Università degli studi di Torino

SCUOLA DI SCIENZE DELLA NATURA

Corso di Laurea Triennale in Informatica

Tesi di Laurea Triennale

Generazione di attacchi squatting tramite modelli generativi

Relatore:
Chiar.mo Prof.
Drago Idilio

Candidato:
Falchi Lorenzo

Anno Accademico 2022/2023

A me,
alla mia famiglia,
a chi c'è.

Abstract

Lo “squatting” e gli “attacchi omografici” rappresentano due sfide significative nel panorama della sicurezza informatica. Lo “squatting” coinvolge la registrazione di nomi di dominio o indirizzi email simili a quelli legittimi al fine di ingannare gli utenti e condurre attività malevole. Gli “attacchi omografici”, d’altra parte, sfruttano la somiglianza visiva tra caratteri di diverse lingue o alfabeti per creare indirizzi web o email che sembrano autentici ma conducono a contenuti fraudolenti. Questi fenomeni richiedono un’analisi approfondita e strategie di difesa efficaci per proteggere utenti e organizzazioni dagli inganni online. Obbiettivo di questa tesi è la generazione di parole che aumentino l’efficacia dell’attacco, utilizzando modelli generativi. In particolare sono stati analizzati due approcci: il primo basato su una CNN attention-based, il secondo su una rete neurale basata su un trasformer. La prima rete è formata da 4 strati convoluzionali e da un modulo di attenzione (CBAM). Riceve in input un’immagine e la classifica come reale o falsa. La seconda rete invece riceve in input un’immagine, (precedentemente pre processata per il riconoscimento ottico dei caratteri) che passa attraverso un encoder per l’elaborazione delle immagini e un decoder per la generazione di testo. L’output è una sequenza di testo che rappresenta il risultato del riconoscimento ottico dei caratteri (OCR) dall’immagine di input. Ognuno dei due approcci è stato singolarmente testato e validato e i dati in output non sono stati generalmente quelli attesi.

Dichiaro di essere responsabile del contenuto dell'elaborato che presento al fine del conseguimento del titolo, di non avere plagiato in tutto o in parte il lavoro prodotto da altri e di aver citato le fonti originali in modo congruente alle normative vigenti in materia di plagio e di diritto d'autore. Sono inoltre consapevole che nel caso la mia dichiarazione risultasse mendace, potrei incorrere nelle sanzioni previste dalla legge e la mia ammissione alla prova finale potrebbe essere negata.

Indice

Capitolo 1

Introduzione

Nell'era digitale in cui viviamo, la sicurezza informatica è diventata una questione cruciale, poiché il mondo online è sempre più suscettibile a una vasta gamma di minacce e attacchi. Tra le sfide più significative che caratterizzano questo panorama abbiamo lo **squatting** e, più in particolare, gli **attacchi omografici**. Questi fenomeni rappresentano minacce pervasive e ingegnose, capaci di mettere a repentaglio la privacy e la sicurezza degli utenti, oltre che la fiducia nelle transazioni e comunicazioni online.

Lo **squatting** coinvolge la registrazione di nomi di dominio o indirizzi email che sono straordinariamente simili, se non praticamente identici, a quelli orginali, al fine di trarre in inganno gli utenti e condurre attività malevole. D'altro canto, gli **attacchi omografici** sfruttano la somiglianza visiva tra caratteri di diverse lingue o alfabeti per creare indirizzi web o email che sembrano legittimi, ma che in realtà conducono a contenuti fraudolenti.

La crescente sofisticazione di tali tecniche richiede un'analisi approfondita e strategie di difesa altamente efficaci per proteggere utenti e organizzazioni dalle frodi online. È necessario comprendere appieno la portata di queste minacce e sviluppare metodi avanzati per mitigare i rischi associati.

Per affrontare queste sfide, la tesi si basa su un'approccio che sfrutta l'apprendimento automatico. Il linguaggio di programmazione principale per l'acquisizione e l'elaborazione dei dati utilizzao è python, mentre lo strumento DNSTwist è stato impiegato per generare i nomi di dominio falsi utilizzati nello **squatting**. Ho analizzato e testato 2 modelli e mi sono posto due quesiti:

- il modello basato su ocr viene ingannato da scritture tipicamente usate per phishing? Come si comporta con immagini di domini fake?
- il modello basato sulla cnn riesce ad individuare in maniera efficace i domini fake?

Nel corso di questa tesi, cercherò di rispondere ai quesiti posti, dando anche una corposa descrizione di contesto e strumenti utilizzati.

1.1 Lavori correlati

Inizialmente mi sono concentrato sulla rete neurale TrOCR [1], la quale è stata utilizzata sui domini contrassegnati per estrarre il testo dalle immagini fornite in input. Si basa su un trasformer, una tecnologia di apprendimento profondo altamente avanzata, che riceve in input un'immagine, la sottopone a un processo di riconoscimento ottico dei caratteri (ocr) e traduce i caratteri dell'immagine in testo. Cardini di questo processo sono un encoder, per l'elaborazione delle immagini, e un decoder per la generazione del testo. Viene così generata una sequenza di caratteri che rappresenta il risultato del ocr.

Successivamente, il focus si è spostato sulla classificazione di immagini (contenenti testo) ricevute in input da una rete neurale convoluzionale (cnn) attention-based (GlyphNet [2]). Le immagini in input, sono generate a partire dai domini ricavati da TrOCR. La rete dimostrava una promettente capacità di riconoscere e classificare immagini di domini fake e autentici. É formata da quattro strati convoluzionali e un modulo di attenzione (CBAM) per migliorare la sua capacità di individuare le differenze tra immagini reali e contraffatte.

1.2 Struttura della tesi

Obiettivo principale di questa tesi è esplorare l'uso di modelli generativi per generare parole o sequenze di caratteri che possano aumentare l'efficacia degli attacchi brevemente descritti precendentemente.

Entrambi i modelli sono stati testati e validati; tuttavia, i risultati ottenuti finora hanno dimostrato non rispettare completamente le aspettative. Questo ha portato a ulteriori interrogativi sulla complessità di queste sfide e sottolinea la necessità di ulteriori ricerche per sviluppare soluzioni efficaci.

La prima parte della tesi tratta un'ampia introuduzione su attacchi analizzati e modelli testati, mentre la seconda parte contiene una descrizione accurata dei due modelli e l'analisi dei risultati ottenuti.

Capitolo 2

Overview

2.1 Squatting e Phishing

Il **cybersquatting** [3], noto anche come **domain squatting**, prevede la registrazione di un nome di dominio che assomiglia a un'organizzazione o a una persona nota, senza avere però l'autorizzazione. Il registrante compra il dominio in malafede, tipicamente con l'obiettivo di fare un profitto dalla buona volontà della persona o dell'organizzazione o di causare un danno reputazionale.

I **cybersquatter** mettono in atto diverse tecniche per raggiungere i propri obiettivi economici. Per esempio, si può aspettare che la registrazione di un dominio web arrivi alla sua naturale scadenza per poi acquisirlo e rivenderlo al suo originale proprietario, che ha tutto l'interesse di rientrarne in possesso.

Ci sono inoltre diversi script che permettono la generazione in pochi secondi di un enorme numero di domini malevoli con lo stesso aspetto di domini reali.

