Università degli studi di Torino

**SCUOLA DI SCIENZE DELLA NATURA**

**Corso di Laurea Triennale in Informatica**


**Tesi di Laurea Triennale**

# Utilizzo di GANs per la generazione di "homoglyph cybersquatting"

**Relatore:**
**Chiar.mo Prof.**
**Drago Idilio**

**Candidato:**
**Cortosi Vittorio**

**Anno Accademico 2021/2022**

Ringrazio me stesso, ovviamente,
Idilio Drago (best relatore eva).
La mia famiglia

Agli amici:
Il grande BU per le lezioni di filosofia
Gianni per i pack opening
CROLE che mi ha insegnato ad usare WASD
Marco che mi ha insegnato ad essere Pepega
1298 per: elenco di cose
Sara per l'ansia
Andrea?

# Abstract

I siti Web di phishing di oggi sono in continua evoluzione per ingannare gli utenti. La maggior parte di questi siti utilizza l'attività illegale del cybersquatting e, appropriandosi di domini che somigliano a domini commerciali famosi, ingannano l'utente spingendolo ad inserire dati personali e lucrando su di essi. Il sistema da me sviluppato utilizza una classe di modelli per l'apprendimento automatico chiamate reti neurali generative avversarie, che permette in maniera proattiva di generare domini di squatting. In particolare, utilizzo reti convoluzionali generative avversarie, per generare (partendo da un dataset di immagini) i domini che possono essere abusati. È stato poi effettuato un caso di studio reale con domini esistenti e i risultati preliminari ottenuti sono promettenti.

*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

Con l'avvento di internet e quindi dell'era digitale, durante gli ultimi decenni si è dovuto far fronte a molte minacce che riguardavano il mondo informatico. I malintenzionati sfruttano vulnerabilità di tutti i tipi, come ad esempio vulnerabilità che coinvolgono un componente informatico oppure che coinvolgono vulnerabilità di un programma e attraverso quest'ultimo prendere il controllo della macchina. Negli ultimi anni però, c'è stato un incremento sostanziale delle truffe che hanno come obbiettivo quello di sfruttare la mancata attenzione da parte dell'utente finale al fine di mettere in atto la truffa. Questo tipo di truffe è conosciuto con il nome di Phishing.
Con questa pratica, appunto, il malintenzionato cerca di ingannare la vittima convincendola a fornire informazioni personali, dati finanziari o codici di accesso, fingendosi un ente affidabile o una persona famosa.
Per commettere la pratica del phishing, spesso ci si avvale di un'altro tipo di cybercrime denominato domain squatting: Il cybersquatting è l'atto da parte del malintenzionato di appropriarsi di nomi di dominio che somigliano ai marchi commerciali più famosi o persone famose per effettuare il phishing.
Cercherò in questo articolo di esporre un sistema in grado di evitare che malintenzionati possano utilizzare domini internet (in particolare domini cybersquatting) per effettuare phishing. Generare potenziali domini di squatting in modo proattivo ci permette di identificarli ancora prima che possano essere utilizzati da malintenzionati per attacchi di phishing. SquatGAN è il prototipo iniziale di un sistema in grado di produrre questo sistema di generazione di domini. I risultati preliminari sono soddisfacenti ma non completi: quasi la totalità di essi sono domini typo squatting, i domini homographic squatting non vengono identificati in quanto il sistema di riconoscimento ottico dell'immagine non è stato ancora adattato per questo tipo di risultati (nel capitolo 4 sarà spiegato più in dettaglio).

## 1.1 Lavoro correlato

In questo articolo cercherò di emulare il comportamento di PishGan[1] il quale genera potenziali domini di squatting in modo pro-attivo utilizzando l'apprendimento automatico. Ciò che caratterizza il lavoro effettuato per PhishGan è il fatto di aver utilizzato come modello di rete neurale, una U-Net. Queste reti sono le più popolari per effet-

tuare quella che viene chiamata "Image segmentation" (Segmentazione dell'immagine), il quale permette di partizionare l'immagine in regioni/segmenti per permettere una comprensione più semplice dell'immagine stessa (la figura 1.1 mostra un esempio di segmentazione con due immagini diverse da cui si evidenziano le caratteristiche principali). Queste reti neurali vengono spesso usate in campo biomedico (ad esempio per la previsione del legame di una struttura proteica) e sono strutturate da due parti chiamate Encoder e Decoder. Nelle reti U-Net (figura 1.2)[1], nel fronte di discesa, vengono apprese le features attraverso strati di convoluzione mentre nel fronte di salita (Decoder) vengono ricostruite le informazioni facendo convoluzione trasposta concatenando le features map nel corrispondente layer dell'Encoder. Il concetto alla base di PhishGan è proprio questo, quello che andrò a sviluppare invece, sarà sempre un software di generazione di domini di squatting, ma utilizzando reti neurali convoluzionali generative (DCGAN).

