Travaglio	93.89 ± 0.20	94.98 ± 0.10	92.08 ± 0.18
	Macklemore	92.25 ± 0.29	96.46 ± 0.09	91.98 ± 0.15
	Vasco	$93.11 \pm 0,42$	$96.76 \pm 0,12$	$92,85 \pm 0,23$
Other	Obama	94.56 ± 0.15	94.56 ± 0.08	91.91 ± 0.12
	Amanpour	95.43 ± 0.28	93.49 ± 0.13	90.42 ± 0.20
	Renzi	95.21 ± 0.21	94.70 ± 0.13	92.23 ± 0.20
	Travaglio	98.99 ± 0.01	97.69 ± 0.04	96.06 ± 0.04
	Macklemore	95.26 ± 0.30	94.78 ± 0.19	92.24 ± 0.27
	Vasco	97.60 ± 0.28	95.88 ± 0.23	93.53 ± 0.37

Conclusioni

Gli esperimenti hanno mostrato che la possibilità di inserire il watermark su di un Social Network dipende fortemente da due fattori, ovvero la natura del testo e il Social in questione.
In particolare su piattaforme dove non ci sono grosse limitazioni sull'utilizzo dei caratteri per inserire il watermark sono sufficienti testi di dimensione ridotta, e quindi è concreta anche la possibilità di veicolare lo stesso messaggio su piattaforme diverse, a patto che queste non aggiungano nuovi limiti.
Se invece il Social limita l'utilizzo di alcuni caratteri saranno necessari testi di lunghezza molto più consistente e si ridurrà così il numero di piattaforme su cui il messaggio possa essere ripubblicato.
Questo metodo risulta quindi sicuramente più efficace su alcune piattaforme e per alcuni tipi di testo, rispetto che per altri.
Se si volesse utilizzare un algoritmo unico, avendo la possibilità di veicolare il watermark attraverso diverse piattaforme escluderei quindi Telegram e Facebook , poiché sono le due che impongono maggiori limiti sull'utilizzo dei caratteri per inserire il watermark, le limitazioni imposte da Twitter, escludendo la lunghezza del messaggio, si sono rivelate invece accettabili, poiché questa piattaforma esclude solo due caratteri confusable, che peraltro non vengono utilizzati molto frequentemente.
Per quanto riguarda invece l'invisibilità del metodo sono stati raccolti risultati molto positivi, poiché nonostante anche in questo caso si è visto che c'è una componente di dipendenza dalla natura del testo utilizzato e quindi anche dalla piattaforma, i post a cui è stato applicato il watermark sono

sempre risultati più simili agli originali di quelli a cui è stato applicato uno shift orizzontale di due pixel , e nella maggior parte dei casi anche di quelli a cui è stato applicato uno shift di un pixel.
Una componente molto influente in questi risultati si è rivelata essere la lunghezza del messaggio stesso, che ancora una volta viene influenzata da alcune piattaforme, poiché non ammettono testi superiori ad una certa lunghezza, mentre il tipo di sostituzione utilizzata dall'algoritmo (di caratteri o spazi) non ha causato grosse differenze come si sarebbe potuto pensare.

Appendice

Script 1 Codice utilizzato per calcolare il confronto MSE [24]

```
def mse(imageA, imageB):   #minimo 0 massimo
                           #195075 in diversita'
        image1 = cv2.imread(imageA)
        image2 = cv2.imread(imageB)
        if image1.size == image2.size:
                err = np.sum((image1.astype("float") -
                      image2.astype("float")) ** 2)
                err /= float(image1.shape[0]
                       * image1.shape[1])
                similarity= (195075.0- err)/195075.0
        else:
                similarity=0
        return similarity
```

Questa funzione prende in input due immagini, e ne fa il confronto, restituendo un valore tra 0 e 195075 che ne indichi la similarità.