Tipicamente collegato allo squatting è il concetto di **phishing**. Il phishing è un tipo di attacco informatico basato, appunto, sul cybersqatting. L'obiettivo è quello di indirizzare gli utenti verso siti web ingannevoli, falsi, con il fine di sottrargli dati sensibili e informazioni riservate. Normalmente il phishing avviene tramite SMS oppure, ancora più frequentemente tramite posta elettronica. Attraverso quest'ultima, il malcapitato riceve un'email piuttosto verosimile, accompagnata dal link ad un sito. L'utente viene indotto in errore perchè, se è vero che il link può essere più o meno diverso dal link reale, il sito a cui si viene indirizzati ha un'interfaccia grafica che è del tutto identica al sito reale.

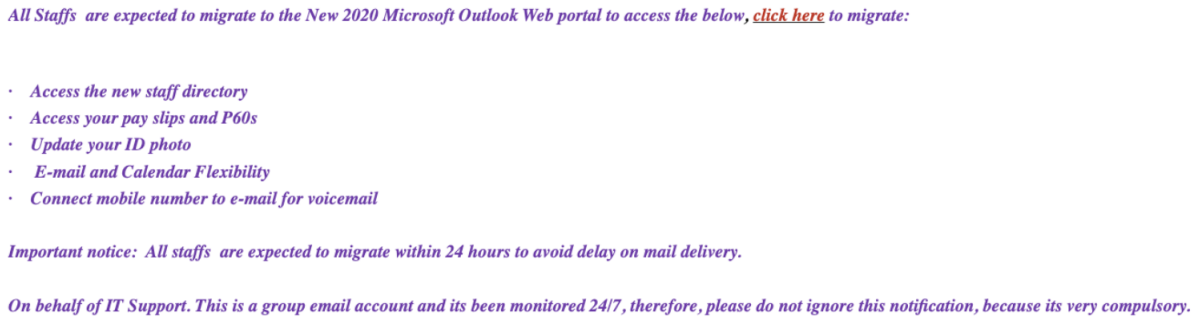

Figura 2.1: Esempio di contenuto di un'email fake

Una parte molto importante del phishing è la preparazione dell'url da inserire nell'eventuale email o SMS. Il link viene creato ad arte seguendo particolari tecniche e sfruttando quelle che sono le criticità più diffuse nell'uso di internet da parte della maggioranza della popolazione. In questo tipo di attacchi infatti è fondamentale la fase di preparazione. Nel caso in cui la vittima sia scelta accuratamente, la preparazione prevede lo studio di aspetti quali le abitudini della vittima e gli strumenti che utilizza di più. Se invece l'obbiettivo è un attacco di massa, per esempio l'invio simultaneo di migliaia di email, si sfruttano le criticità menzionate prima, tra tutte, gli errori di battitura. Questo tipo di attacco prende il nome di **Typosquatting**.

Figura 2.2: Esempio di dominio reale.

Figura 2.3: Esempio di dominio malevolo.

2.2 Tipi di attacchi

Nel seguito, illustrerò alcuni degli attacchi più importanti e maggiormente utilizzati.

Typosquatting. Il **typosquatting**[4] è una variante del cybersquatting e del phishing che sfrutta gli errori di battitura più comuni commessi dagli utenti durante la navigazione sul web. Questo tipo di attacco può manifestarsi in diverse forme, ciascuna mirata a sfruttare una specifica fonte di errore da parte dell'utente. Tra gli errori più sfruttati, troviamo:

- **errori di spelling dovuti a errori di battitura**;
- **riformulazione del nome di dominio**;
- **TLD differente**.

Errori di spelling dovuti a errori di battitura: in questa forma di typosquatting, gli utenti commettono errori di battitura specifici durante la digitazione degli indirizzi web. **esmepio.com** o **esempoi.com** rappresentano varianti di errore di battitura di **esempio.com**. Questa tattica sfrutta la similitudine tra i due nomi di dominio per ingannare gli utenti.

Riformulazione del nome di dominio (per esempio al plurale): questa tattica coinvolge la registrazione di nomi di dominio simili a quelli dei siti web legittimi, ma con piccole variazioni intenzionali, come la riformulazione del nome al plurale. Ad esempio, **esempi.com** è una riformulazione del nome di dominio "esempio.com." Gli utenti che cercano di accedere al sito web originale potrebbero non notare immediatamente la sottile differenza e finire per atterrare sul sito falso. Anche questa tattica, come la precedente, sfrutta la similitudine tra i due nomi di dominio.

TLD differente: Il Top-Level Domain (TLD) è la parte finale di un dominio, come **.com**, **.org**, **.net** ecc. Nel typosquatting che coinvolge TLD differenti, i truffatori registrano un dominio simile a uno legittimo, ma con un TLD diverso. Ad esempio, potrebbero registrare **esempio.org** invece di **esempio.com**. Gli utenti che inseriscono l'indirizzo web con il TLD errato possono finire su un sito fraudolento che potrebbe cercare di ingannarli o rubare informazioni. Questo tipo di typosquatting sfrutta la familiarità degli utenti con un determinato dominio per indirizzarli verso un contesto diverso.

In generale per prevenire questi tipi di attacchi, è importante prestare molta attenzione agli indirizzi web digitati e verificare sempre che il sito web sia quello corretto prima di inserire informazioni sensibili.

Attacchi omografici. Gli **attacchi omografici** rappresentano un altro sottotipo di attacco legato al typosquatting e al cybersquatting, ma sono ancora più sofisticati e subdoli. Questi attacchi sfruttano le somiglianze tra caratteri per creare nomi di dominio che sembrano identici a quelli legittimi, ma che sono in realtà diversi.

L'idea è di sfruttare il fatto che lingue e alfabeti diversi condividano caratteri visivamente simili, se non identici (**caratteri omografici**). Ad esempio, in alcuni casi, caratteri di due alfabeti diversi possono sembrare uguali ma rappresentare suoni diversi. Gli attaccanti sfruttano questa somiglianza per registrare domini identici a quelli legittimi ma che in realtà utilizzano, appunto, caratteri omografici.

Fake "apple.com"			Real "apple.com"		
Glyph	**Unicode Name**	**Unicode Hex**	**Glyph**	**Unicode Name**	**Unicode Hex**
a	Cyrillic small letter A	U+0430	a	Latin small letter A	U+0061
p	Cyrillic small letter Er	U+0440	p	Latin small letter P	U+0070
l	Cyrillic small letter Palochka	U+04CF	l	Latin small letter L	U+006C
e	Cyrillic small letter Ie	U+0435	e	Latin small letter E	U+0065

Tabella 2.1: esempio di omografi [5]

Dalla tabella 2.1, si vede chiaramente che i due domini (entrambi **apple.com**), almeno all'occhio umano, risultano identici. Andando poi in dettaglio, notiamo che in realtà sono composti da lettere completamente diverse, addirittura appartenenti ad alfabeti diversi (cirillico e latino). Questo è un tipico esempio di attacco omografico.

Con l'introduzione e il rapido sviluppo di modelli di intelligenza artificiale, in particolare modelli generativi, il fenomeno degli attacchi omografici e del typosquatting diventa ancora più subdolo ed efficace. Questi modelli possono essere utilizzati per generare nomi di dominio come quelli che abbiamo visto, per scopi fraudolenti.