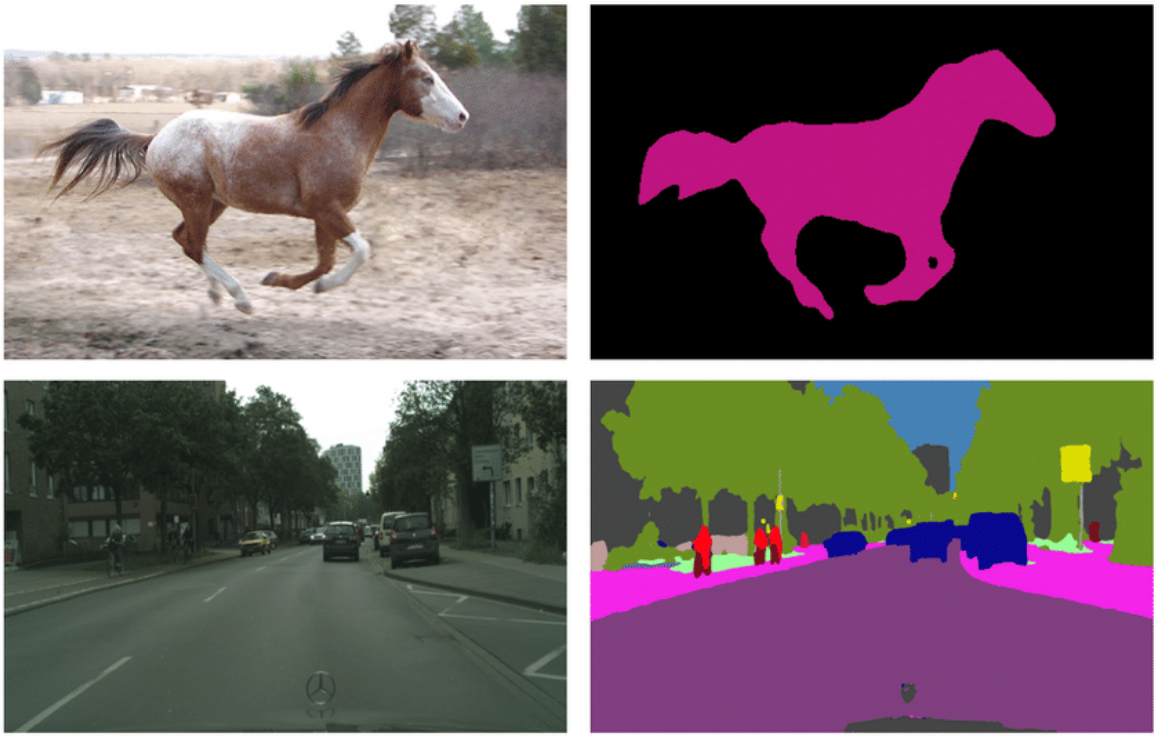

Figura 1.1: IEsempio di image segmentation

[1]https://www.frontiersin.org/articles/10.3389/fnins.2020.568614

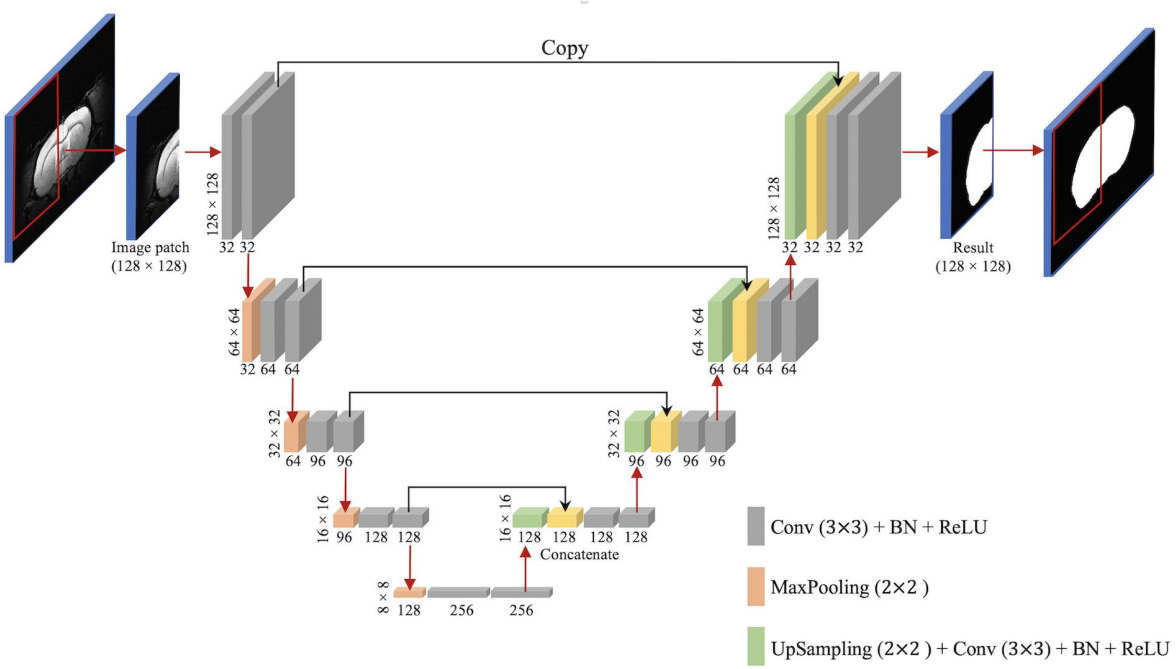


Figura 1.2: Esempio di architettura U-Net ('U' perchè la forma che assume la rete assomiglia alla lettera 'U')

## 1.2 Come è strutturato questo articolo

Nella prima parte dell'articolo illustrerò contesti in cui può essere applicato il phishing e come viene usato insieme al cybersquatting per ingannare gli utenti al fine di rubare dati sensibili. Illustrerò con che strumento sarebbe possibile risolvere questo problema (utilizzando reti neurali generative) per poi concludere illustrando alcune applicazioni per la cybersecurity.
Nella seconda parte, illustrerò l'architettura utilizzata in SquatGAN mentre nell'ultima parte del documento illustrerò i risultati preliminari e come sono arrivato ad ottenere quest'ultimi.

# Capitolo 2

# Background

## 2.1 Squatting & phishing

Il termine cybersquatting[2] si riferisce alla registrazione e all'uso non autorizzati di nomi di dominio Internet identici o simili a marchi, marchi di servizio, nomi di società o nomi personali. I registranti di cybersquatting ottengono e utilizzano il nome di dominio con l'intento in malafede di trarre profitto dalla buona volontà dell'effettivo proprietario del marchio. All'interno di questi nomi di dominio, è molto probabile incappare nel crimine comunemente chiamato Phishing. In sostanza il dominio "squattato" è il vettore che porta l'utente all'inganno, mentre il phishing è ciò che effettivamente raccoglie informazioni sensibili sugli utenti che cadono in inganno alla truffa. Detto questo, il phishing è quindi un tipo di truffa effettuata su Internet in cui un malintenzionato cerca di ottenere informazioni riservate (dati sensibili), o peggio, dati finanziari e codici di accesso, fingendosi un ente affidabile. Tutto questo può essere sfruttato in diversi scenari applicati alla vita quotidiana di un individuo.
Un esempio recente è quello di utilizzare gli SMS: in un contesto in cui le persone utilizzano molto gli acquisti online, e quindi utilizzano corrieri espressi per le consegne a domicilio, è facile pensare (per un malintenzionato) di inviare un messaggio che comunica ad esempio che il proprio pacco è stato smarrito e di cliccare su un determinato link per tracciarlo. La vittima ignara della truffa clicca sul link e inserisce dati sensibili o dati bancari cadendo nella trappola del malintenzionato.
Questo esempio può essere applicato identico in un contesto in cui si utilizzano Mail al posto di SMS. Qui il tutto è ancora più ingannevole: oltre al testo è possibile usare come vettori dell'attacco delle immagini: il malintenzionato utilizza delle immagini che ricordano molto pagine web di cui si fida la vittima ma in cui si nascondono hyperlink che indirizzano a pagine di phishing.
La figura 2.1 mostra un esempio di MAIL phishing mentre la figura 2.2 mostra un esempio di SMS Phishing


Figura 2.1: Mail Phisging

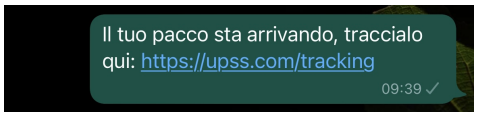


Figura 2.2: SMS Phishing

## 2.2 Tipi di squatting

Per effettuare il phihing, i malintenzionati utilizzano diverse tecniche di cybersquatting. Nella figura 2.2 ad esempio, viene sfruttato quello che viene chiamato "typo-squatting": viene acquistato un dominio che si basa su errori di battitura/digitazione. Come si può notare questo dirotta l'utente verso un sito differente da quello che voleva raggiungere. Ovviamente le minacce non si circoscrivono solamente a typosquatting.
In generale vengono definiti 5 tipi di cybersquatting:

- typo: come spiegato sopra, sono quei domini che si basano su errori di battitura e digitazione.
- homoglyph: questo tipo di squatting invece è uno dei più ingannevoli. Utilizza il fatto che molti caratteri tipografici sono simili tra loro. Nel dominio google, posso sostituire una 'o' con una lettera simile, ad esempio 'ŏ' (goŏgle.com). Questo tipi di dominio vengono chiamati IDN (internationalized domain name). Sono appunto nomi di dominio che contengono caratteri di alfabeto non latini (cinese, cirillico, greco, etc...). Questi nomi di dominio vengono salvati sui server DNS come stringhe ASCII utilizzando la trascrizione Punycode come visto nella Tabella 2.1.
- bit: questa forma di cybersquatting si basa su errori di bit-flip che si verificano durante il processo di richiesta DNS. Questi cambi di bit possono verificarsi a

Tabella 2.1

| IDN | Punycode |
|---|---|
| www.facebŏŏk.com | www.xn–facebk-tgba.com |
| www.googlè.com | www.xn–googl-8ra.com |

causa di fattori quali hardware difettoso o interferenze elettromagntiche.

- combo: il combosquatting aggiunge termini familiari negli URL che gli utenti incauti potrebbero non notare a prima vista. Questa tecnica si basa su analisi statistiche dei termini più utilizzati nelle pagine di enti affidabili, ad esempio, su instagram si usa molto la parola "story", "stories", "tags". Sapendo questo è possibile creare domini di squatting concatenando il dominio originario con uno dei termini più utilizzati: instagram-stories.com, instagram-tags.com, e così via.

- wrongTLD: Tutte le tecniche di squatting di cui sopra si concentrano sul nome di dominio ma ignorano il TLD. questa tecnica si riferisce a domini che cambiano il TLD ma mantengono il nome di dominio uguale. Per esempio, google.kekw appartiene alla categoria wrongTLD.

Sul web[1] è possibile reperire dati e grafici riguardanti il numero di domini abusati, quale tipologia viene più utilizzata e quale marchio è più preso di mira dai malintenzionati del caso. La figura 2.3, ad esempio mostra un grafico a torta che illustra le tipologie più utilizzate. Ovviamente chi abusa del cybersquatting per commettere crimini come phishing, mira ad utilizzare domini squatted di marchi che statisticamente gli utenti utilizzano più frequentemente o i più indispensabili (come le banche ad esempio). In figura 2.4 una vista dei marchi che vengono presi più di mira negli ultimi anni.