Script 2 Codice utilizzato per inserire lo shift di 1 e 2 pixel nei post

```
def shift(img, name, n_px ):
        im=Image.open(img)
        ar=np.asarray(im) #vedo l'immagine come una
                          # matrice di pixel

        l,h = im.size
        white=list(ar[h-1][l-1])   # considero come bianco il pixel
                                   # nell'angolo in basso a destra
        #print white
        start=end=0     #indicano inizio e fine della riga che voglio shiftare
        n=end # conta le righe
        while 1:
                if n> h:
                                #siamo alla fine, salva tutto e
                                shifted=Image.fromarray(ar)
                                shifted.save(name+".png")
                                return
                for line in ar[end:h]:
                        found=False
                        for px in line:
                                if list(px)!= white: #riga non completamente bianca
                                        print "assegno start"
                                        start=n
                                        found=True
                                        break
                        if found:
                                break
                        n+=1
                if found:
                        for line in ar[start:h]:
                                found=False
                                if all(list(px)==white for px in line):
                                        print "assegno end"
                                        end=n
                                        found=True
                                        break
                                n+=1
                print start, end #indici di inizio e fine della riga da shiftare
                if found:
                        print "shifto"
                        lr=randint(0,9) #mi serve per decidere se sciftare a dx o a sx
                        #print "LR", lr
                        col =0   #seconda colonno tutta bianca
                        if lr%2 == 0:
                                direction= range(1,l-1)
```

```
else :
        direction= reversed(range(1,l-1))
for i in direction: #devo scandire tutte le colonne
        is_white=True
        is_white2=True
        for line in ar[start:end+1]:
                if list(line[i])!= white:
                        is_white=False
                        break # se non e' tutta bianca
        if is_white: #se la colonna e' tutta bianca
                          si esegue lo shift
                for line in ar[start:end+1]:
                        if lr%2 ==0:        #da sx a dx
                                if list(line[i+1])!= white:
                                        is_white2=False
                                        break
                        else:
                                if list(line[i-1])!= white:
                                        is_white2=False
                                        break
        if is_white and is_white2:# 2 colonne bianche
                col=i
                break

if col<l: #se ho trovato lo spazio devo shiftare
        print "col", col
        sec=tuple(map(tuple, ar))
        print start, end
        for j in range(start-1, end+1):
                if lr%2 == 0:
                        direction= range(col,l)
                else :
                        direction= reversed(range(0,col))
                sec=list(sec)

                for z in direction:
                        sec[j]=list(sec[j])
                        if lr%2 == 0:
                                if z < l-n_px:
                                        sec[j][z]=sec[j][z+n_px]
                                        sec[j][z]=tuple(sec[j][z])
                                else:
                                        sec[j][z]=white
                                        sec[j]=tuple(sec[j])
                        else:
                                if z >= 0+n_px:
                                        #print z
```

```
                                        #print z-n_px
                                        sec[j][z]=sec[j][z-n_px]
                                        sec[j][z]=tuple(sec[j][z])
                                else:

                                        sec[j][z]=white
                                        sec[j]=tuple(sec[j])
                        sec=tuple(sec)

                ar=np.asarray(sec, np.uint8)
```

In questo caso l'immagine viene considerata come una matrice di pixel, per prima cosa quindi si deve cercare una riga di testo; questa inizierà alla prima riga di pixel dove compare un pixel non bianco (start), e finirà alla prima riga di pixel, successiva a quella appena a start, completamente bianca (end).
Determinata una riga sarà necessario cercare al suo interno due colonne di pixel bianche, in modo da poter effettuare lo shift, una volta trovate anche queste e stabilito se questo dovrà essere a dx o sx si sposta tutta la riga e si riempe lo spazio vuoto all'altra estremità di bianco.
Queste operazioni vengono ripetute ginché non è stata analizzata l'intera immagine.

