



UNIVERSITY OF SALERNO

DEPARTMENT OF INFORMATION ENGINEERING,
ELECTRICAL ENGINEERING AND APPLIED
MATHEMATICS

PROJECT REPORT

AI FOR CYBERSECURITY

TRACCIA 1: VALUTAZIONE DELLA SICUREZZA DI UN
SISTEMA DI FACE RECOGNITION

prof. Greco Antonio

Group 07	
Students	Id number
Alberti Andrea	0622702370
Attianese Carmine	0622702355
Capaldo Vincenzo	0622702347
Esposito Paolo	0622702292

CONTENTS

1	Introduzione	4
1.1	Presentazione del progetto	4
1.1.1	Analisi dell'efficacia degli attacchi sulla rete NN1	4
1.1.2	Analisi della trasferibilità degli attacchi	4
1.1.3	Analisi del sistema di difesa implementato	5
1.2	Dataset utilizzato (VGG-Face2)	5
1.2.1	Costruzione del Test Set	5
1.2.2	Costruzione del Detectors Train Set	6
1.2.3	Preprocessing delle immagini	6
1.3	Security by Design	7
1.3.1	Conoscere l'avversario	7
1.3.2	Essere proattivo	7
1.3.3	Implementare le difese	11
2	Analisi dell'efficacia degli attacchi sulla rete NN1	12
2.1	Informazioni sulla rete NN1: InceptionResnetV1	12
2.2	Prestazioni della rete NN1 sulle immagini clean	12
2.3	Prestazioni della rete sulle immagini adversarial FGSM	13
2.3.1	FGSM untargeted	13
2.3.2	FGSM targeted	13
2.4	Prestazioni della rete sulle immagini adversarial BIM	14
2.4.1	BIM untargeted	14
2.4.2	BIM targeted	15
2.5	Prestazioni della rete sulle immagini adversarial PGD	16
2.5.1	PGD untargeted	16
2.5.2	PGD targeted	17
2.6	Prestazioni della rete sulle immagini adversarial DF	18
2.6.1	DF untargeted	19
2.7	Prestazioni della rete sulle immagini adversarial CW	20
2.7.1	CW untargeted	21
2.7.2	CW targeted	22
2.8	Riepilogo sull'efficacia degli attacchi sulla rete NN1	23
3	Analisi della trasferibilità degli attacchi	24
3.1	Informazioni sulla rete NN2: SENet50	24
3.2	Preprocessing delle immagini	24
3.3	Prestazioni della rete NN2 sulle immagini clean	25
3.4	Prestazioni della rete sulle immagini adversarial FGSM	25
3.4.1	FGSM untargeted	25
3.4.2	FGSM targeted	26
3.5	Prestazioni della rete sulle immagini adversarial BIM	26

3.5.1	BIM untargeted	27
3.5.2	BIM targeted	28
3.6	Prestazioni della rete sulle immagini adversarial PGD	29
3.6.1	PGD untargeted	30
3.6.2	PGD targeted	31
3.7	Prestazioni della rete sulle immagini adversarial DF	32
3.7.1	DF untargeted	33
3.8	Prestazioni della rete sulle immagini adversarial CW	34
3.8.1	CW untargeted	34
3.9	Riepilogo sull'analisi della trasferibilità degli attacchi	36
4	Analisi del sistema di difesa implementato	37
4.1	Sistema di difesa implementato	37
4.2	Architettura dei singoli detector	38
4.3	Addestramento dei singoli detectors	38
4.3.1	Costruzione del train set	39
4.4	Testing dei singoli detectors	40
4.4.1	FGSM Detector	41
4.4.2	BIM Detector	42
4.4.3	PGD Detector	43
4.4.4	DF Detector	44
4.4.5	CW Detector	45
4.5	Criteri di valutazione del classificatore NN1+Detectors	46
4.6	Prestazioni della rete sulle immagini clean	46
4.7	Prestazioni della rete sulle immagini adversarial FGSM	47
4.7.1	FGSM untargeted	47
4.7.2	FGSM targeted	47
4.8	Prestazioni della rete sulle immagini adversarial BIM	48
4.8.1	BIM untargeted	48
4.8.2	BIM targeted	49
4.9	Prestazioni della rete sulle immagini adversarial PGD	51
4.9.1	PGD untargeted	51
4.9.2	PGD targeted	52
4.10	Prestazioni della rete sulle immagini adversarial DF	54
4.10.1	DF untargeted	54
4.11	Prestazioni della rete sulle immagini adversarial CW	55
4.11.1	CW untargeted	56
4.12	Riepilogo sull'analisi del sistema di difesa	57
5	Conclusione	58

ABSTRACT

Il lavoro svolto ha come obiettivo la valutazione della sicurezza e della robustezza di un sistema di riconoscimento facciale, basato su reti neurali profonde, rispetto ad attacchi adversarial. L'analisi è stata condotta nel rispetto di specifici vincoli progettuali sulla perturbazione massima ammessa rispetto all'immagine originale. L'approccio seguito è il *security by design*, che pone al centro del processo progettuale la prevenzione delle vulnerabilità, attraverso la conoscenza dell'attaccante, la simulazione di possibili scenari di attacco e l'integrazione di contromisure robuste all'interno dell'architettura del sistema.

In una prima fase, è stata evidenziata l'estrema fragilità della rete NN1 (InceptionResNetV1) in assenza di meccanismi di difesa: anche perturbazioni minime, impercettibili all'occhio umano, sono risultate sufficienti per causare gravi errori di classificazione, portando in alcuni casi l'accuracy della rete a valori prossimi allo zero. Sono stati presi in esame cinque tipi di attacchi (FGSM, BIM, PGD, DeepFool e Carlini-Wagner), analizzandone l'efficacia in modalità untargeted e targeted, al variare dei principali parametri. L'impatto degli attacchi è stato misurato tramite *security evaluation curves*, con l'obiettivo di quantificare il degrado delle prestazioni della rete in funzione dell'intensità delle perturbazioni.

Successivamente, è stata effettuata un'analisi della trasferibilità degli attacchi, che ha mostrato come le immagini adversarial generate per la rete NN1 riescano spesso a compromettere anche le prestazioni della rete NN2 (SENet50). Ciò è particolarmente rilevante in contesti reali, dove l'attaccante potrebbe non conoscere l'architettura della rete target.