[1]https://www.catonetworks.com/blog/cato-networks-adds-protection-from-the-perils-of-cybersquatting/

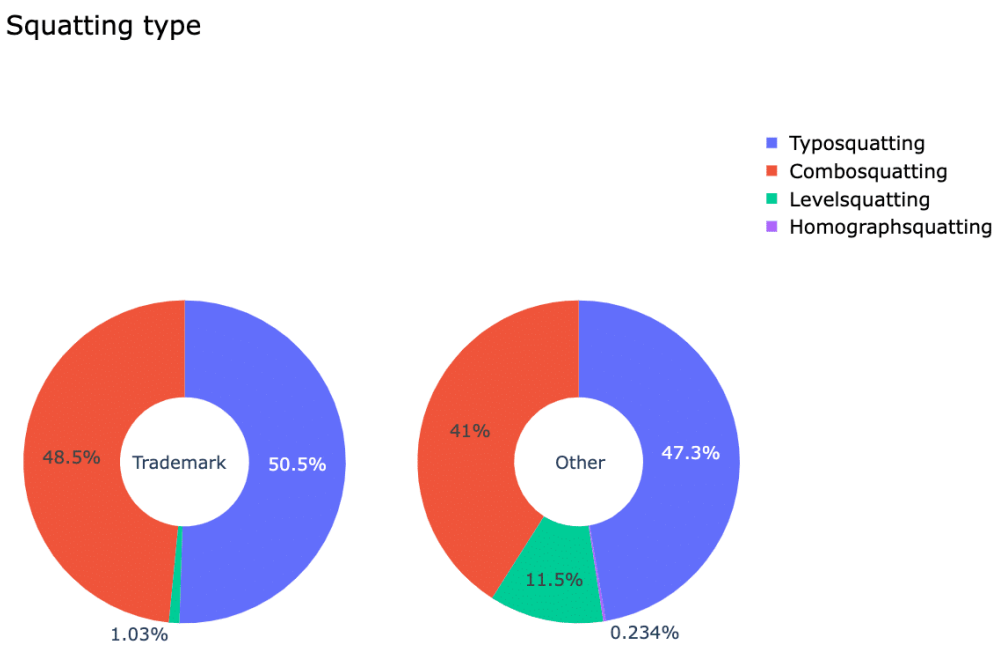


Figura 2.3: Due esempi delle percentuali di squatting più utilizzate. Il primo luogo le percentuali riguardanti i marchi più famosi, in secondo luogo tutti gli altri domini

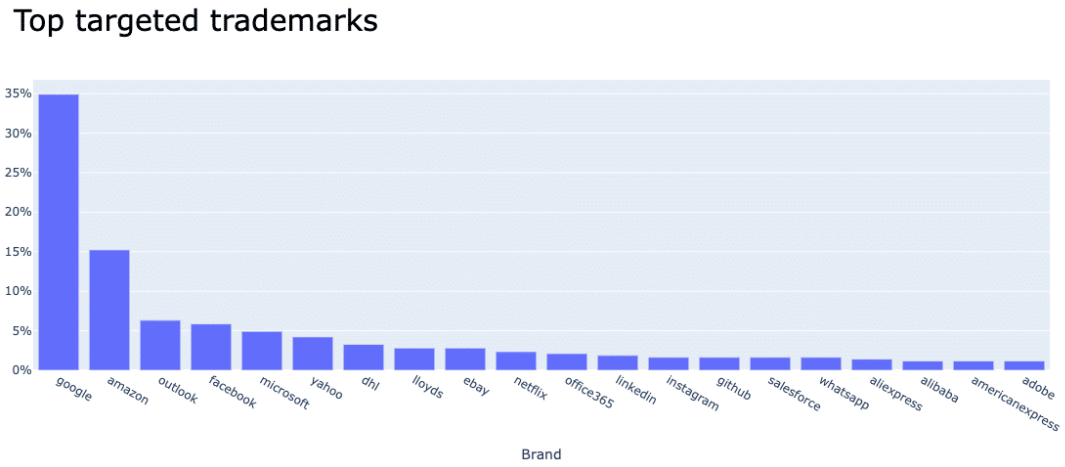


Figura 2.4: Dati riguardanti i marchi registrati più famosi, e quali di questi vengono più presi di mira dal cybersquatting

## 2.3 Come difendersi

Per quanto riguarda il phishing attraverso Mail, uno dei modi migliori per difendersi è quello di analizzare il sorgente e di conseguenza analizzare i collegamenti ipertestuali per verificarne la validità. Inoltre, molti antivirus moderni, introducono un modulo per la supervisione in tempo reale delle pagine web che visitiamo: i software antivirus


Figura 2.5: La Mail inserita da me e resa pubblica sul sito GitHub

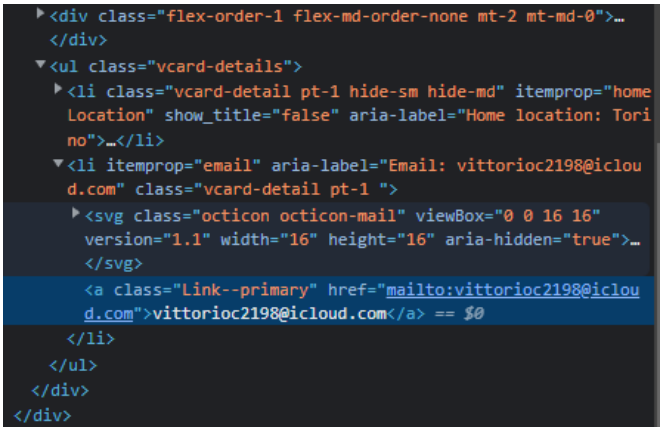


Figura 2.6: La Mail che compare nel sorgente della pagina

inglobano un modulo che confronta i domini che visitiamo con un database di URL che la compagnia ha etichettato come "malevoli"[3].
Un altro accorgimento che possiamo adottare è quello di analizzare la semantica e la veridicità delle informazioni[2.7] che, ad esempio, nel caso di un SMS o di una MAIL possono essere alterate facilmente: se si riceve una Mail o un SMS in cui viene indicato che un nostro pacco è stato smarrito, ma non abbiamo ordinato nessun prodotto, è facilmente riconducibile ad un caso di phishing, senza neanche ricondurci ad analizzare il sorgente della MAIL o analizzare l'URL in oggetto.
Inoltre, un modo per aggirare il problema alla radice, e quindi avere la minor probabilità possibile di ricevere Mail/SMS Phishing, è proprio quello di evitare di diffondere indirizzi di posta (o numeri di telefono) ad enti/pagine web che non riteniamo del tutto affiabili. Questo perchè nel momento in cui ci iscriviamo ad un forum/applicazione/website in generale, stiamo cedendo i nostri dati alla compagnia che ne gestisce il servizio. Questi dati, nel peggiore dei casi, verranno venduti per altri scopi ad altre compagnie o chissà a chi (evitiamo di iscriverci alle newsletter).
In ultimo, ma non meno importante, è ciò che viene chiamato Web Crawler: Un Web crawler (spesso abbreviato in crawler) è un bot Internet che naviga sistematicamente nel World Wide Web e che è tipicamente gestito dai motori di ricerca ai fini dell'indicizzazione del Web. Il problema nasce quando non sono i motori di ricerca che utilizzano il crawling, ma enti a scopo maligno. In sostanza analizzano le pagine del WWW (scrapping) ed estrapolano qualsiasi indirizzo Mail/Numeri di telefono inserendoli successivamente in un database che utilizzeranno per i lor scopi non puliti.
Un'esempio più che reale è ciò che è successo a me, dopo aver pubblicato il mio indirizzo Mail sul sito di GitHub (figura 2.5). Successivamente a quella mia azione, qualche web crawler avrà fatto scrapping della mia pagina trovando la mia Email all'interno nel sorgente dell'ipertesto (si può notare in figura 2.6 la mail estraibile dal sorgente)