La sfida principale nel contrastare questi attacchi generati da modelli AI, sta nell'identificare le minacce in modo automatico e nella creazione di sistemi di sicurezza avanzati che siano in grado di rilevare domini fraudolenti, anche quando sono estremamente simili a quelli legittimi.

2.3 Modelli generativi

I **modelli generativi** [6] utilizzano algoritmi di intelligenza artificiale che mirano a **generare dati e contenuti originali** simili a quelli prodotti da esseri umani. A differenza dei **modelli discriminanti** che si concentrano sulla **classificazione o sulla predizione**, i modelli generativi si dedicano alla creazione di nuovi esempi che si adattano a uno specifico set di dati tramite l'apprendimento della struttura e le caratteristiche dei dati di addestramento. L'obiettivo è rendere i nuovi esempi generati quanto più autentici e coerenti con il valore di input.

Ci sono diversi tipi di modelli generativi, uno dei più noti è quello dei generative adversarial networks (GAN). Un GAN è composto da due reti neurali: un **generatore** e un **discriminatore**. Il generatore è responsabile della creazione di nuovi esempi, mentre il discriminatore cerca di distinguere tra gli esempi generati e quelli reali. Queste due componenti si allenano a vicenda in un processo di competizione e collaborazione, migliorando gradualmente le loro abilità fino a raggiungere un equilibrio.

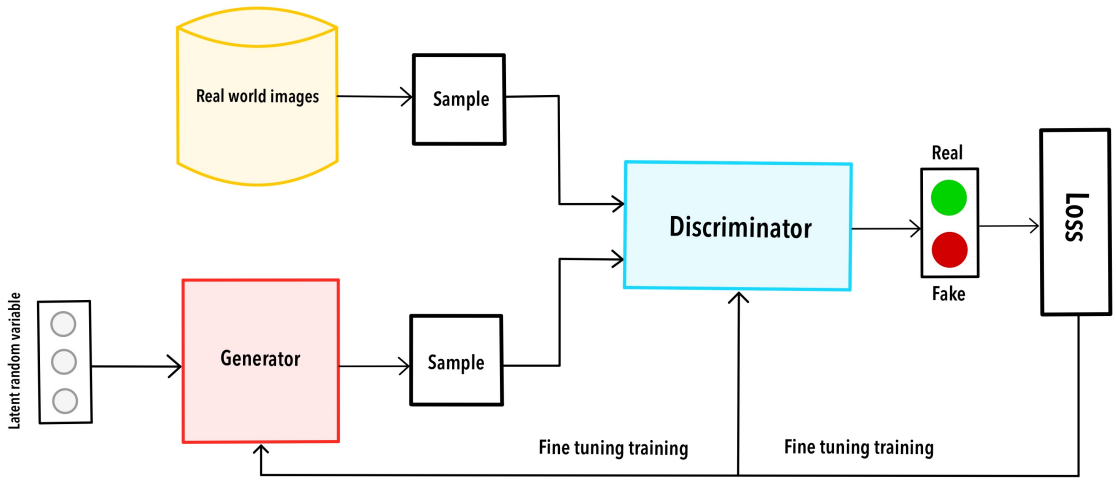

Figura 2.4: schema di un modello generativo di tipo GAN

Un altro tipo di modello generativo molto diffuso è quello degli **autoencoder**. Gli autoencoder sono reti neurali composte da più livelli e l'aspetto che definisce un autoencoder è che i livelli di input contengono esattamente la stessa quantità di informazioni del livello di output. Il motivo per cui il livello di input e il livello di output hanno lo stesso numero di unità è che un autoencoder mira a replicare i dati di input per emettere poi una copia dei dati dopo averli analizzati e ricostruiti senza supervisione. L'architettura di questa rete contiene due parti principali:

- **encoder**, una rete feedforward densamente connessa. Lo scopo degli strati di codifica è quello di prendere i dati di input e comprimerli in una rappresentazione di spazio latente, generando una nuova rappresentazione dei dati che ha una dimensionalità ridotta.
- **decoder**, responsabile di prendere i dati compressi e riconvertirli in una rappresentazione con le stesse dimensioni dei dati originali e inalterati. La conversione viene eseguita con la rappresentazione dello spazio latente creata dall'encoder.

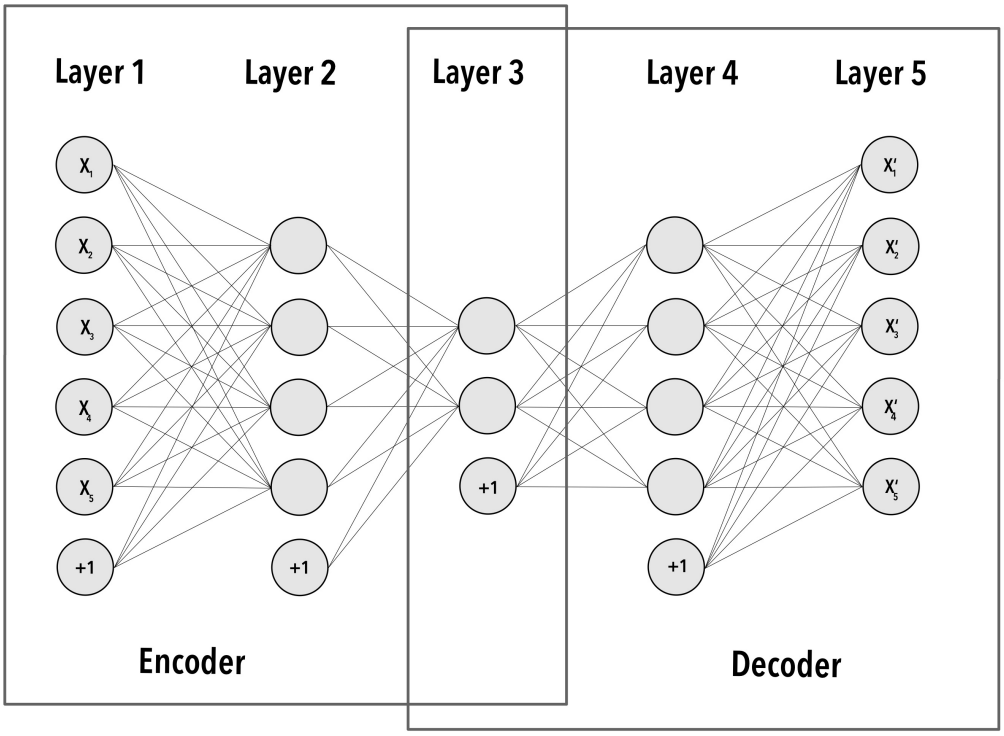

Figura 2.5: schema di un modello di tipo autoencoder

Esempio di un modello generativo di questo tipo è **GPT-3**, alla base dell'ormai usatissimo **ChatGPT**, ma sopratutto uno dei due modelli trattati in questa tesi, ovvero **TrOCR**.

2.3.1 OCR

ocr è l'acronimo di "Optical Character Recognition," (Riconoscimento Ottico dei Caratteri). Il processo di ocr coinvolge l'acquisizione di un'immagine contenente testo e l'interpretazione di caratteri e parole all'interno di quell'immagine, al fine di creare un testo digitale.

ocr opera attraverso diverse fasi:

- acquisizione dell'immagine;
- elaborazione;
- riconoscimento dei caratteri;
- creazione di un documento digitale.