Infine, per far fronte a questi attacchi, è stato progettato un meccanismo di difesa (integrato alla rete NN1), basato su un insieme di detectors specializzati (uno per ciascun tipo di attacco). I detectors operano come classificatori binari e agiscono a monte della rete NN1: un'immagine viene considerata sicura e quindi inoltrata alla classificazione solo se tutti i detectors la riconoscono come non adversarial. I risultati sperimentali dimostrano che questo approccio è altamente efficace: in presenza del sistema di difesa, il modello NN1 è riuscito a mantenere un'elevata accuratezza anche sotto attacco, mitigando quasi completamente tutte le tecniche di attacco. Tuttavia, l'utilizzo dei detectors introduce un sovraccarico nel costo computazionale della classificazione dei campioni e un leggero degrado delle performance sui dati clean, a causa della rilevazione di alcuni falsi positivi.

In conclusione, il progetto ha dimostrato l'importanza di un'analisi approfondita della sicurezza nei sistemi basati su intelligenza artificiale, evidenziando come anche modelli estremamente performanti possano essere vulnerabili ad attacchi mirati. La comprensione delle possibili minacce, unite all'integrazione di opportune tecniche di difesa, rappresenta un passo fondamentale verso la costruzione di sistemi robusti, affidabili e realmente utilizzabili in contesti applicativi critici.

CHAPTER 1

INTRODUZIONE

1.1 Presentazione del progetto

L’obiettivo di questo progetto è analizzare la sicurezza e la robustezza di un sistema di riconoscimento facciale basato su reti neurali profonde, specificamente la rete pre-addestrata Inception-ResnetV1 (da qui in poi ”NN1”), rispetto ad attacchi avversari. Questi attacchi consistono nell’applicazione di perturbazioni minime e spesso impercettibili alle immagini di input, con il preciso scopo di indurre il classificatore a produrre classificazioni errate, pur mantenendo l’immagine indistinguibile dall’originale per l’occhio umano.

Il sistema in esame si basa su una rete neurale profonda pre-addestrata sul dataset **VGGFace2**, un ampio set di immagini di volti umani ampiamente utilizzato come standard di riferimento in ambito di face recognition.

1.1.1 Analisi dell’efficacia degli attacchi sulla rete NN1

Nel corso di questa sperimentazione, il comportamento della rete NN1 è stato analizzato sia in condizioni operative normali, utilizzando immagini non modificate provenienti da un test set appositamente costruito, sia in presenza di **attacchi avversari**. Questi ultimi sono stati generati artificialmente utilizzando la libreria *ART (Adversarial Robustness Toolbox)*, un framework robusto che offre diverse tecniche di attacco all’avanguardia. Gli attacchi specifici impiegati nel progetto sono: *Fast Gradient Sign Method (FGSM)*, *Basic Iterative Method (BIM)*, *Projected Gradient Descent (PGD)*, *DeepFool (DF)* e *Carlini-Wagner (CW)*.

È stato valutato l’effetto di attacchi *untargeted (error-generic)* e di attacchi *targeted (error-specific)*, entrambi al variare dei parametri degli attacchi, quantificando la robustezza del sistema attraverso l’implementazione e la discussione di opportune *security evaluation curves*, le quali offrono una rappresentazione visiva dell’accuratezza del modello al variare dell’intensità dell’attacco.

Gli attacchi utilizzati possono avere intensità massima L_{∞} pari al 5% del range di valori rappresentabili. La perturbazione L_{∞} è definita come la massima differenza assoluta tra i valori di ogni singolo pixel di due immagini. In altre parole, rappresenta il ”cambiamento più grande” consentito a qualsiasi pixel di un’immagine rispetto all’originale.

1.1.2 Analisi della trasferibilità degli attacchi

Un altro aspetto cruciale dello studio ha riguardato la verifica della trasferibilità di tali attacchi. Si è investigato se gli esempi adversarial generati per ingannare NN1 fossero efficaci anche contro un’altra rete neurale pre-addestrata per il riconoscimento facciale (da qui in poi ”NN2”), senza aver accesso ai suoi parametri interni.

Questa analisi è fondamentale per comprendere la **generalizzabilità** delle minacce avversarie in scenari più realistici dove l’attaccante potrebbe non avere piena conoscenza del modello target.

1.1.3 Analisi del sistema di difesa implementato

Infine, il progetto esplora anche l'implementazione di un possibile metodo di difesa. In particolare, sono stati sviluppati e addestrati dei **detectors** basati su reti neurali: ciascun detector è stato addestrato per una specifica tipologia di attacco (FGSM, BIM, PGD, DF, CW). L'obiettivo primario di questi detectors è identificare e mitigare la presenza di campioni avversari. Per l'addestramento di ciascun detector, è stato utilizzato un dataset bilanciato composto da 1000 campioni clean e 1000 campioni adversarial (generati specificamente per l'attacco corrispondente). Per garantire la massima sicurezza, un'immagine, prima di essere classificata dalla rete principale (NN1), deve superare positivamente il controllo di tutti i detectors.

1.2 Dataset utilizzato (VGG-Face2)

Il dataset utilizzato nell'ambito di questo progetto è il **VGGFace2**, uno dei dataset più significativi nella comunità di ricerca per lo sviluppo di sistemi di riconoscimento facciale, utilizzato anche per l'addestramento delle reti NN1 e NN2. Esso include circa 3.31 milioni di immagini di volti di celebrità, distribuite su un totale di 9.131 soggetti (8.631 per il train e 500 per il test). Per ogni identità, il numero di immagini varia da un minimo di 80 a un massimo di 843, con una media di circa 362 immagini per soggetto. Dal punto di vista del genere, il dataset è relativamente bilanciato, anche se si osserva una leggera prevalenza di soggetti maschili, pari al 59.3%. Ogni immagine è composta da tre canali e ha dimensioni (verticale e orizzontale) variabili.

Il file *identity_meta.csv* fornisce i metadati per ciascuna identità presente nel dataset; in particolare, ogni riga rappresenta un soggetto e contiene le seguenti informazioni:

Campo	Descrizione
<i>Class_ID</i>	Identificativo univoco per ogni identità
<i>Name</i>	Nome dell'identità
<i>Sample_Num</i>	Numero totale di immagini disponibili per quell'identità del dataset
<i>Flag</i>	Indica se il soggetto è presente nel train set (1) o test set (0)
<i>Gender</i>	Genere del soggetto (m o f)

Inoltre, è stato utilizzato il file *rcmalli_vggface_labels_v2.npy* che contiene una mappatura tra i nomi delle identità e gli ID numerici delle classi (ogni classe rappresenta un'identità), utile per associare ogni identità alla propria label in fase di classificazione.