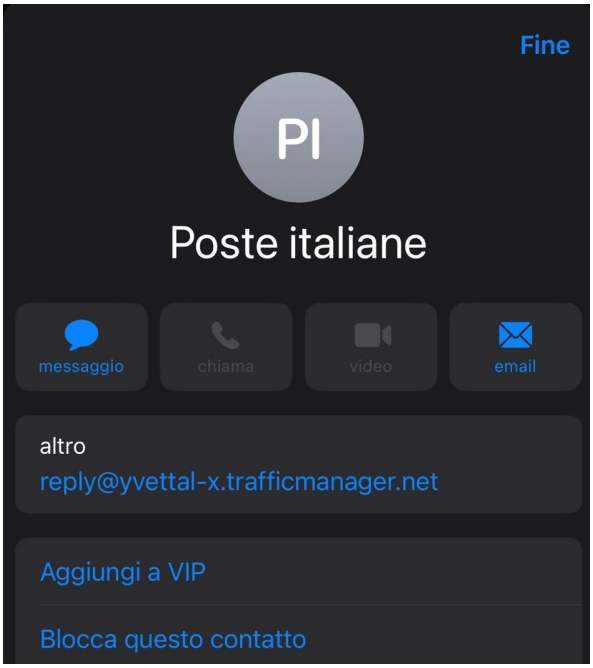


Figura 2.7: In riferimanento alla figura 2.1, ci si potrebbe difendere andando a controllare l'indirizzo da cui è stata inviata la mail

## 2.4 Reti neurali generative

Costruire un sistema in grado di generare in modo proattivo una quantità considerevole di domini di squatting non è banale. Esistono già sistemi in grado di farlo (DNSTwist è un esempio), ma come indicato dagli autori dall'articolo Needle in a Haystack[4], questi generatori di domini di squatting sono in parte affidabili in quanto sono limitati in quanto non riescono a gestire in modo efficace domini di squatting combinati (combo) o domini che cambiano il TLD, inoltre, gli strumenti esistenti sono molto incompleti nel rilevamento dei domini omografici. Troviamo che strumenti come DNSTwist non riescono a mappare l'elenco completo di caratteri Unicode simili. Ad esempio, ci sono 23 diversi caratteri Unicode che sembrano simili alla lettera "un", ma DNSTwist ne cattura solo 13. Queste limitazioni danneggeranno seriamente le nostre possibilità di catturare pagine di phishing occupate.
Uno dei modi per cui si può pensare di generare in modo proattivo dei domini di squatting, è con l'utilizzo di Reti neurali, in particolare, utilizzando reti neurali generative. Le reti neurali generative sono una classe di metodi per l'apprendimento automatico in cui due reti neurali si sfidano diventando uno l'avversario dell'altro (infatti vengono anche chiamate reti neurali generative avversarie). In questo processo, una rete neurale chiamata Generatore genera dati candidati che poi la controparte, chiamata Discriminatore, le valuta. Il generatore quindi cerca di ingannare il discriminatore generando il più possibile dati che rispecchiano quelli reali. Nella figura 2.8 uno schema semplice di GAN

Il modello di rete neurale generativa che utilizzerò per la generazione proattiva di

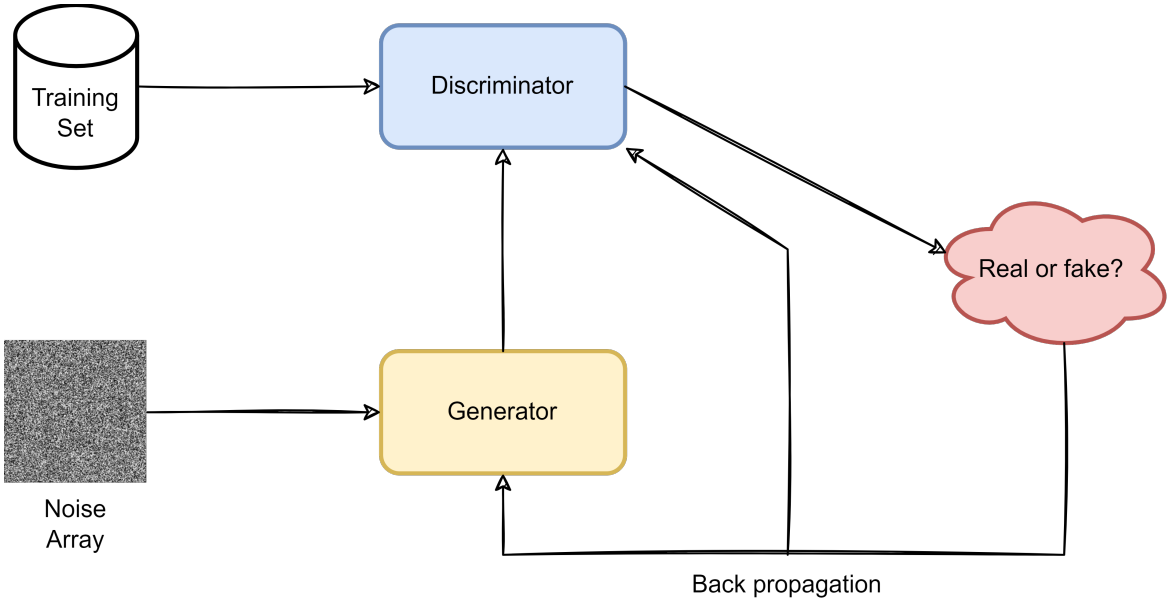


Figura 2.8: Simple Gan Scheme

domini di squatting, è chiamata Deep Convolutional Generative Adversarial Network

## 2.5 Le DCGAN

Le reti convoluzionali generative, seguono la filosofia delle reti generative classiche ma procedono utilizzando una struttura dei layer totalmente differente. Nelle reti generative convoluzionali, sia il generatore che il discriminatore utilizzano reti profonde costituite interamente da strati di convoluzione-deconvoluzione, ovvero da qui il nome "rete convoluzionale generativa".
Le reti convoluzionali (non per forza generative, CNN) sono utilizzate nell'apprendimento automatico in maniera importante soprattutto per il riconoscimento di immagini.

## 2.6 Le GANs nella cybersecurity

Le applicazioni[5] per questi modelli di rete neurale sono davvero molteplici ed il potenziale è davvero enorme, in quanto, questi modelli di reti neurali possono essere istruiti a creare informazioni estremamente simili a qualsiasi dominio della vita reale: immagini, musica, audio, testo.
Vengono, inoltre, utilizzati anche nella branca della sicurezza informatica, la cosa meno entusiasmante è che vengono usate in primo luogo come metodo di attacco.
In primo luogo vengono utilizzate per eludere sistemi di riconoscimento di Malware: un sistema che genera malware appoggiandosi ad un modello di rete GAN è in grado di generare dei malware che sono non identificabili (o difficilmente) dal sistema su cui dovrà essere eseguito.
In secondo luogo possono essere usate per violare sistemi di autenticazione biometrica: sistemi che potrebbero basare i loro permessi di accesso attraverso l'uso della voce o del volto. Come ultimo esempio, ma non meno importante, è l'applicazione delle GAN nei sistemi di password guessing: l'autenticazione della password è uno dei metodi più comunemente utilizzati dagli utenti che tendono a scegliere password facili da indovinare poichè utilizzano stringhe comuni. Questi tipi di stringhe sono soggetti ad attacchi chiamati password guessing in cui un utente malintenzionato tenta di accedere utilizzando un database di stringhe comuni, dizionari di parole e database di password leaked. L'efficacia dell'attacco si basa sulla capacità di testare rapidamente un gran numero di password altamente probabili rispetto a ciascun hash di password. Una tecnica avanzata si basa sull'intuizione su come gli utenti scelgono le password definendo un'euristica per le trasformazioni delle password, che include combinazioni di più parole e lettere maiuscole e minuscole, etc... Poichè lo sviluppo e il test di nuove regole ed euristiche è un'attività dispendiosa in termini di tempo e che richiede competenze specializzate, entrano in gioco le GANs. Poiché la password è una stringa codificata in testo, è possibile utilizzare un approccio basato su GAN in cui una rete neurale viene addestrata per determinare autonomamente le caratteristiche e le strutture delle password e per sfruttare questa conoscenza per generare nuovi campioni che seguono la stessa distribuzione. Le reti neurali profonde sono sufficientemente espressive da acquisire una gamma di proprietà e strutture che descrivono la maggior parte delle password scelte dall'utente e possono essere addestrate senza alcuna conoscenza o ipotesi pregressa. Ciò implica un'ampia gamma di conoscenze per indovinare le password

che includono e superano ciò che viene catturato nelle regole generate dall'uomo e nei processi di generazione delle password.