La parte centrale del processo coinvolge il riconoscimento dei caratteri all'interno dell'immagine. Qui, il software ocr analizza l'immagine pixel per pixel alla ricerca di forme e strutture che corrispondono a lettere, numeri e simboli. I caratteri riconosciuti vengono quindi convertiti in testo digitale.

2.4 Reti neurali convoluzionali (CNN)

Le reti neurali convoluzionali [7], comunemente abbreviate con cnn (Convolutional Neural Networks), rappresentano un pilastro dell'elaborazione delle immagini e del riconoscimento dei modelli visivi. Queste reti neurali specializzate sono progettate per l'estrazione e l'analisi delle caratteristiche all'interno di immagini e dati visivi complessi.

L'intuizione chiave delle cnn risiede nell'applicazione di filtri convoluzionali alle diverse regioni dell'immagine, consentendo di catturare dettagli e strutture rilevanti. Questi filtri svolgono il ruolo di strumenti di **convoluzione** che scorrono sull'immagine per individuare pattern, bordi, texture e altri elementi di interesse. Questo approccio consente alle cnn di apprendere gerarchie di feature sempre più complesse e di rappresentare efficacemente i dati visivi in modo stratificato.

Le cnn sono state utilizzate in una vasta gamma di applicazioni, tra cui il riconoscimento di oggetti, la classificazione di immagini, la segmentazione delle immagini e l'analisi di testo contenuto in immagini. La loro capacità di apprendere autonomamente dalle immagini stesse, evitando la necessità di estrazione manuale di feature, le rende uno strumento potente nell'ambito dell'elaborazione delle immagini. Oltre alle applicazioni tradizionali, le cnn hanno trovato impiego anche in settori come l'analisi medica e la guida autonoma.

Glyphnet è un esempio di cnn ed è stata utilizzata in questo lavoro per la classificazione di immagini contenenti un dominio. Il testo delle immagini passate però, è stato generato direttamente dal testo generato da TrOCR.

2.4.1 CNN Attention-Based

Le reti neurali convoluzionali attention-based, o cnn attention-based, costituiscono un'estensione avanzata delle reti neurali convoluzionali (cnn) progettate per l'elaborazione di immagini e l'analisi di modelli visivi. Queste reti combinano l'abilità delle cnn di catturare le caratteristiche salienti nelle immagini con meccanismi di attenzione che consentono loro di concentrarsi su regioni specifiche. L'uso di questa attenzione selettiva è ispirato al modo in cui le persone elaborano le informazioni visive, evidenziando elementi rilevanti e ignorando il resto. La capacità di selezionare attentamente le parti chiave di un'immagine per l'analisi ha dimostrato di migliorare significativamente le prestazioni in compiti complessi.

Capitolo 3

Applicazione di ocr per estrazione di testo

In questo capitolo, rispondo alla prima delle due domande che mi sono posto nell'introduzione, ovvero "**il modello basato su ocr viene ingannato da scritture tipicamente usate per phising? Come si comporta con immagini di domini fake?**". Segue una rapida panoramica del modello e l'analisi dei risultati.

3.1 TrOCR

TrOCR è composto da un encoder di immagini basato su Transformer e un decoder di testo basato su Transformer autoregressivo per eseguire il riconoscimento ottico dei caratteri tramite ocr.

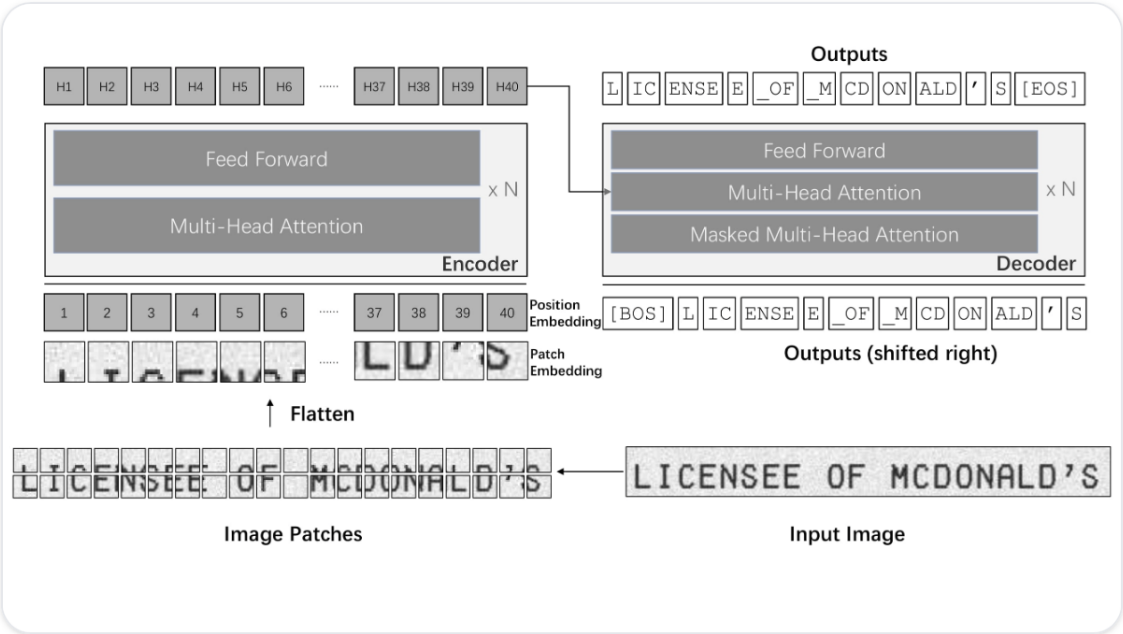

Figura 3.1: Architettura di TrOCR [1]

Come mostra la figura 3.1, TrOCR riceve in input un'immagine generata a partire da testo e restituisce in output il testo che è riuscito a generare dall'immagine fornita. L'immagine in input viene prima preprocessata dall'encoder: viene ridimensionata per ridurre la complessità computazionale, quindi suddivisa in una serie di patch, ciascuna delle quali rappresenta una piccola regione dell'immagine originale. Le patch vengono a loro volta trasformate in una serie di vettori di caratteristiche che rappresentano le informazioni contenute nelle patch stesse. Questi vettori di caratteristiche sono poi combinati utilizzando un meccanismo di attenzione per creare un embedding dell'immagine. L'embedding è una rappresentazione vettoriale che può essere utilizzata dal decoder per generare il testo riconosciuto. Arrivato a questo punto, il decoder utilizza l'embedding per generare una **sequenza di token** (unità di testo di base, come lettere o parole). **Il token con la probabilità più alta viene selezionato come output del decoder**.

3.2 Metodologie e dataset

La libreria principale utilizzata per la creazione di TrOCR è PyTorch. Il lavoro si è concentrato prima e dopo l'esecuzione vera e propria del modello. Infatti, per fornire al modello dati consistenti, con l'ausilio della libreria PIL ho scritto una funzione python che riceve del testo e, a partire da questo, crea l'immagine corrispondente. Dopodichè ho cercato di **ripulire l'output**, poichè tra gli errori ricorrenti ma trascurabili del modello, ci sono:

- cancellazione di "." in corrispondenza del dominio;
- sostituzione di lettere maiuscole con minuscole (e viceversa);
- aggiunta di spazi bianchi.