1.2.1 Costruzione del Test Set

Il test set, impiegato per la valutazione delle prestazioni delle reti "NN1", "NN2" e "NN1 + detectors", include sia immagini clean che immagini adversarial ed è stato costruito seguendo i seguenti passaggi:

- **Test Set Clean:** sono state selezionate casualmente 100 identità distinte dal file *identity_meta.csv* e sono state inserite all'interno del file *test_set.csv* e poi, per ciascuna delle 100 identità selezionate, sono state estratte in modo casuale 10 immagini dal dataset originale, per un totale di 1000 immagini. Queste immagini clean sono state organizzate e memorizzate all'interno della directory *dataset/test_set/clean/original*.
- **Test Set Adversarial:** a partire da queste 1000 immagini clean, sono state generate 1000 immagini adversarial per ciascuna delle tipologie di attacco considerate (FGSM, BIM, PGD, DF, CW), sia untargeted che targeted (ad eccezione di DF che prevede solo attacchi untargeted) e per ognuna delle configurazioni di parametri stabilite. Gli attacchi sono stati generati utilizzando la rete NN1 come modello target e rispettando i vincoli di perturbazione (ove possibile). Queste immagini adversarial sono state organizzate e memorizzate all'interno della directory *dataset/test_set/adversarial_examples*.

Dunque, ogni modello è stato testato individualmente sui dati clean, utilizzando un dataset composto da 1000 immagini originali, e sui dati adversarial, utilizzando un dataset composto da 1000 immagini adversarial per ciascuna delle tipologie di attacco considerate e per ognuna delle configurazioni di parametri stabiliti.

1.2.2 Costruzione del Detectors Train Set

Il train set, impiegato per l'addestramento dei detectors di campioni adversarial, include sia immagini clean che immagini adversarial ed è stato costruito seguendo i seguenti passaggi:

- **Detectors Train Set Clean:** sono state selezionate casualmente 1000 immagini dal dataset originale. Queste immagini clean sono state organizzate e memorizzate all'interno della directory *dataset/detectors_train_set/clean/original*.
- **Detectors Train Set Adversarial:** a partire da queste 1000 immagini clean, sono state generate 1000 immagini adversarial per ciascuna delle tipologie di attacco considerate (FGSM, BIM, PGD, DF, CW), scegliendo in maniera casuale i valori dei parametri dell'attacco di maggiore interesse. Gli attacchi sono stati generati utilizzando la rete NN1 come modello target. Queste immagini adversarial sono state organizzate e memorizzate all'interno della directory *dataset/detectors_train_set/adversarial_examples*.

Dunque, ogni detector è stato addestrato individualmente utilizzando un dataset bilanciato, composto da 1000 campioni clean e 1000 campioni adversarial relativi al tipo di attacco da rilevare.

1.2.3 Preprocessing delle immagini

Le immagini clean, sia quelle del test set che quelle del detectors train set, vengono sottoposte alle seguenti due operazioni di preprocessing, al fine di fare in modo che siano nel formato e nelle dimensioni richiesti dai modelli di classificazione:

1. **Resize e CenterCrop:** ogni immagine viene dapprima ridimensionata in modo che il suo lato più corto sia pari a 256 pixel, mantenendo le proporzioni originali, e poi viene eseguito un ritaglio centrale per ottenere una dimensione finale di 224×224 pixel. È importante sottolineare che, sebbene la rete neurale *NN1* sia stata originariamente addestrata su immagini di dimensione 160×160 , questa scelta è motivata dalla necessità di valutare la trasferibilità degli attacchi avversari sulla rete *NN2*, la quale accetta esclusivamente immagini di dimensione 224×224 (quindi gli attacchi generati sulla rete *NN1* devono essere eseguiti su immagini 224×224). Questa variazione introduce una leggera perdita di accuratezza della rete *NN1* sulle immagini clean, ma è accettabile ai fini dello studio. Queste immagini processate sono state organizzate e memorizzate all'interno delle directory *dataset/test_set/clean/processed* e *dataset/detectors_train_set/clean/processed*.



Figure 1.1: Esempio di *Resize* e *CenterCrop* applicato alle immagini.

2. **Trasformazione dell'intervallo di rappresentazione:** successivamente, quando le immagini vengono lette dalla memoria, viene modificato il loro intervallo di rappresentazione dei pixels. In particolare, le immagini vengono convertite in tensori e il loro intervallo di rappresentazione viene convertito da valori interi nell'intervallo $[0, 255]$ a valori in virgola mobile nell'intervallo $[0.0, 1.0]$. Inoltre, viene applicata un'operazione di normalizzazione a ciascuno dei tre canali (RGB), sottraendo la media 0.5 e dividendo per la deviazione standard 0.5, in maniera tale che l'intervallo di rappresentazione venga convertito da $[0.0, 1.0]$ a $[-1.0, 1.0]$. Tale normalizzazione è richiesta dal modello della rete NN1.

1.3 Security by Design

La metodologia seguita in questo progetto è guidata dall'approccio proattivo *Security by Design*, il cui obiettivo è prevenire le vulnerabilità e costruire sistemi robusti per natura, anziché tentare di rimediare a problemi di sicurezza a posteriori. Questo paradigma si basa sulle seguenti tre regole fondamentali, che sono state applicate nel contesto di questo studio per analizzare e migliorare la robustezza del sistema di riconoscimento facciale:

1. **conoscere l'avversario;**
2. **essere proattivo;**
3. **implementare le difese.**

1.3.1 Conoscere l'avversario

Il primo passo dell'approccio security by design è *conoscere l'avversario*, ovvero definire una modellazione accurata degli obiettivi, della conoscenza e delle capacità di un potenziale attaccante. Nell'ambito di questo progetto, questi aspetti sono stati definiti come segue:

- **obiettivo dell'attaccante:**
 - **untargeted (error-generic):** l'attaccante mira a causare un errore di classificazione generico, senza specificare una classe desiderata;
 - **targeted (error-specific):** l'attaccante mira a causare un errore di classificazione specifico, specificando la classe desiderata.
- **conoscenza dell'attaccante:**
 - **white-box:** per la rete NN1, dove l'attaccante ha pieno accesso alle informazioni sul sistema (dataset di addestramento, il feature set, l'algoritmo di apprendimento, la funzione obiettivo, i parametri e gli iperparametri addestrati);
 - **black-box:** per la rete NN2, dove l'attaccante non ha accesso alle informazioni sul sistema (serve a valutare la trasferibilità, ovvero quanto gli attacchi generati su NN1 fossero efficaci anche su un modello differente).
- **capacità dell'attaccante:**
 - **data manipulation constraints:** gli attacchi utilizzati possono avere intensità massima L_{∞} pari al 5% del range di valori rappresentabili;
 - **attack influence:** l'attaccante può manipolare solo il test set (*evasion attack*).