# Capitolo 3

# SquatGAN: Architettura e Design

## 3.1 Architettura

Il sistema SquatGAN è basato quindi sull'utilizzo principale di una rete neurale generativa convoluzionale (DCGAN), la cui struttura verrà spiegata nei capitoli successivi, e a supporto della rete neurale sono stati aggiunti moduli di supporto, ognuno dei quali con una funzionalità specifica.
Ciò che segue è una spiegazione ad alto livello dell'architettura di SquatGAN in cui verranno illustrati i vari componenti che sono stati necessari alla realizzazione di quest'ultimo.
Come mostrato in figura 3.1, il primo modulo implementato mi permette di produrre un dataset di immagini che sono necessarie per l'addestramento della rete neurale. In secondo luogo abbiamo la rete generativa convoluzionale che riceve come input il dataset di immagini generate e che utilizzerà per addestrare la rete nell'apprendimento e la generazione di nuove immagini. Infine ho implementato un modulo per il riconoscimento ottico di caratteri (OCR. Optical Character Recognition) in modo che le immagini generate dalla rete neurale potessero essere interpretate e tradotte in stringhe di testo (che alla fine saranno i domini di squatting generati).

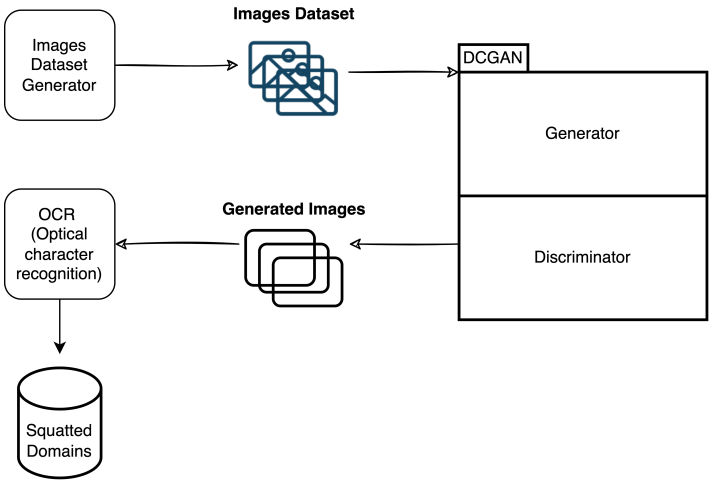


Figura 3.1: SquatGAN high level architecture

## 3.2 Dataset & training

Per la generazione del dataset, ho fatto ricorso all'ausilio di Google Fonts per avere a disposizione una quantità di font esaustiva che mi ha permesso di testare la rete neurale anche utilizzando diversi tipi di font (ma ne parlerò più avanti). Partendo dalla generazione del dataset di immagini, si è utilizzato la libreria PIL(Pillow) di python per produrre delle immagini partendo da del testo (che nel nostro caso il testo in input saranno nomi di dominio). Per un test preliminare ho utilizzato un singolo font per testare il comportamento della GAN su un dataset che non avesse troppa varianza sui dati, senza aggiungere rumore di alcun tipo.
In una versione successiva ho provato ad aggiugnere alle immagini generate (sempre attraverso l'utilizzo di PIL) del rumore all'immagine per verificare il comportamento della DCGAN e le relative immagini generate dal modello.
Una versione più avanzata del modello (di cui non parlerò in questo articolo) consisterebbe nel dare in pasto alla rete le immagini generate fino a quel momento.

## 3.3 Modello di GAN

Il modello di DCGAN sviluppato in una prima versione riceve come input immagini di dimensione 400 x 40 x 1 (1 canale in scala di grigio) normalizzate tra 0 e 1. Il modello è stato costruito utilizzando principalmente in python utilizzando le librerie di Keras e Tensorflow insieme all'ausilio di altre librerie come ad esempio numpy per avere delle strutture dati efficienti su cui memorizzare i tensori. Il generatore è descritto secondo la tabella 3.1

| Filters | Strides | Kernel | Layer | Dropout |
|---|---|---|---|---|
| 5*50*256 | () | () | Dense | 0.1 |
| 256 | (2, 2) | (5, 5) | Conv2DTranspose | 0.1 |
| 128 | (2, 2) | (5, 5) | Conv2DTranspose | 0.1 |
| 1 | (2, 2) | (5, 5) | Conv2DTranspose | 0 |

Tabella 3.1: SquatGAN generator

Ogni strato di Deconvoluzione (conv2DTranspose) è seguito da un BatchNormalization attivato da una LeakyRELU seguito da uno strato di pooling (MAX), Il batch size è stato settato a 256 per tutte le versioni. Per quanto rigurada il discriminatore, è possibile visualizzarlo attraverso la tabella 3.2

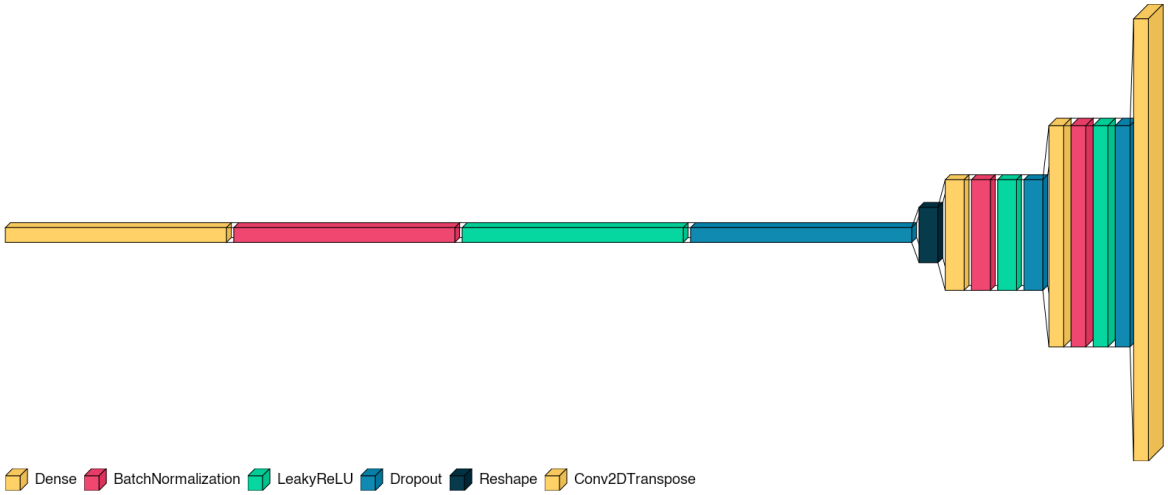


Figura 3.2: SquatGAN generator

| Filters | Strides | Kernel | Layer | Dropout |
|---|---|---|---|---|
| 16 | (2, 2) | (3, 5) | Conv2D | 0.1 |
| 32 | (2, 2) | (3, 5) | Conv2D | 0.1 |
| 64 | (2, 2) | (3, 5) | Conv2D | 0.1 |
| 128 | (2, 2) | (3, 5) | Conv2D | 0.1 |
| 128 | (2, 2) | (3, 5) | Conv2D | 0.1 |
| 256 | () | () | Dense | 0 |
| 128 | () | () | Dense | 0 |
| 64 | () | () | Dense | 0 |

Tabella 3.2: SquatGAN discriminator

## 3.4 Modulo OCR

Il modulo di OCR si occupa dell'interpretazione delle immagini in testo. Nel nostro contesto, le immagini che il modulo OCR riceve sono quelle che il modello GAN produce in output dalla generazione, mentre il testo in uscita dal modulo OCR restituisce i potenziali domini squatted che sono stati generati dalle immagini della DCGAN. Per sviluppare l'OCR ho utilizzato sempre python come linguaggio di programmazione, sfruttando Keras-OCR che permette appunto di interpretare le immagini in stringhe di testo.