Script 3 Codice utilizzato per inserire il watermark [23] in versione originale.

```
import siphash

confusables = [(u'\u217D',u'\u0063'),
                                (u'\u216D',u'\u0043'),
                                (u'\u217E',u'\u0064'),
                                (u'\u216E',u'\u0044'),
                                (u'\u2170',u'\u0069'),
                                (u'\u217C',u'\u006C'),
                                (u'\u216C',u'\u004C'),
                                (u'\u216F',u'\u004D'),
                                (u'\u2174',u'\u0076'),
                                (u'\u2164',u'\u0056'),
                                (u'\u2179',u'\u0078'),
                                (u'\u2169',u'\u0058'),
                                (u'\u0458',u'\u006A'),
                                (u'\u212A',u'\u004B'),
                                (u'\u037E',u'\u003B'),
                                (u'\u2010',u'\u002D')
                                ]

conversion_table = {x[1]:x[0] for x in confusables}
conversion_table_space = {'000': u'\u0020',
                                                        '001' : u'\u2005',
                                                        '010' : u'\u2007',
                                                        '011' : u'\u202F',
                                                        '100' : u'\u2009',
                                                        '101' : u'\u205F',
                                                        '110' : u'\u2004',
                                                        '111' : u'\u2008'}

inverted_table_space = {conversion_table_space[key]:key
                        for key in conversion_table_space}

inverted_table = {x[0]:x[1] for x in confusables}
for key in conversion_table_space:
        inverted_table[conversion_table_space[key]] = u'\u0020'


originals = [x[1] for x in confusables]
specials = [x[0] for x in confusables]
spaces = [conversion_table_space[key] for key in conversion_table_space]



def mark(s,k):
```

```
        while len(k) < 16:
                k += '0'
        k = k[:16]
        s=s.encode('utf-8')
        #print "Original text to be watermarked:\s%s" % s
        hash = siphash.SipHash_2_4(k, s).hash()
        payload = [x for x in bin(hash)[2:]]
        while len(payload) < 64:
                payload.insert(0,'0')
        #print "Watermark:\n%s" % payload
        ws = u''
        print "--------------------Encoding watermark:-----------------"
        for char in s.decode('utf-8'):
                if payload:
                        if char in originals:
                                bit = payload.pop(0)
                                if bit == '1':
                                        char = conversion_table[char]

                                #print "%s:%s|" % (bit,char)

                        elif char == ' ':
                                bits = payload.pop(0)
                                if payload:
                                        bits += payload.pop(0)
                                        if payload:
                                                bits += payload.pop(0)
                                        else:
                                                bits += '0'
                                else:
                                        bits += '00'


                                char = conversion_table_space[bits]
                                #print "%s:%s|" % (bits,char)

                ws += char
        if payload:
                return False
        else:
                return ws


def verify(s,k):
        print "------------------Verifying Watermark-------------------"
        while len(k) < 16:
                k += '0'
```

```
k = k[:16]
payload = []
for char in s:
        if len(payload)<64:
                if char in originals:
                        payload.append('0')
                elif char in specials:
                        payload.append('1')
                elif char == ' ':
                        payload.extend(['0','0','0'])
                elif char in spaces:
                        bits = inverted_table_space[char]
                        payload.extend([x for x in bits])
payload = payload[:64]
#print payload
source = ''.join([inverted_table[x] if x in inverted_table else x for x in s ])
source = source.encode('utf-8')
#print source
hash = siphash.SipHash_2_4(k, source).hash()
payload1 = [x for x in bin(hash)[2:]]
while len(payload1) < 64:
        payload1.insert(0,'0')
#print payload1
a = ''.join(payload1)
b = ''.join(payload)
if a==b:
        return True
else:
        return False
```

Script 4 Modifiche per inserire il watermark su FB. Le Due matrici, confusable e space presenti nello script precedente sono state sostituite dalla matrice che segue

```
confusables = [(u'\u217D',u'\u0063'),
                                    (u'\u216D',u'\u0043'),
                                    (u'\u217E',u'\u0064'),
                                    (u'\u216E',u'\u0044'),
                                    (u'\u2170',u'\u0069'),
                                    (u'\u217C',u'\u006C'),
                                    (u'\u216C',u'\u004C'),
                                    (u'\u216F',u'\u004D'),
                                    (u'\u2174',u'\u0076'),
                                    (u'\u2164',u'\u0056'),
                                    (u'\u2179',u'\u0078'),
                                    (u'\u2169',u'\u0058'),
                                    (u'\u0458',u'\u006A'),
                                    (u'\u212A',u'\u004B'),
                                    (u'\u037E',u'\u003B'),
                                    (u'\u2010',u'\u002D'),
                                    (u'\u205F', u'\u0020') #unico spazio alternativo che accetta fb
```

Similmente a questo caso, per Telegram è semplicemente stata rimossa l'intera matrice di conversione degli spazi, senza aggiungere l'unico spazio alternativo nella prima; mentre per Twitter sono state mantenute entrambe, eliminando però le codifiche per i caratteri K e ; .

Bibliografia

[1] *Annoucing the standard for secure hash standard*, http://www.nymphomath.ch/crypto/moderne/fip180-1.html, Accessed: 2016.

[2] *Byte order mark*, https://en.wikipedia.org/wiki/Byte_order_mark, Accessed: 2016.

[3] *Comparing data sets using statistical analysis in excel*, http://www.aspfree.com/c/a/braindump/comparing-data-sets-using-statistical-analysis-in-excel/, Accessed: 2016.

[4] *Funzione di ripartizione*, https://it.wikipedia.org/wiki/Funzione_di_ripartizione, Accessed: 2016.

[5] *Intervallo di confidenza*, https://it.wikipedia.org/wiki/Intervallo_di_confidenza, Accessed: 2016.

[6] *Test t*, https://it.wikipedia.org/wiki/Test_t, Accessed: 2016.

[7] Tomio Amano and Daigo Misaki, *A feature calibration method for watermarking of document images*, Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on, IEEE, 1999, pp. 91–94.