1.3.2 Essere proattivo

Il secondo passo dell'approccio security by design è *essere proattivo*, ovvero anticipare l'avversario e prevedere quali possono essere i suoi possibili attacchi. In linea con questo principio, nell'ambito di questo progetto sono stati simulati e analizzati una vasta gamma di attacchi di evasione contro la rete di riconoscimento facciale NN1. Gli attacchi considerati, tutti generati utilizzando la libreria **ART (Adversarial Robustness Toolbox)**, includono:

- **Fast Gradient Sign Method (FGSM):** è un attacco che sfrutta la conoscenza dei gradienti della funzione di perdita del modello target per generare esempi avversari. La logica intrinseca di FGSM consiste nell'applicare una perturbazione calcolata in modo da massimizzare il valore della funzione di perdita per la classe corretta dell'immagine. In figura 1.2 è riportato un esempio di attacco FGSM al variare della forza dell'attacco. Il parametro principale che regola il comportamento dell'attacco FGSM è:
 - *epsilon*: determina la perturbazione massima applicabile a ciascun pixel dell'immagine.

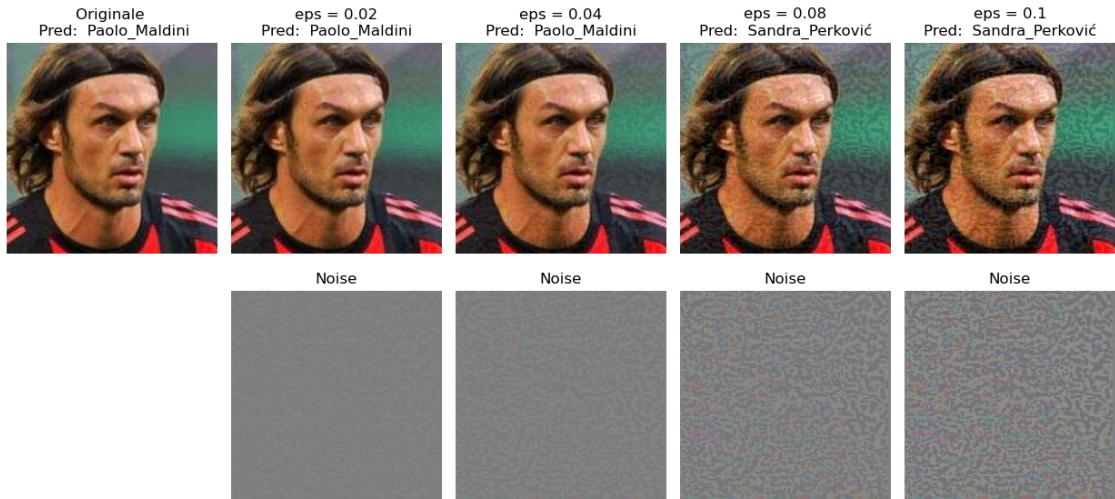


Figure 1.2: Esempio di attacco FGSM al variare della forza dell’attacco.

- **Basic Iterative Method (BIM):** è un’estensione dell’attacco FGSM. Anche in questo caso, l’attaccante sfrutta i gradienti della funzione di perdita del modello target per generare esempi avversari. Tuttavia, a differenza di FGSM che applica la perturbazione in un unico passaggio, BIM lo fa in maniera incrementale, distribuendo la perturbazione su più iterazioni successive. Questo approccio iterativo consente di affinare progressivamente la perturbazione, rendendo l’attacco spesso più efficace nel causare una misclassificazione. In figura 1.3 è riportato un esempio di attacco BIM al variare della forza dell’attacco. I parametri principali che regolano il comportamento dell’attacco BIM sono:

- *epsilon*: determina la perturbazione massima applicabile a ciascun pixel dell’immagine;
- *epsilon_step*: indica la variazione che può essere applicata ad ogni step del processo iterativo;
- *max_iter*: rappresenta il numero massimo di iterazioni.



Figure 1.3: Esempio di attacco BIM al variare della forza dell’attacco.

- **Projected Gradient Descent (PGD)**: è un attacco simile a BIM. Anche in questo caso, l'attaccante sfrutta i gradienti della funzione di perdita del modello target per generare esempi avversari, in maniera iterativa. Tuttavia, a differenza di BIM, PGD introduce un'inizializzazione casuale: l'attacco non parte dall'immagine originale, ma da una versione perturbata di essa, ottenuta aggiungendogli un rumore casuale preso dall'intervallo $[-\epsilon; \epsilon]$. Questo consente una maggiore esplorazione dello spazio delle perturbazioni e aumenta le probabilità di trovare attacchi più efficaci. Inoltre, dopo ogni passo di perturbazione, PGD proietta il campione perturbato all'interno di una sfera di raggio ϵ , in modo da garantire che la perturbazione rimanga entro i limiti consentiti. In figura 1.4 è riportato un esempio di attacco PGD al variare della forza dell'attacco. I parametri principali che regolano il comportamento dell'attacco PGD sono:

- *epsilon*: determina la perturbazione massima applicabile a ciascun pixel dell'immagine;
- *epsilon_step*: indica la variazione che può essere applicata ad ogni step del processo iterativo;
- *max_iter*: rappresenta il numero massimo di iterazioni;
- *num_random_init*: specifica quante inizializzazioni casuali vengono effettuate per ciascuna immagine. L'attacco viene ripetuto più volte, partendo ogni volta da un punto diverso all'interno della sfera di raggio ϵ , e viene selezionato il risultato migliore tra quelli ottenuti. Per questo motivo, al fine di rendere l'attacco più robusto ed efficace, questo parametro è stato fissato al valore di 5 in tutte le prove.



Figure 1.4: Esempio di attacco PGD al variare della forza dell'attacco.