# Capitolo 4

# Risultati preliminari

## 4.1 Dataset

Nella prima versione sviluppata, il dataset di immagini generate utilizzando PIL è illustrato nelle immagini 4.1

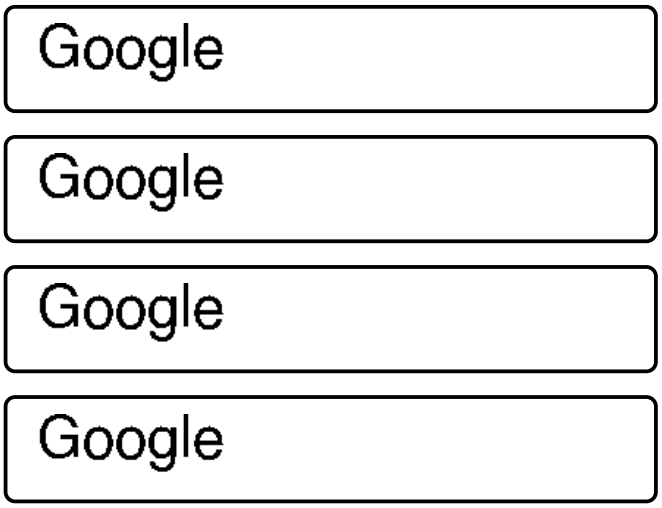


Figura 4.1: In figura sono mostrati 4 esempi di immagini realizzate secondo i criteri illustrati

Le immagini utilizzate come dataset nella prima versione sono semplici immagini 400 x 40 x 1 in cui ho considerato un singolo dominio internet (e.g Google, Facebook, Apple, ...) generando, nel mio caso, 1024 immagini contenenti quel dominio e utilizzando questo set di immagini come Dataset (come input della DCGAN)
In una seconda versione ho incluso nelle immagini del dataset, una maschera di rumore per riprodurre un dataset più dinamico e far in modo che il modello generasse caratteri imprecisi (che è proprio il mio scopo). Nella figura 4.2 è possibile notare la densità del rumore (non ho scelto una densità troppo alta, in quanto il modello non risultava convergere).

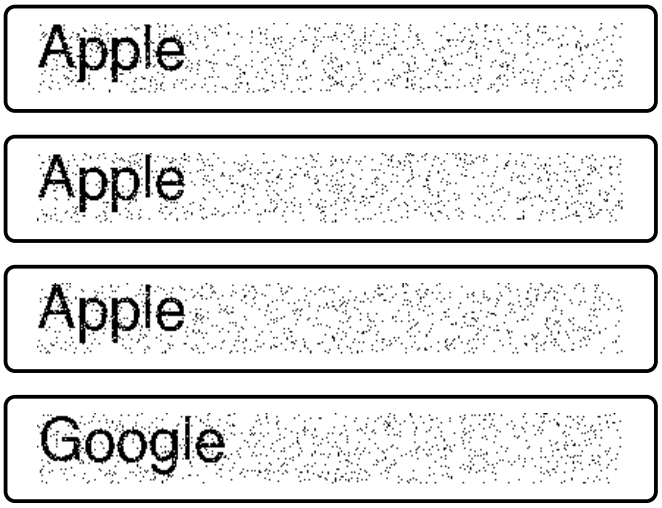


Figura 4.2: La seconda versione del dataset utilizzato dalla DCGAN per produrre immagini. Da notare il rumore aggiunto

Una versione più avanzata del modello sarebbe quella di generare immagini aventi padding arbitrario come mostrato in figura 4.3.

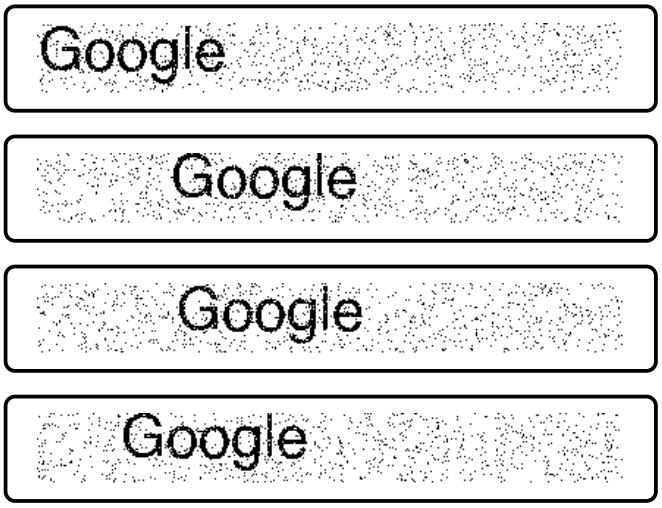


Figura 4.3: Questa versione del dataset presenta padding randomico su entrambi gli assi.

Infine, un'ultima challenge sarebbe quella di produrre un modello che sia in grado di generare immagini, da un dataset di ingresso che utilizza più font, come quelli mostrati in figura 4.4


Figura 4.4: Questo dataset presenta immagini con domini scritti utilizzando font differenti.

## 4.2 Dati generati

Una volta prodotto il dataset, ho utilizzato la DCGAN per analizzare le immagini che era in grado di produrre, cercando di lavorare sui vari strati convoluzione/deconvoluzione in modo da renderla il più fedele possibile.
Dalla prima versione del dataset [4.1], il modello non è riuscito a produrre immagini che rappresentassero in maniera corretta i domini squatted, il problema risiedeva nel fatto che il dataset di partenza era troppo statico (le immagini erano completamente identiche l'une dalle altre). Questo non ha permesso alla rete di produrre immagini con sufficiente rumore tra le singole lettere. In figura 4.5 è visualizzato l'output del modello nella versione 1, mentre in figura 4.6 un estratto della losses dopo 1000 epoch.

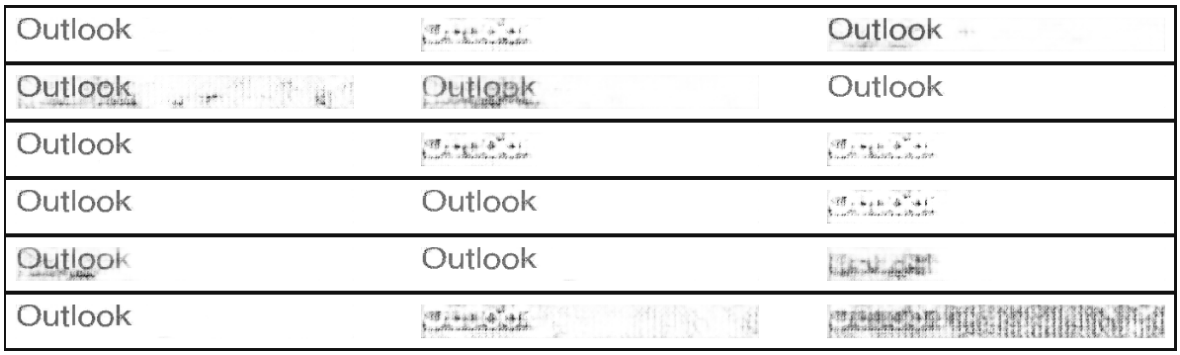


Figura 4.5: I risultati del modello nella prima versione del dataset

Nella seconda versione, avendo immagini con rumore casuale all'interno del dataset, la rete converge in maniera più corretta verso le immagini reali producendo anche testo

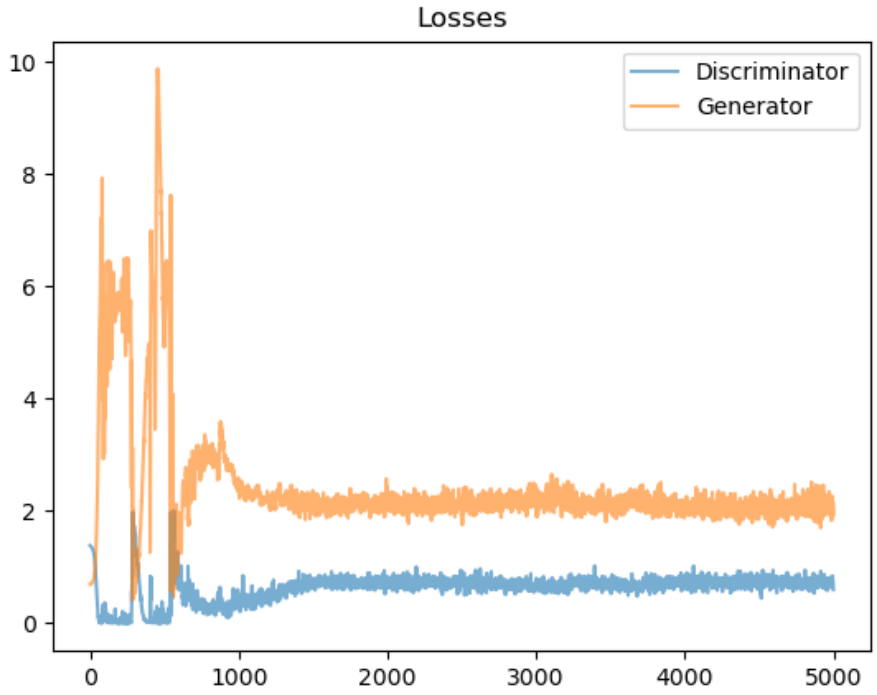


Figura 4.6: losses nella versione 1. Da notare che il generatore ha una loss abbastanza alta rispetto al discriminatore

avente lettere distorte dal rumore (che è proprio quello che si vuole produrre). In figura 4.7 un esempio di output per la seconda versione del modello. In figura 4.9 l'andamento delle losses nell'arco di 1000 epoche.