Le immagini sono state create utilizzando diverse modalità (es. modifica della grandezza del font, utilizzo di font diversi), per cercare di testare il modello a 360 gradi, su quanti più dati possibili. Il dataset è composto da circa 100 domini reali e da 1000 fake (circa 10 fake per ogni reale). I fake sono stati generati utilizzando DNStwist e con diverse modalità:

- sostituzione di caratteri con numeri (es. o con 0);
- sostituzione di caratteri latini con caratteri di altri alfabeti;
- uso di lettere simili tra di loro (es. "w" e "vv").

Real	**Fake**
youtube.com	youtubę.com
wikipedia.org	wikipadia.org
milannews.it	milannevvs.it
live.com	lıve.com

Tabella 3.1: Esempi di domini reali e fake

Si porta l'attenzione in particolare sull'ultima riga della tabella: infatti nonostante le parole appaiano sostanzialmente identiche, il dominio fake ha in realtà una lettera azera (anche usata nell'alfabeto turco) al posto della "i" del dominio reale. Questi sono i casi più interessanti in cui, come vedremo più avanti, il modello ha sbagliato sostanzialmente sempre.

3.3 Risultati

Il modello è stato testato su una parte dei domini reali e su una parte di domini fake (il 70% di entrambi i dataset). Per misurare quanto il modello fosse preciso nell'estrarre il testo dall'immagine, ho utilizzato la **edit distance**, misurando la distanza (in termini di operazioni su stringhe) tra il testo generato e il testo fornito in input (sotto forma di immagine). Ho anche calcolato la media di queste misurazioni durante il test su tutto il dataset.

Ho infine riportato i dati su una CDF, così definita:

- sull'asse x è rappresentata l'accuratezza delle parole (**in termini di edit distance**), ordinato in maniera crescente da sinistra a destra;
- sull'asse y è rappresentata la probabilità cumulativa associata all'accuratezza delle parole. L'asse delle y varia da 0 a 1 e indica la probabilità cumulativa di ottenere un'accuratezza pari o inferiore a un determinato valore sull'asse delle x.

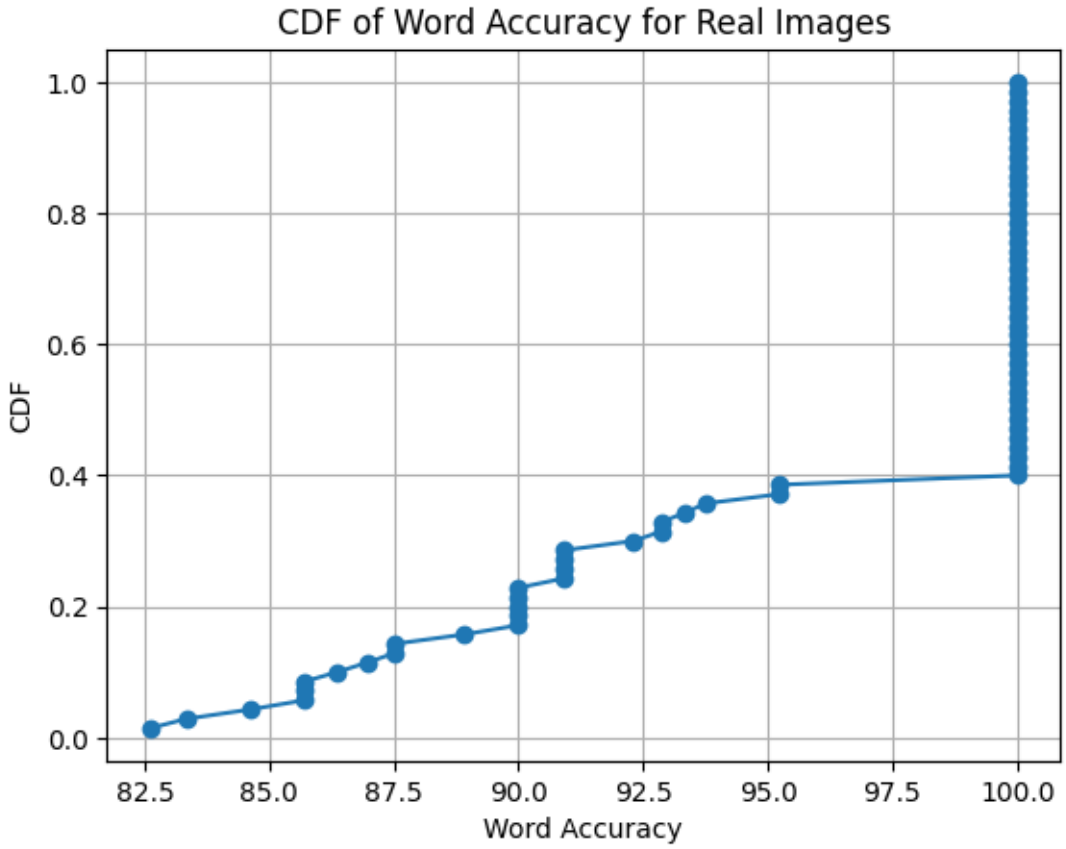

Figura 3.2: CDF Real images

L'analisi dei dati fornisce diversi spunti:

- il valore minimo di correttezza registrato è 82.5% mentre il massimo è 100%;
- 1 parola su 5 viene generata corretta per meno del 90% dei suoi caratteri;
- la curva è asimmetrica, spostata verso destra rispetto all'asse centrale;
- la media di correttezza delle parole è superiore al 96%;
- il valore più frequente è il 100%.

La situazione varia di molto quando cerco di ingannare il modello, ovvero quando tratto i domini fake. Qui, complici numeri simili a caratteri, caratteri sostituiti da caratteri molto simili ma di altri alfabeti, il modello ha un comportamento ancora generalmente buono (in termini di accuratezza media), ma, a livello microscopico ci sono delle considerazioni da fare.