- **DeepFool (DF)**: è un attacco che cerca la minima perturbazione necessaria per far sì che un input sia classificato erroneamente, spostandolo appena oltre il confine di decisione. È più preciso ed efficace di molti attacchi standard, ma computazionalmente più oneroso. In figura 1.5 è riportato un esempio di attacco DF al variare della forza dell'attacco. I parametri principali che regolano il comportamento dell'attacco DF sono:

- *epsilon*: è un parametro di overshoot;
- *nb_grads*: indica il numero di gradienti (cioè classi) che vengono considerati nel calcolo della perturbazione adversarial (siccome all'aumentare di questo parametro aumenta anche il tempo di calcolo necessario per l'attacco, nell'ambito di questo progetto non è stato possibile esplorare valori troppo grandi);
- *max_iter*: rappresenta il numero massimo di iterazioni.



Figure 1.5: Esempio di attacco DF al variare della forza dell’attacco.

- **Carlini-Wagner (CW):** un attacco che mira a generare esempi avversari con la minima distorsione possibile, pur garantendo la misclassificazione. È più preciso ed efficace di molti attacchi standard, ma computazionalmente più oneroso. Dato il vincolo sulla norma L_{∞} , è stato utilizzato l’attacco *CarliniLInfMethod*, ottimizzato per minimizzare la norma L_{∞} . In figura 1.6 è riportato un esempio di attacco CW al variare della forza dell’attacco. I parametri principali che regolano il comportamento dell’attacco CW sono:

- *confidence*: indica quanto deve essere ”sicura” la misclassification dell’immagine adver-sarial;
- *learning_rate*: learning rate iniziale dell’algoritmo (più è basso e migliori saranno i risul-tati, ma l’algoritmo convergerà più lentamente);
- *max_iter*: rappresenta il numero massimo di iterazioni;
- *initial_const*: costante iniziale che bilancia il trade-off tra la quantità di perturbazione e il successo dell’attacco (valori più alti rendono più probabile la riuscita dell’attacco, ma possono aumentare la perturbazione). In tutte le prove è stato fissato a 0.1, valore scelto in quanto rappresenta un buon compromesso tra efficacia e contenimento della perturbazione.

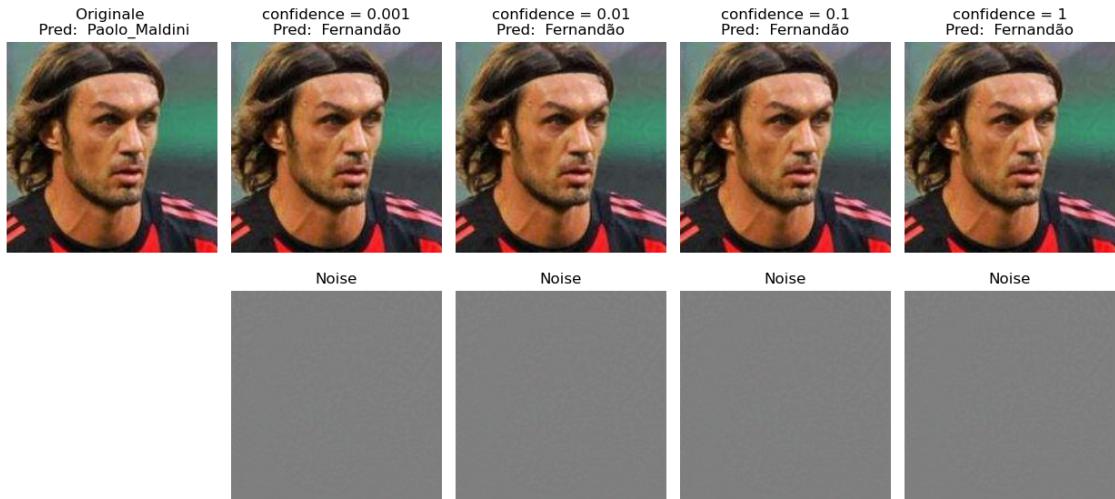


Figure 1.6: Esempio di attacco CW al variare della forza dell’attacco.

Attacchi untargeted e targeted

Al fine di fornire una valutazione completa della robustezza del sistema, sono state considerate sia le varianti **untargeted** che **targeted** di questi attacchi (ad eccezione di DeepFool, per il quale la libreria ART fornisce solo l'implementazione dell'attacco untargeted):

- per gli attacchi *untargeted*, verrà calcolata l'**accuracy** del modello (indica la percentuale di campioni classificati correttamente) al variare dei parametri dell'attacco, calcolata secondo la seguente formula:

$$\text{Accuracy} = \frac{N_{\text{campioni_correttamente_classificati}}}{N_{\text{totale_campioni}}}$$

- per gli attacchi *targeted*, oltre all'accuracy standard, verrà riportata anche la **targeted accuracy** (indica la percentuale di campioni classificati come classe target), definita come:

$$\text{Targeted_accuracy} = \frac{N_{\text{campioni_classificati_come_classe_target}}}{N_{\text{totale_campioni}}}$$

Per tutti i test di attacchi *targeted*, è stata selezionata come classe obiettivo (target) la classe *Cristiano_Ronaldo*.

Security evaluation curves

Inoltre, per ogni parametro di ciascun attacco, sono state tracciate le *security evaluation curves*, le quali offrono una rappresentazione visiva dell'*accuracy* (ed eventualmente della *targeted_accuracy*) del modello al variare dell'intensità dell'attacco. In particolare, per ogni attacco verranno mostrati due grafici distinti:

- **grafico a sinistra:** rappresenta l'accuracy (ed eventualmente della *targeted_accuracy*) del modello (asse *y*) al variare di un parametro specifico dell'attacco (asse *x*), indicato nel grafico stesso. Questo parametro è scelto in modo da simulare la potenza dell'attacco e consente di valutare progressivamente quanto il modello sia sensibile all'aumento della potenza dell'attacco considerato;
- **grafico a destra:** mostra l'accuracy (ed eventualmente della *targeted_accuracy*) del modello (asse *y*) in funzione della perturbazione massima (asse *x*) applicata ai pixels dell'immagine. In questo grafico, verrà tracciata una linea rossa verticale tratteggiata che rappresenta la perturbazione massima accettabile (imposta dal vincolo su Linf): i valori superiori a tale soglia non saranno considerati ammissibili.

In tutte le *security evaluation curves*, il primo punto del grafico rappresenta l'accuracy (ed eventualmente della *targeted_accuracy*) del modello sui dati clean.

1.3.3 Implementare le difese

L'ultimo passo dell'approccio security by design è *implementare le difese*, ovvero sviluppare efficaci contromisure per mitigare o neutralizzare le minacce identificate. A fronte degli attacchi simulati, la strategia di difesa adottata in questo progetto si basa sull'impiego di un sistema di **detectors** di campioni adversarial.