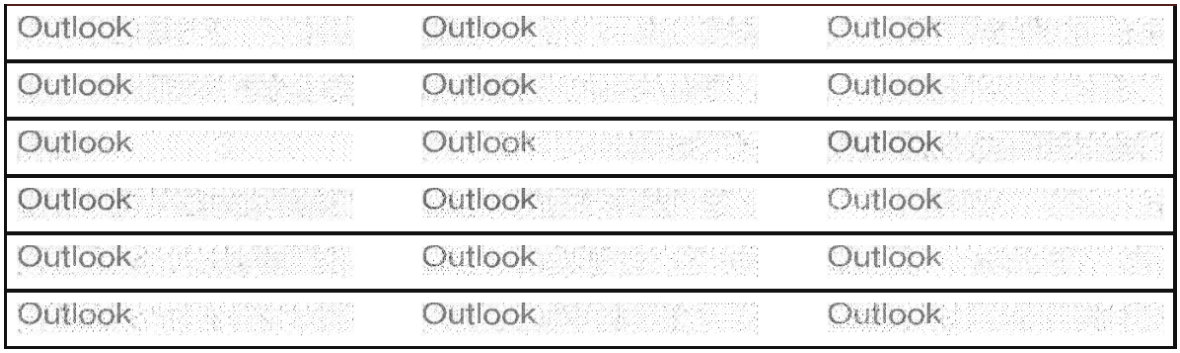


Figura 4.7: Output della DCGAN nella seconda versione. Il test è stato effettuato sul dominio "Outlook". Questi sono i risultati dopo 1000 epoche.

Ho anche provato ad usare dei layer di AveragePooling anzichè MaxPooling ma le immagini generate (figura 4.10) si presentano molto più sfocate, come previsto.

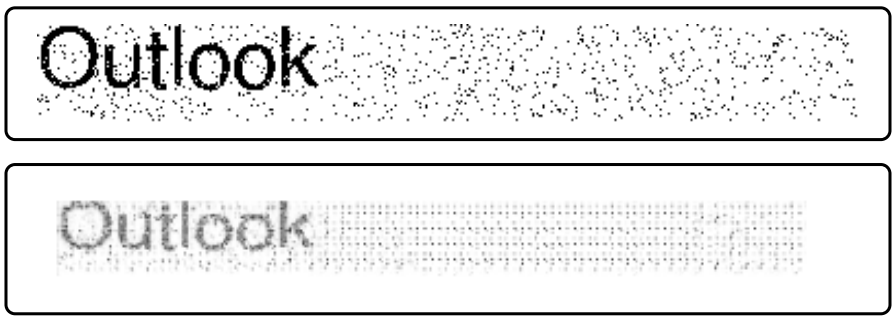


Figura 4.8: Immagine prodotta dalla DCGAN dopo 1000 epoche rispetto ad un'immagine del dataset di partenza

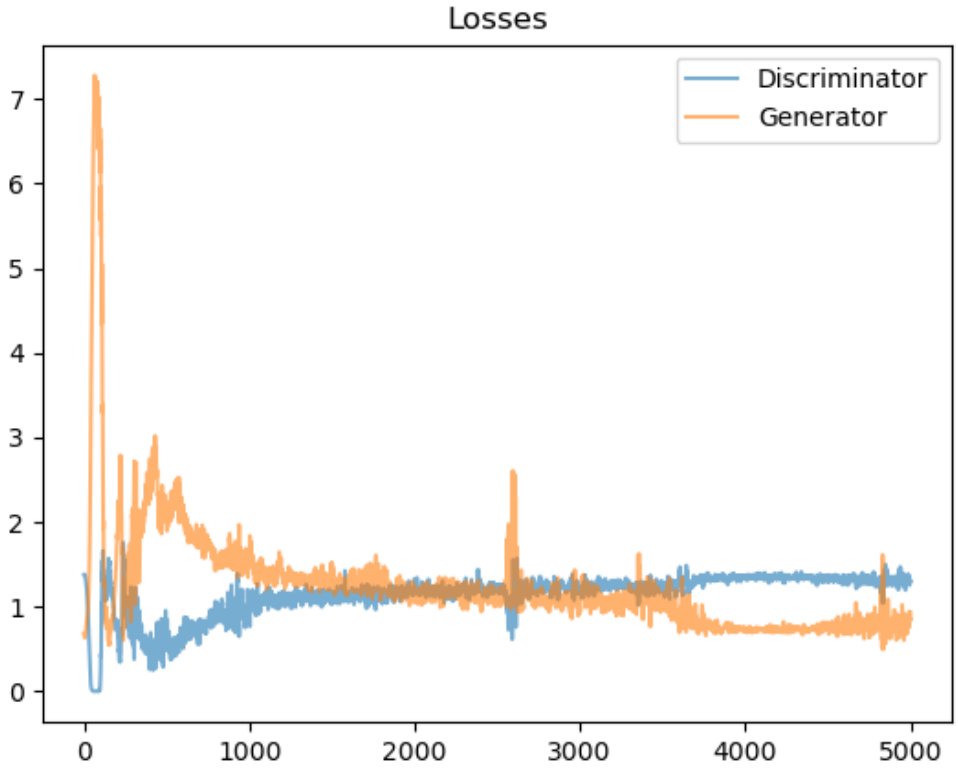


Figura 4.9: losses utilizzando la seconda versione del dataset. In questa versione la losses è nettamente più decente rispetto alla prima versione.

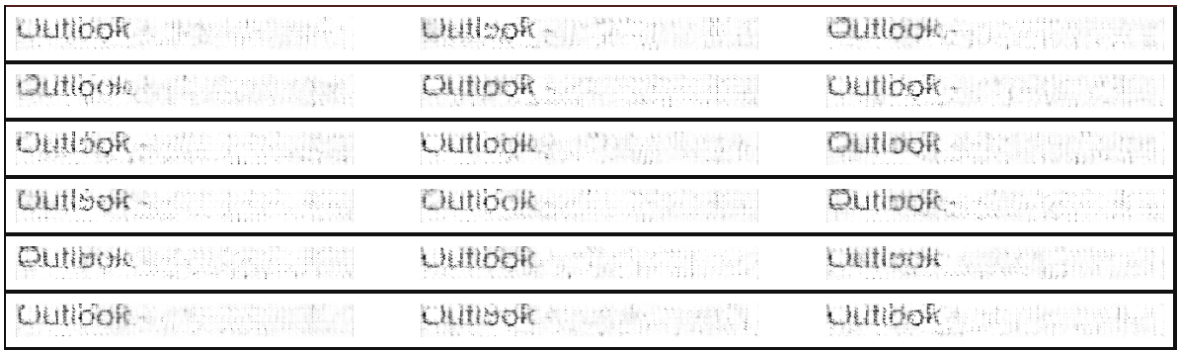

Figura 4.10: DCGAN nella versione 2. Le immagini generate utilizzando layers di AvgPool al posto di MaxPooling le quali presentano troppe sgranature