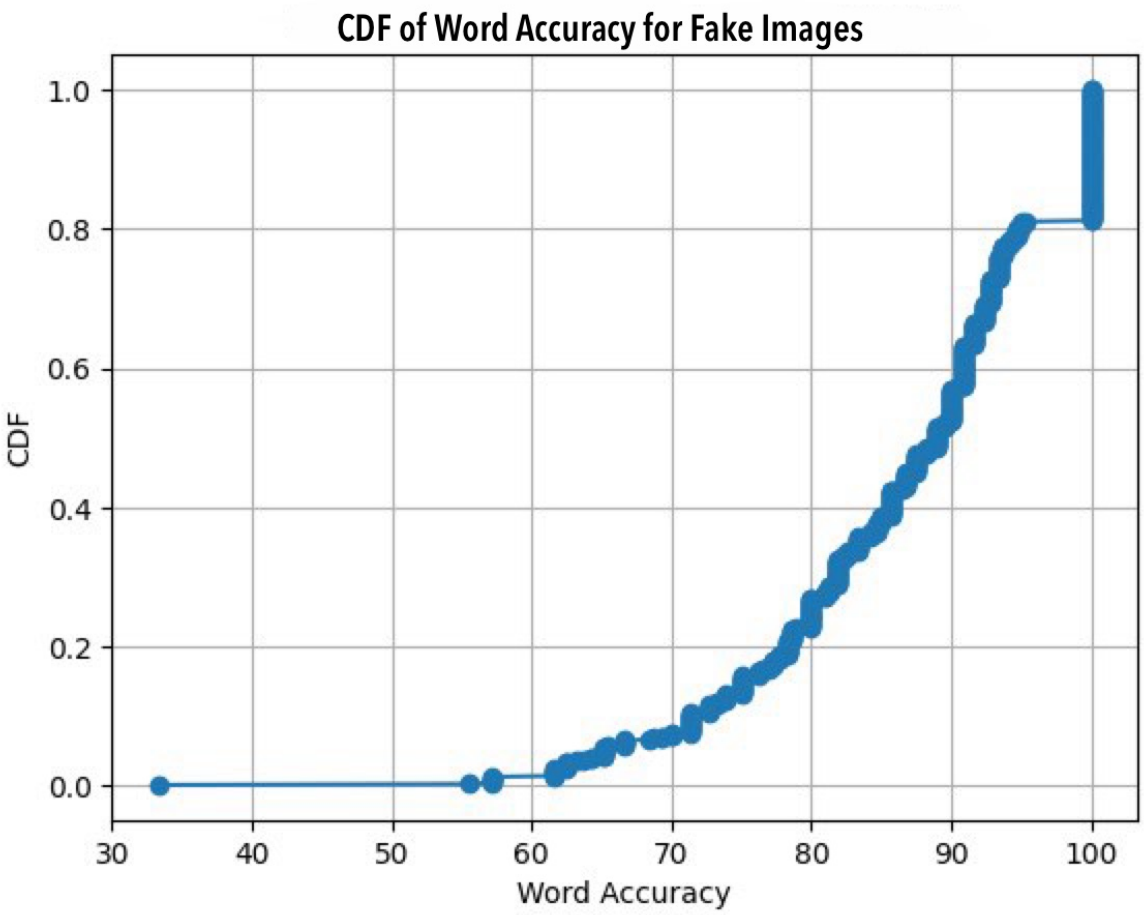

Figura 3.3: accuratezza Fake

Riporto alcuni dati, come fatto con i domini reali:

- il valore minimo di correttezza registrato è 30% mentre il massimo è 100%;
- più della metà delle parole viene generata corretta per meno del 90% dei suoi caratteri;
- la curva è comunque asimmetrica, spostata verso destra rispetto all'asse centrale ma molto meno rispetto alla cdf dei domini reali;
- la media di correttezza delle parole è di poco superiore al 85%, 11 punti percentuali in meno rispetto alla misurazione precedente;
- i valori di precisione più frequenti si trovano, per la stragrande maggioranza dei casi, nell'intervallo (60,90).

Riguardo la edit distance: il minimo valore registrato è 0 (corrispondente ad una parola generata perfettamente) mentre il massimo è 9. Segue una tabella con percentuali, su un totale di 700 esempi considerati.

Valore edit distance	**Percentuale di presenza**
0	$\sim 8\%$ (57 su 700)
1	$\sim 25\%$ (178 su 700)
2	$\sim 25\%$ (177 su 700)
3	15% (105 su 700)
4	$\sim 12\%$ (83 su 700)
5	$\sim 5\%$ (39 su 700)
6	5% (35 su 700)
7	$< 1\%$ (6 su 700)
8	$< 1\%$ (4 su 700)
9	$< 1\%$ (6 su 700)

Tabella 3.2: Valori di edit distance con relativa percentuale

Dalla tabella 3.2 si vede che solo l'8% delle parole è generato perfettamente, che la metà delle parole generate ha tra gli 1 e i 2 caratteri diversi da quelli della parola originale e che, cumulativamente, circa il 40% delle parole ha più di 2 errori.

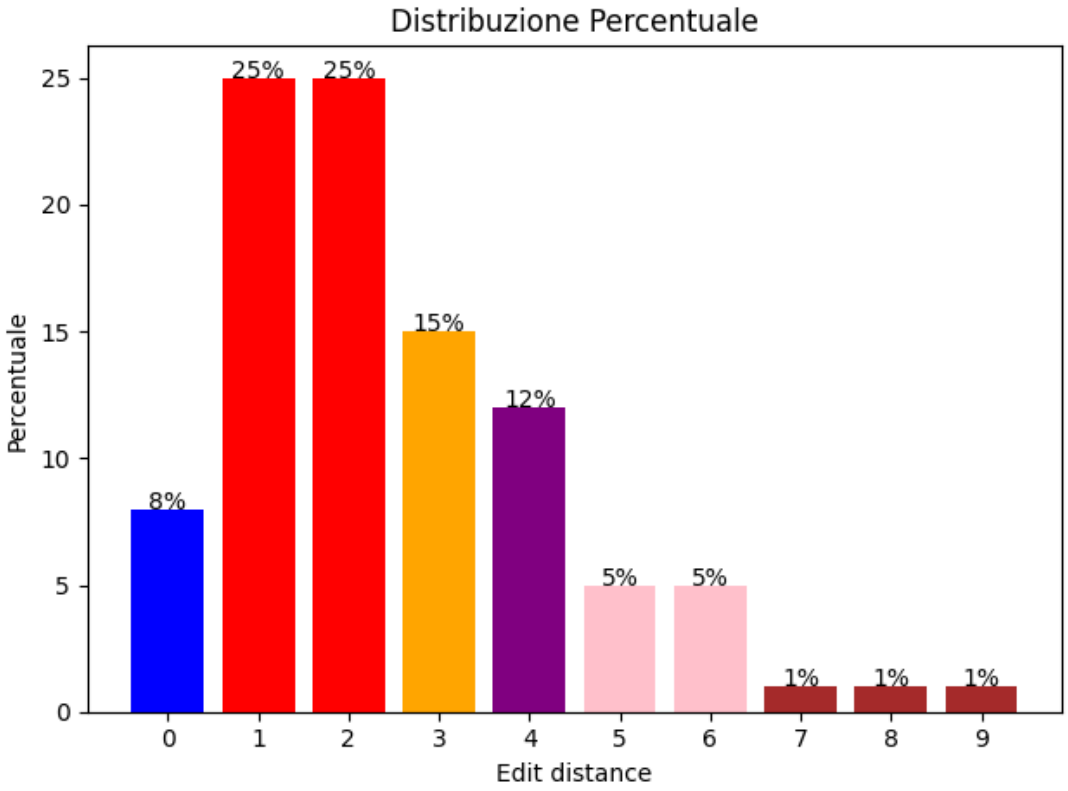

Figura 3.4: Distribuzione percentuale dei valori di edit distance

Le sostituzioni effettuate di più dal modello:

Carattere/numero reale	**Carattere/numero sostituto**	**Percentuale di sostituzione**
a	g,e	∼ 4% (27 su 754)
b	o,h,d	∼ 6% (14 su 233)
c	a,e,o,n	∼ 18% (128 su 699)
d	b	∼ 4% (6 su 153)
e	o	∼ 4% (20 su 498)
g	9	∼ 7% (15 su 207)
i	l,j	∼ 3% (24 su 764)
l	1	25% (90 su 358)
m	n	9% (54 su 600)
n	h	5% (31 su 317)
r	i,o	5% (28 su 567)
u	n	∼ 3% (5 su 182)
v	u	∼ 32% (25 su 79)
w	r	∼ 3% (2 su 67)
z	e	7% (3 su 42)
1	l,i,h	60% (111 su 185)
0	o,e	90% (134 su 148)

Tabella 3.3: Sostituzioni effettuate dal modello

In aggiunta a questi dati, il 100% dei caratteri accentati o con appendici di qualche tipo, converge negli stessi caratteri ma "ripuliti" (es.'ç' diventa 'c') Il carattere più volte confuso con altri caratteri risulta essere "**c**", mentre la coppie carattere-numero più volte scambiate sono "**(1,l)**", "**(1,i)**" e "**(0,o)**". Sono state registrate globalmente 5 situazioni:

- sostituzione di caratteri dello stesso alfabeto:
  - (originale: **fcimtcrmews.it**, generato: **faintcimers.it**);
- sostituzione di caratteri di alfabeti diversi:
  - (originale: **lıve.com**, generato: **live.com**);
- sostituzione di numeri con caratteri:
  - (originale: **aranzu1ia.it**, generato: **aranzulia.it**);
- sostituzioni di una coppia di caratteri con un singolo carattere:
  - (originale: **f0rrnulapassion.it**, generato: **formulapassion.it**);
- aggiunta di uno o più caratteri:
  - (originale: **div.is**, generato: **divis.is**).

Quest'ultima casistica risulta essere la più rara ma viene riportata comunque, per completezza. In aggiunta alle situazioni riportate, menziono il fatto che qualsiasi tipo di accento viene ignorato rendendo, di fatto, molto più semplice forzare il modello a generare testo diverso da quello ricevuto in input sotto forma di immagine. In generale possiamo dire che il modello sembra essere abbastanza facilmente ingannabile in quanto non risulta particolarmente sensibile a micro varizioni sui caratteri. Sembra inoltre tendere a "ripulire" il testo che riceve in input, riportandolo ad una forma che conosce.

Capitolo 4

Classificazione dei domini generati

Questo capitolo risponde alla seconda delle due domande presenti nell'introduzione, ovvero "**il modello basato sulla cnn riesce ad individuare in maniera efficace i domini fake?**". Come fatto per TrOCR, vediamo prima una rapida panoramica sul modello.

4.1 GlyphNet

GlyphNet è un dataset e un metodo per rilevare attacchi omografici. Il metodo proposto si basa su una rete neurale convoluzionale attention-based, composta da 4 layer di convoluzione, ognuno dei quali seguito da un layer di max-pooling. A sua volta, ogni blocco convoluzionale è seguito da un CBAM (convolutional block attention module). Il modello è sviluppato in keras e anche questa volta, come per TrOCR, riceve in input un'immagine. L'output però è un'etichetta "true" o "false".

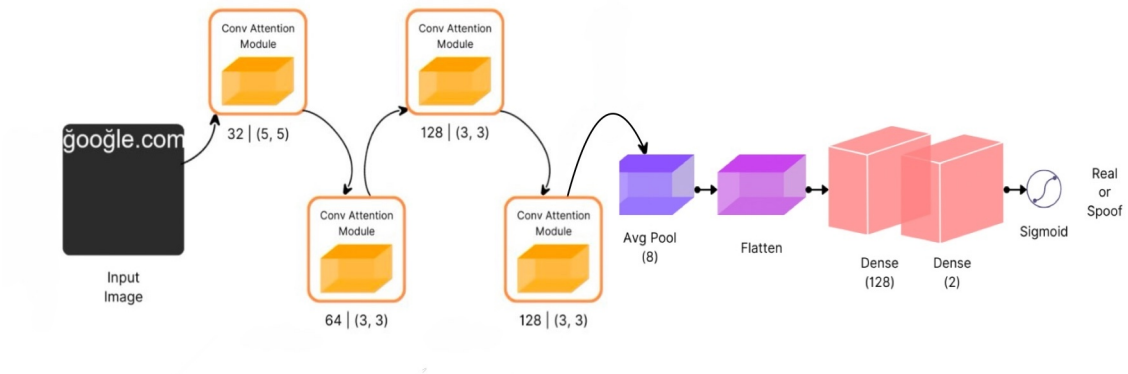

Figura 4.1: schema di GlyphNet

I 4 layer convoluzionali utilizzano dei filtri per rilevare le caratteristiche visive a diverse scale. I filtri vengono addestrati su un set di dati di immagini, in modo da imparare a rilevare le caratteristiche che sono più importanti per la classificazione delle immagini. Gli strati di max-pooling riducono le dimensioni dell'immagine, mantenendo le informazioni più importanti.

Il modulo CBAM è così composto:

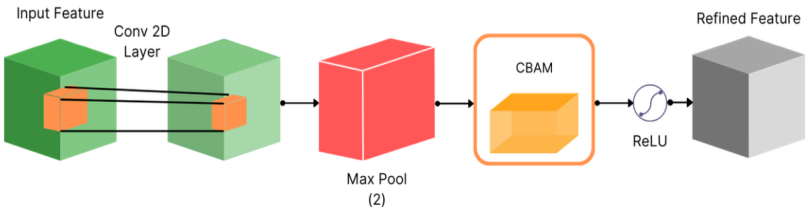

Figura 4.2: Schema modulo CBAM

questo modulo di attenzione aiuta la rete neurale convoluzionale a concentrarsi sulle parti più importanti dell'immagine di un dominio, come i caratteri, e a sopprimere le parti meno importanti, come lo sfondo. Questo aiuta la rete neurale a distinguere tra domini omografi e domini legittimi in modo più accurato.

4.2 Dataset

Il modello è preaddestrato a partire da un dataset composto da 2 milioni di domini reali. Di questi, 1 milione è stato usato per creare domini con 1 carattere omografico e 1 milione per creare domini con due caratteri omografici. Infine il dataset è stato diviso con un rateo di 70:20:10 per, rispettivamente, addestramento, validazione e testing. A partire da questo, il modello è stato testato sui domini che sono stati "ripuliti" da TrOCR.

4.3 Risultati

Sui 700 domini, generati da TrOCR a partire da domini fake, circa 1 su 6 riesce ad "ingannare" il classificatore, classificandosi come "reale": infatti dei 700 iniziali, 583 vengono correttamente classificati come falsi e 107 come reali. Sui domini reali invece, 52 vengono correttamente classificati come reali e 18 come falsi.