In particolare, sono stati implementati cinque detector, ognuno dei quali specificamente addestrato per riconoscere immagini perturbate da una delle cinque tipologie di attacco considerate (FGSM, BIM, PGD, DF, CW). Ogni detector agisce come un classificatore binario, la cui funzione è distinguere tra immagini "clean" e "adversarial". Per l'addestramento di ciascun detector, è stato utilizzato un dataset bilanciato composto da 1000 campioni clean e 1000 campioni adversarial (generati specificamente per l'attacco corrispondente).

E' stato previsto che un'immagine, prima di essere classificata dalla rete principale NN1, debba superare positivamente il controllo di tutti i detectors. Dunque, un'immagine viene considerata "adversarial" e bloccata se **almeno** uno dei detectors la classifica come tale (viene effettuato un OR logico tra gli output dei detectors), per cui solo le immagini che vengono riconosciute come "clean" da tutti i detectors vengono poi elaborate dalla rete di riconoscimento facciale per la classificazione. D'ora in poi si farà riferimento a questa architettura col nome di "NN1+detectors".

CHAPTER 2

ANALISI DELL'EFFICACIA DEGLI ATTACCHI SULLA RETE NN1

In questo capitolo verranno trattate le informazioni, utili ai fini del progetto, sulla rete NN1 e verrà effettuata un'analisi delle performance della rete, in termini di *accuracy* e *targeted accuracy*, sia su dati clean che su dati adversarial. Quest'ultimi sono stati generati con attacchi *Fast Gradient Sign Method (FGSM)*, *Basic Iterative Method (BIM)*, *Projected Gradient Descent (PGD)*, *DeepFool (DF)* e *Carlini-Wagner (CW)*, appositamente per ingannare la rete NN1.

2.1 Informazioni sulla rete NN1: InceptionResnetV1

La rete neurale NN1 si basa sull'architettura **InceptionResnetV1**, una rete neurale profonda che unisce i punti di forza delle architetture *Inception* e *ResNet*, con l'obiettivo di apprendere rappresentazioni facciali ricche e altamente discriminative.

Il modello utilizzato è stato pre-addestrato sul dataset **VGGFace2**, ed è disponibile pubblicamente al seguente repository GitHub: <https://github.com/timesler/facenet-pytorch>. Da quest'ultimo è possibile ricavare la *loss function* e l'*optimizer* utilizzati in fase di addestramento:

- $loss = \text{torch.nn.CrossEntropyLoss}()$
- $optimizer = \text{Adam}(\text{NN1.parameters}(), lr=0.001)$

Tali configurazioni sono state mantenute anche nella fase di integrazione con il framework ART, al fine di generare attacchi *white-box* in modo coerente con il modello addestrato.

Il modello attende in input immagini con 3 canali (RGB), ognuno dei quali ha un intervallo di rappresentazione $[-1.0, 1.0]$, e restituisce in output un vettore di dimensione 8631 (numero di classi), che corrisponde al numero di identità presenti nel dataset di addestramento. Inoltre, è importante sottolineare che la rete neurale NN1, sebbene sia stata addestrata con immagini di dimensione 160×160 , verrà utilizzata con immagini di dimensione 224×224 . Questa scelta è motivata dalla necessità di valutare la trasferibilità degli attacchi adversariali sulla rete NN2, la quale accetta esclusivamente immagini di dimensione 224×224 (quindi gli attacchi generati sulla rete NN1 devono essere eseguiti su immagini 224×224). Questa variazione introduce una leggera perdita di accuratezza della rete NN1 sulle immagini clean, ma è accettabile ai fini dello studio.

2.2 Prestazioni della rete NN1 sulle immagini clean

Sul test set costituito da immagini clean, il classificatore NN1 ha ottenuto un'**accuracy** pari al **94.4%**, classificando correttamente 944 immagini su un totale di 1000.

La **targeted accuracy** sullo stesso test set è risultata pari allo **0.01%**, con 10 immagini su 1000 classificate come appartenenti alla classe target utilizzata (*Cristiano_Ronaldo*). Tali immagini corrispondono esattamente alle 10 istanze reali associate a quella classe presenti nel test set.

2.3 Prestazioni della rete sulle immagini adversarial FGSM

Per valutare le prestazioni della rete NN1 sulle immagini adversarial generate dall'attacco FGSM, sono state tracciate diverse security evaluation curve. Nello specifico, è stata calcolata una curva per ciascuno dei principali parametri dell'attacco, calcolando sia l'*accuracy* che la *targeted_accuracy* al variare di esso. Questo approccio consente di valutare l'impatto specifico di ciascun parametro dell'attacco sulle prestazioni della rete e di avere un'analisi dettagliata sulla robustezza della rete in funzione della forza dell'attacco. In particolare, verranno tracciati i seguenti plot al variare dei seguenti parametri:

Plot	<i>epsilon</i>
1	[0.01, 0.02, 0.04, 0.06, 0.08, 0.1]

2.3.1 FGSM untargeted

- **Plot 1:** il grafico in Figura 2.1 mostra che, per valori molto bassi di *epsilon* (come 0.01 e 0.02), l'accuracy subisce già un calo rispetto al caso "clean", pur mantenendo in alcuni casi la predizione corretta, mentre all'aumentare di *epsilon*, l'accuracy del modello decresce rapidamente, fino a raggiungere valori prossimi allo zero per $\epsilon \geq 0.06$.

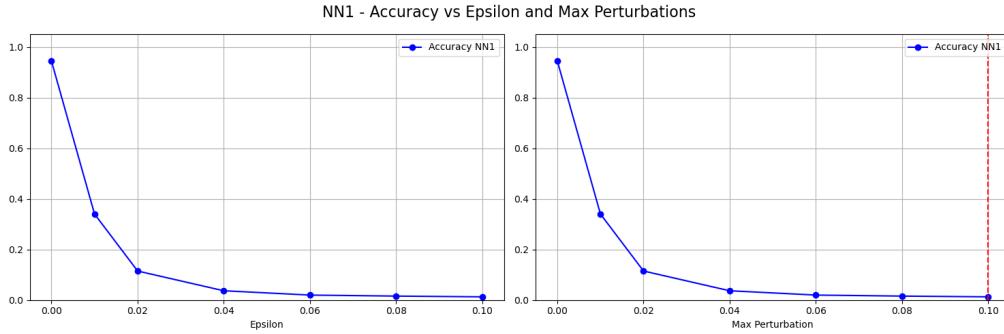


Figure 2.1: NN1 - FGSM untargeted - Plot 1

2.3.2 FGSM targeted

- **Plot 1:** il grafico in Figura 2.2 mostra che, all'aumentare di *epsilon*, l'accuracy cala rapidamente, indicando che il modello commette un numero crescente di errori. Parallelamente, la targeted accuracy aumenta, segnalando che l'attacco sta diventando sempre più efficace nel forzare l'output verso la classe target. Tuttavia, la targeted accuracy non raggiunge mai valori molto alti (il valore migliore osservato è circa 0.45), a causa della natura one-shot dell'attacco FGSM, che presenta limiti intrinseci nella capacità di condurre la predizione verso una classe specifica in un solo passo.