## 4.3 Validazione

Utilizzando keras come OCR sono riuscito a produrre delle stringhe di testo partendo dalle immagini in output della rete neurale. Nella prima versione del dataset, non si è riusciti ad estrarre una quantità di domini rilevanti, in quanto le immagini prodotte [4.5] risultavano appunto prive di rumore e imprecisioni sui caratteri.
Per estrarre i potenziali domini di squatting, ho allenato il modello inizialmente con la prima versione del dataset [4.5] fino a 1000 epoche di addestramento (per le successive versioni del dataset ho utilizzato lo stesso approcio). Successivamente ho fatto produrre alla rete delle immagini di test che ho utilizzato come input per il modulo OCR (Keras). Il modulo OCR produceva una quantità di domini non rilevante e soprattutto con ripetizioni. Ho utilizzato un algorimo per la rimozione dei duplicati ma risultavano essere comunque rilevanti. Il problema non era il fatto che fossero tanti ma che alcuni domini riconosciuti si discostavano troppo da quello che era il dominio di partenza. Per ovviare a questo problema ho utilizzato un algoritmo di Edit Distance (The Levenshtein distance algorithm) per estrarre solo i domini che si discostassero di una certa edit distance dal dominio di partenza. Inzialmente utilizzando come valore di limite 5, ma infine come valore ottimale ho osservato che andasse bene 3. Qui una lista di potenziali domini di squatting estratti dal Modulo OCR utilizzando la prima versione de dataset.

outlook,outlock,outlogk,cutlock,outosk,cutgo,cutlook,outlgok,cutlgok,
cutloss,outloock,utogks,wlook,outlouk,outrok.

apple,acoli,pplet,applet,aple,updle,spple,apute,narlle,aprle,appiet,applel,
cpple

Utilizzando invece la seconda versione del dataset, introducendo il rumore alle immagini, si è visto un aumento dei potenziali domini prodotti. Qui una lista per i domini più famosi.

gutlook,cltlook,cutlook,cutook,dutlook,butlook,butook,gatlook,
cutiook,sutook,sutlook,outlook,sutiook,dutloor,gutook,cutloo,sltlook,
eatlook,eutlook,outiook,duatlook,cutloek,gutloek,cuttook,dutook,suatlook,
outook,dutiook,eutiook,outlooks,autloor,mutlook,suttook,dutlock,oatiook,
oltlook,utook,outtook,sutlock,sutlookk,wutlook,buttook,dutloon,butiook,
cutleok,sutloek,atlook,gutloon,wetlook,gltlook,gutiook,outlooke,outioon,
outloor,cutloor,ontiook,catlook,guatlook,mutook,guttook,putloor,utlook,
cuatlook,gutlock,butlock,oatlook,cutloos,euttook,outlock,sutloor,cutloon,
nutlook,butloak,putloo,duttook,dutloo,bltlook,sutloo,oltook,outloo,dltlook,
butloo,eutlooks,sutloon,eutloor,cutlock,datlook,gutloor,satlook,eutook,
otlook,qutlook,outoek,dutloek,oltlooks,putlook,outlolt,rutlook,olatlook,
dutloos,qutook.

poogle,googe,soogle,google,coogle,gooale,cooge,foogle,saogle,eoogle,gocgle,
oogle,ooge,sooge,cooglle,cooale,cgogle,fooge,gtogle,socgle,goegle,eooge,
ggogle,ooogle,sooale,gdogle,oegle,scogle,sgogle,oocgle,googie,gcogle,saoogle,

sooole,cogle,gogle,fcogle,caoogle,fgogle,fogle,soegle,soosle,sogle,agogle,
googlle,tsoogle.

appias,apdle,saole,faple,appled,adple,sepple,asple,appls,spple,aoples,
eliple,apple,spples,adpls,sltle,appie,apople,aople,atple,acpie,spps,afple,
sapple,fappis,sppie,aple,fapple,applg,faaple,appile,fappie,fadple,ooe,aepie,
applc,aples,aoe,appies,appec,anple,aaple,adplu,sadple,aeals,sadole,adte,
apoles,seple,appe,adols,asol,apples,apole,sadpie,sols,dple,adpole,fapole,
sole,apols,adole,adpe,adpile,acpln,acqie,appla,acole,apile,arple.

amazon,anazon,atazon,aazon,amazan,amnazon,asenazon,aanazon,amezon,
aeazon,eazoc,aaazor,asmazoe,aaazon,aetazon,aazor,armazon,amelzon,amaaon,
amaizon,aeton,ameion,aazos,aaezon,ametzan,amaon,amzon,mazon,amason,
aiazon,amaizan,amtton,amtazon,ameizon,arnazon,amezan,atmazen,ameszon,
rmazon,aanezon,aegazon,aatazon,atazen,amezen,smazon,samazon,aesazon,
aseazon,araen,asmazon,agazon,dazon,amazn,aeaon,orazo,pazon,aezon,amazo,atmazon,
aenazon,anaon,famazon,aazoe,aoazon,anatzon,asnazon,aareon,amaton,auazon,
amazen,anoazon,aazan,ahazon,aaazod,aniizon,aagazos,aeazan,anezon,aagazoen,
ramaizon,amazoe,amaor,annazon,amrzon,

# Conclusione e lavori futuri

Il lavoro sviluppato fino ad ora è sicuramente un punto di inizio per creare un sistema completo in grado di produrre in modo proattivo dei domini di squatting di tutte le tipologie elencate ad inizio articolo. In primo luogo bisognerebbe adattare la rete neurale ad essere in grado di auto apprendere il fatto che alcuni tipi di rumori sono utili per l'obbiettivo che vogliamo raggiungere.
In secondo luogo è necessario utilizzare un OCR in grado di riconoscere alfabeti differenti in modo da generare domini omografici più complessi che comprendano lettere di alfabeti differenti in quanto i domini generati e riconosciuti dalla versione attuale comprendono solo lettere dell'alfabeto inglese. Utilizzando Keras come OCR, ed essendo anch'esso una rete neurale, ho caricato i pesi necessari al riconoscimenti di lettere dell'alfabeto inglese, bisognerebbe caricare un HDF (Hierarchical Data Format) consono al riconoscimento di altri alfabeti oltre quello inglese. Per completare il lavoro bisognerebbe aggiungere un modulo supplementare successivo all'OCR in grado di individuare quali, tra i domini riconosciuti e prodotti da SquatGAN, sono effettivamente attivi e potenziali siti di phishing. Questo sarebbe possibile farlo attraverso semplici lookup DNS. L'architettura spiegata in figura 3.1 risulterebbe quindi come quella mostrata in figura 4.11

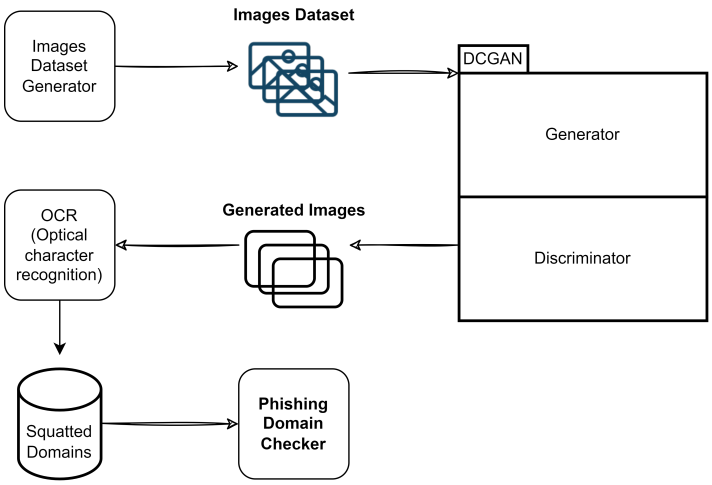


Figura 4.11: un'ipotetica SquatGAN Architecture futura

# Bibliografia


[1] L. J. Sern, Y. G. P. David, and C. J. Hao, "Phishgan: Data augmentation and identification of homoglyph attacks," in *2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*, pp. 1–6, IEEE, 2020.

[2] N. Dewani, Z. Khan, A. Agarwal, M. Sharma, and S. Khan, *Handbook of Research on Cyber Law, Data Protection, and Privacy*. Advances in Information Security, Privacy, and Ethics Series, IGI Global, 2022.

[3] I. Dutta, B. Ghosh, A. Carlson, M. Totaro, and M. Bayoumi, "Generative adversarial networks in security: A survey," 10 2020.

[4] K. Tian, S. T. K. Jan, H. Hu, D. Yao, and G. Wang, "Needle in a haystack: Tracking down elite phishing domains in the wild," IMC '18, (New York, NY, USA), p. 429–442, Association for Computing Machinery, 2018.

[5] I. K. Dutta, B. Ghosh, A. Carlson, M. Totaro, and M. Bayoumi, "Generative adversarial networks in security: A survey," in *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pp. 0399–0405, 2020.