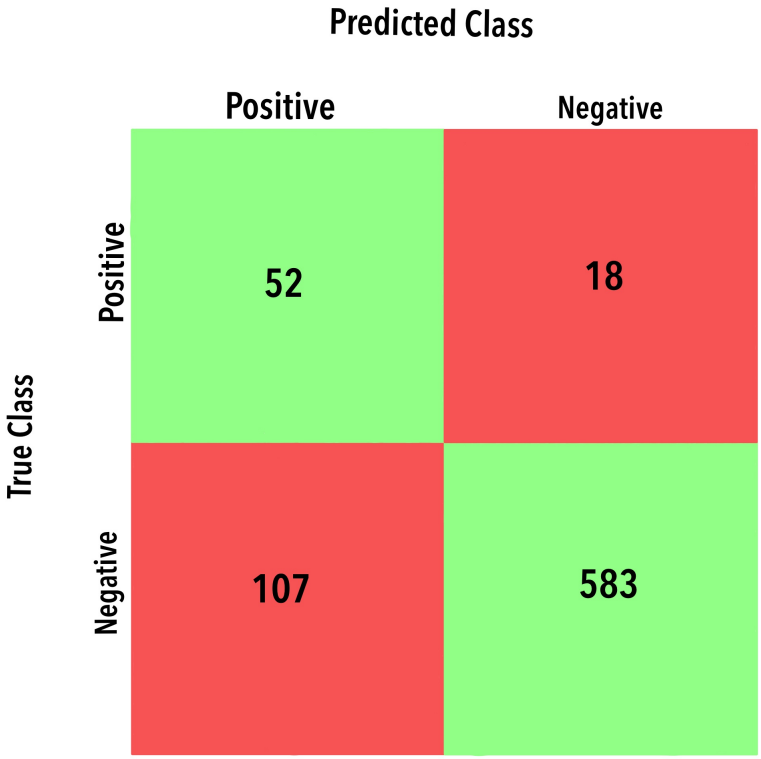

Figura 4.3: Matrice di confusione dei risultati del modello

Segue l'elenco dei domini falsi, classificati come reali:

dirette.it2	assa'it	rarpage.it2	postcilt.
annazonit	fandor.com	rampagcitt	annsaint
tanboncom	pospect.	banoeier.p.	iastampait
Immobiliereit	corticreit.It	cormiere.it	ilpost.co
Virginia.it0	postext.	rampagcitt	postext
mediasetit	amsa'it	rankier.pl2	rarpage.it2
postext	arnsait	Gazzetta.it	gazzettait
cormiere.it	wikipediaO.org	lastarripait	gazzettait
assa'it	posteit	rampageit	ebai.it2
libertist	ebey.it3	liberoOut	ansa'it
ebai.it2	tuttoMercercatoweb.com	bookinocom	ipostit
liberolist	NCSA-Ourrowepportalcom	postcitt	idpost.it0
rampagcitt	liberopit	lastampait	ilposto.it
corriere.it	impessageroit	Gazzetta.it	correctorit
ligiornaie.it	NCSA-Ourrowepportalcom	poste.it	googlep.com
annazon.it	ammazon.it	poste.it	librettist
modiasciut	cortiere.it	liberotit	impressaggerOUT
gazzettait	mediasetit	lastampa'it	libeto.it2
itpost.It	NCSA-ourrowebportalcom	Wharksap-com	arsait
ding.com	ECAA.Eurovebportalcon	assa'it	vocathercom
lastampait	ECAA.Eurovebportalcon	impost.it0	lipost.it2
rampage.it2	gazzettait	gazzettait	oppostit
rannpage.it	googleacom	impostit	gazzettait
lastampait	rampage.it2	liberait	contorcit
anazomit	NCSA-Ourrowepportalcom	cormere.it	banjier.P.P
Gazzetta.it	lastampa.it2	cormiere.it	Whagtsap.com
lastampait	poste.it	ansa'it	twitter.com

Capitolo 5

Conclusioni e possibili sviluppi futuri

In questo capitolo finale, riassumerò le scoperte e i risultati ottenuti nei capitoli precedenti, discuterò le implicazioni e le limitazioni del lavoro e proporrò alcune direzioni per futuri sviluppi.

5.1 Conclusioni

In questo studio, ho affrontato l'importante questione della rilevazione dei domini fake utilizzati per scopi di phishing, concentrandomi sull'applicazione di un modello ocr (TrOCR) per il riconoscimento di testo in immagini di domini web.

Durante l'analisi dei risultati, ho constatato che TrOCR non è ancora abbastanza efficace nel riconoscere il testo all'interno di immagini di domini reali, ottenendo risultati ancora peggiori nel riconoscere testo di immagini di domini fake. Se il riconoscimento di domini fake è comprensibilmente difficoltoso, quello di domini reali poteva e doveva essere più preciso. Ciò porta a concludere che queste tecnologie, sebbene molto promettenti (TrOCR è stato allenato soprattutto con testo scritto a mano), siano ancora lontane dal poter essere considerate utili ed affidabili nel combattere le minacce informatiche.

Anche GlyphNet ha dimostrato una non elevatissima precisione nel rilevare la natura dei domini. Infatti sui 70 domini reali, 52 sono stati classificati come real e ben 18 come fake. Come riportato nei risultati, circa 1 dominio fake su 6 riesce ad ingannarlo e a farsi classificare come **reale**. Ovviamente il fatto che sbagli la classificazione sui domini reali è meno problematico rispetto all'errore sui fake, nonostante questo, ciò è indice del fatto che questo strumento non è ancora pronto ne affidabile per gli scopi che mi sono posto.

5.2 Possibili sviluppi futuri

Per quanto riguarda i futuri sviluppi, alcune delle direzioni di ricerca potenziali includono:

- **miglioramento di ocr**: questo potrebbe comportare l'implementazione di tecniche avanzate di riconoscimento del testo e l'addestramento su dataset più ampi e diversificati per migliorare il riconoscimento del testo in immagini di domini falsi;
- **esplorazione di nuovi tipi di domini fake**: bisognerebbe considerare altre varianti di domini fake e tattiche più sofisticate utilizzate dai truffatori online. L'obiettivo è sviluppare modelli sempre più precisi nel riconoscere i pattern degli attachi;
- **integrazione con altri metodi di sicurezza**: l'uso combinato di ocr e modelli di classificazione come GlyphNet può essere una strategia efficace per rilevare domini falsi. Una valida strada da percorrere potrebbe essere quella di combinare diversi tipi di modelli e esplorare ulteriori metodi e tecniche di sicurezza per migliorare la precisione complessiva.

In definitiva, questo lavoro getta le basi per ulteriori ricerche sulla sicurezza informatica e sulla protezione da minacce di phishing legate ai domini web. Il riconoscimento e la classificazione dei domini falsi sono aspetti critici della sicurezza cibernetica, e i risultati di questo lavoro forniscono un punto di partenza per future ricerche in questo campo.

Bibliografia

[1] M. Li, T. Lv, J. Chen, L. Cui, Y. Lu, D. Florencio, C. Zhang, Z. Li, and F. Wei, "Trocr: Transformer-based optical character recognition with pre-trained models," 2022.

[2] A. Gupta, L. S. Tomar, and R. Garg, "Glyphnet: Homoglyph domains dataset and detection using attention-based convolutional neural networks," 2023.

[3] A. Rubino, "Cybersquatting, cos'è, come difendersi," 2021.

[4] Wikipedia, "Typosquatting — wikipedia, l'enciclopedia libera," 2020. [Online; in data 23-ottobre-2023].

[5] J. Umawing, "Out of character: Homograph attacks explained," 2017.

[6] D. Entratici, "Modelli generativi: L'alchimia dell'intelligenza artificiale," 2023.

[7] M. Capurso, "Le architetture comuni di deep learning: le reti neurali convoluzionali (cnn)," 2023.