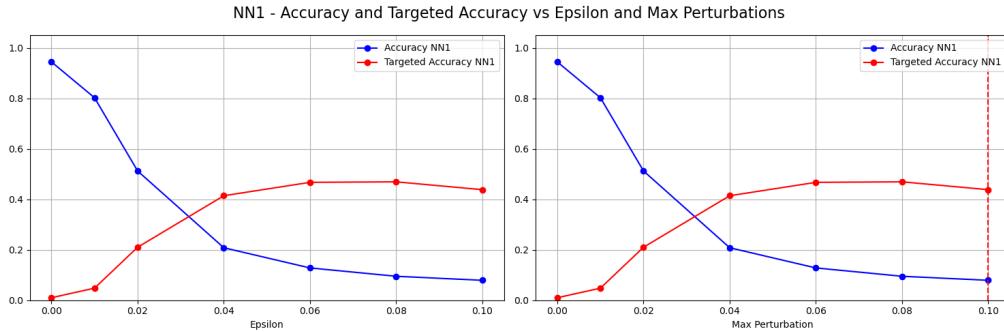


Figure 2.2: NN1 - FGSM targeted - Plot 1

2.4 Prestazioni della rete sulle immagini adversarial BIM

Per valutare le prestazioni della rete NN1 sulle immagini adversarial generate dall'attacco BIM, sono state tracciate diverse security evaluation curve. Nello specifico, è stata calcolata una curva per ciascuno dei principali parametri dell'attacco, calcolando sia l'accuracy che la targeted_accuracy al variare di esso. Questo approccio consente di valutare l'impatto specifico di ciascun parametro dell'attacco sulle prestazioni della rete e di avere un'analisi dettagliata sulla robustezza della rete in funzione della forza dell'attacco. In particolare, verranno tracciati i seguenti plot al variare dei seguenti parametri:

Plot	<i>epsilon</i>	<i>epsilon_step_value</i>	<i>max_iter</i>
1	[0.01, 0.02, 0.04, 0.06, 0.08, 0.1]	0.01	10
2	0.1	[0.01, 0.02, 0.03, 0.04, 0.05]	10
3	0.1	0.01	[1, 3, 5, 7, 10]

2.4.1 BIM untargeted

- **Plot 1:** il grafico in Figura 2.3 mostra che l'attacco è altamente efficace anche con valori di *epsilon* molto piccoli. Infatti, già con *epsilon*=0.02, l'accuracy crolla a zero. Questo indica una bassa robustezza del modello agli attacchi BIM con questa configurazione.

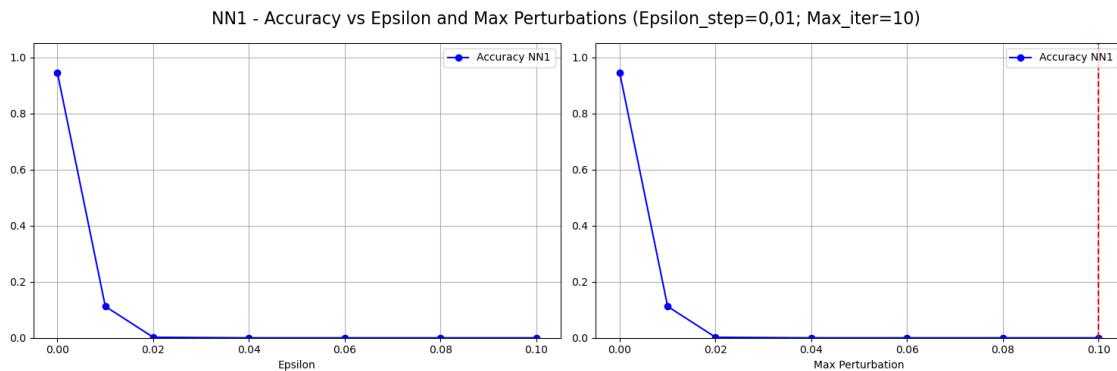


Figure 2.3: NN1 - BIM untargeted - Plot 1

- **Plot 2:** il grafico in Figura 2.4 mostra che anche un *epsilon_step* piccolo (come *epsilon_step*=0.01) è già sufficiente a compromettere la classificazione del modello, grazie al fatto che viene utilizzato un numero di iterazioni (10) sufficiente a far arrivare la perturbazione massima al limite consentito.

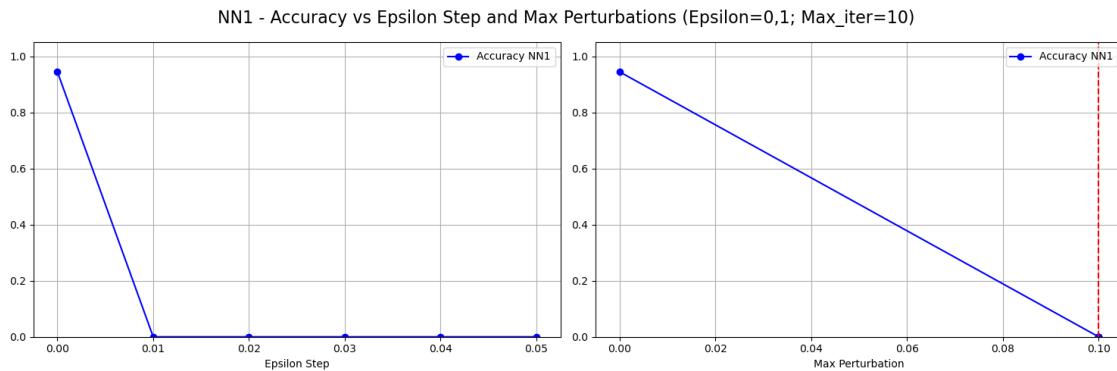


Figure 2.4: NN1 - BIM untargeted - Plot 2

- **Plot 3:** il grafico in Figura 2.5 mostra che anche un numero di iterazioni basso (come $max_iter= 3$) è già in grado di abbattere completamente l'accuratezza del modello. Tuttavia, usando $epsilon_step= 0.01$, è preferibile usare $max_iter= 10$, in maniera tale da sfruttare la massima perturbazione consentita (0.1).