[8] Mikhail J Atallah, Victor Raskin, Michael Crogan, Christian Hempelmann, Florian Kerschbaum, Dina Mohamed, and Sanket Naik, *Natural*

language watermarking: Design, analysis, and a proof-of-concept implementation, International Workshop on Information Hiding, Springer, 2001, pp. 185–200.

[9] Jean-Philippe Aumasson and Daniel J Bernstein, *Siphash: a fast short-input prf*, International Conference on Cryptology in India, Springer, 2012, pp. 489–508.

[10] Anoop K Bhattacharjya and Hakan Ancin, *Data embedding in text for a copier system*, Image Processing, 1999. ICIP 99. Proceedings. 1999 International Conference on, vol. 2, IEEE, 1999, pp. 245–249.

[11] Jack T Brassil, Steven Low, Nicholas F. Maxemchuk, and Lawrence O'Gorman, *Electronic marking and identification techniques to discourage document copying*, IEEE Journal on Selected Areas in Communications **13** (1995), no. 8, 1495–1504.

[12] James H Burrows, *Secure hash standard*, Tech. report, DTIC Document, 1995.

[13] Ding Huang and Hong Yan, *Interword distance changes represented by sine waves for watermarking text images*, IEEE Transactions on Circuits and Systems for Video Technology **11** (2001), no. 12, 1237–1245.

[14] Stefan Katzenbeisser and Fabien Petitcolas, *Information hiding techniques for steganography and digital watermarking*, Artech house, 2000.

[15] Manmeet Kaur and Kamna Mahajan, *An existential review on text watermarking techniques*, International Journal of Computer Applications **120** (2015), no. 18.

[16] Young-Won Kim, Kyung-Ae Moon, and Il-Seok Oh, *A text watermarking algorithm based on word classification and inter-word space statistics.*, ICDAR, 2003, pp. 775–779.

[17] Young-Won Kim and Il-Seok Oh, *Watermarking text document images using edge direction histograms*, Pattern Recognition Letters **25** (2004), no. 11, 1243–1251.

[18] Steven H Low, Nicholas F Maxemchuk, and Aleta M Lapone, *Document identification for copyright protection using centroid detection*, IEEE Transactions on Communications **46** (1998), no. 3, 372–383.

[19] Hasan Mesut Meral, Bülent Sankur, A Sumru Özsoy, Tunga Güngör, and Emre Sevinç, *Natural language watermarking via morphosyntactic alterations*, Computer Speech & Language **23** (2009), no. 1, 107–125.

[20] Nighat Mir, *Copyright for web content using invisible text watermarking*, Computers in Human Behavior **30** (2014), 648–653.

[21] Lip Yee Por, KokSheik Wong, and Kok Onn Chee, *Unispach: A text-based data hiding method using unicode space characters*, Journal of Systems and Software **85** (2012), no. 5, 1075–1082.

[22] D. Giardino Prof. A. De Santis, *Digital watermarking*, Slide universitarie, 2008, pp. 8–9.

[23] Stefano Giovanni Rizzo, Flavio Bertini, and Danilo Montesi, *Content-preserving text watermarking through unicode homoglyph substitution*, Proceedings of the 20th International Database Engineering & Applications Symposium, ACM, 2016, pp. 97–104.

[24] A. Rosenbrock, *How to: Python compare two images*, `http://www.pyimagesearch.com/2014/09/15/python-compare-two-images/`, Accessed: 2016.

[25] Mercan Topkara, Umut Topkara, and Mikhail J Atallah, *Information hiding through errors: a confusing approach*, Proceedings of the SPIE International Conference on Security, Steganography, and Watermarking of Multimedia Contents, vol. 29, 2007.

[26] Umut Topkara, Mercan Topkara, and Mikhail J Atallah, *The hiding virtues of ambiguity: quantifiably resilient watermarking of natural language text through synonym substitutions*, Proceedings of the 8th workshop on Multimedia and security, ACM, 2006, pp. 164–174.

[27] Olga Vybornova and Benoit Macq, *Natural language watermarking and robust hashing based on presuppositional analysis*, 2007 IEEE International Conference on Information Reuse and Integration, IEEE, 2007, pp. 177–182.

Ringraziamenti

Desidero ringraziare tutti coloro che mi hanno aiutato nella realizzazione di questa Tesi:
il relatore Danilo Montesi;
il correlatore Stefano Giovanni Rizzo;
ed il correlatore Flavio Bertini.

Le persone citate in questa pagina hanno svolto un ruolo fondamentale nella stesura della tesi, ma desidero precisare che ogni errore o imprecisione è imputabile soltanto a me.