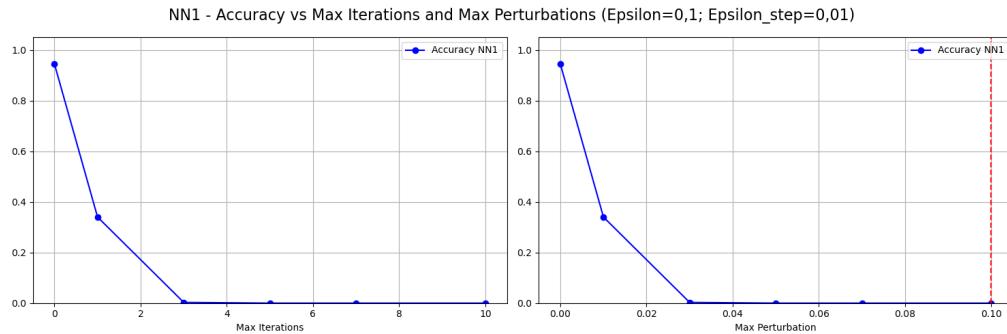


Figure 2.5: NN1 - BIM untargeted - Plot 3

2.4.2 BIM targeted

- **Plot 1:** il grafico in Figura 2.6 mostra che il modello è altamente vulnerabile all'attacco targeted. Infatti, anche una perturbazione minima (come $epsilon= 0.04$) è sufficiente per far sì che il modello assegna sistematicamente l'input alla classe target.

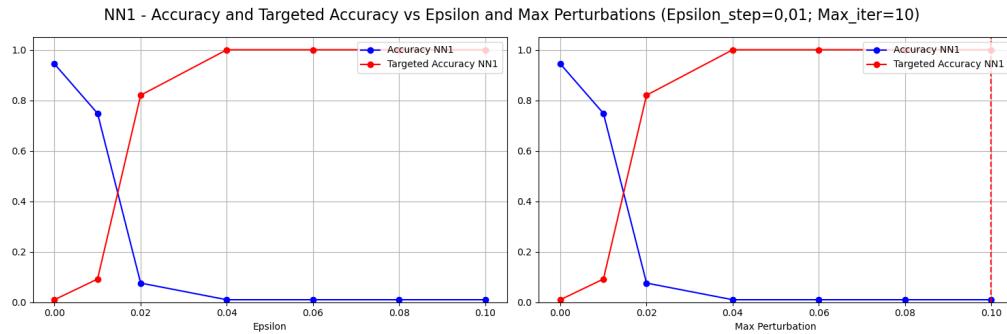


Figure 2.6: NN1 - BIM targeted - Plot 1

- **Plot 2:** il grafico in Figura 2.7 mostra che anche un $epsilon_step$ piccolo (come $epsilon_step= 0.01$) è già sufficiente a compromettere la classificazione del modello, grazie al fatto che viene utilizzato un numero di iterazioni (10) sufficiente a far arrivare la perturbazione massima al limite consentito.

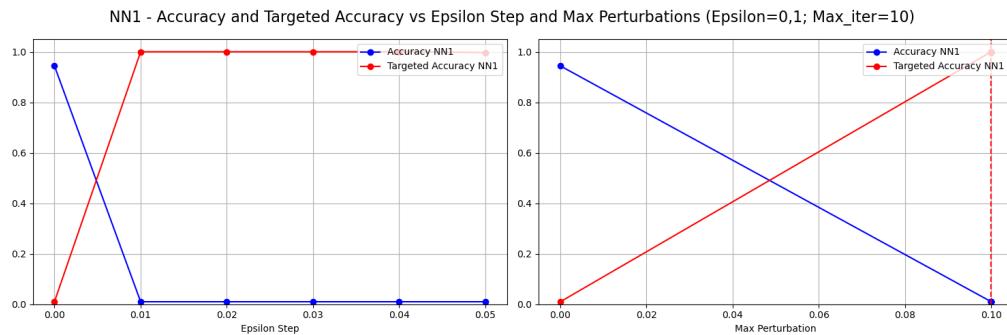


Figure 2.7: NN1 - BIM targeted - Plot 2

- **Plot 3:** il grafico in Figura 2.8 mostra che anche un numero di iterazioni basso (come `max_iter= 5`) è già sufficiente per far sì che il modello assegna sistematicamente l'input alla classe target. Tuttavia, usando `epsilon_step= 0.01`, è preferibile usare `max_iter= 10`, in maniera tale da sfruttare la massima perturbazione consentita (0.1).

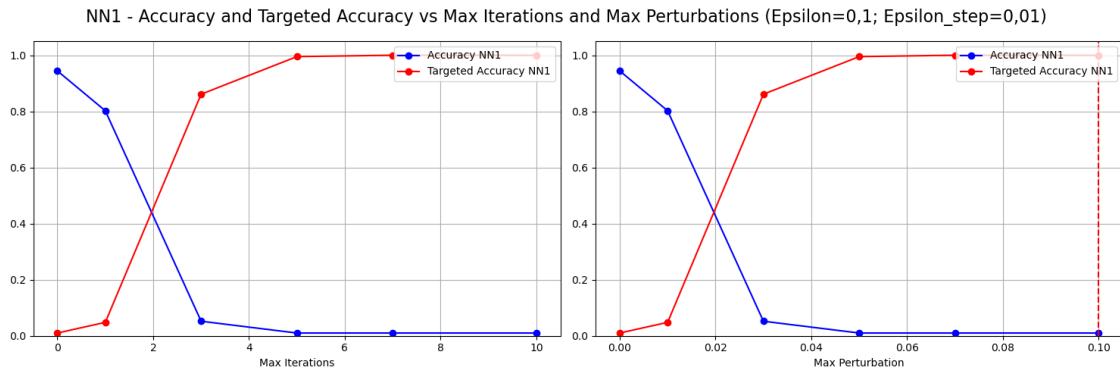


Figure 2.8: NN1 - BIM targeted - Plot 3

2.5 Prestazioni della rete sulle immagini adversarial PGD

Per valutare le prestazioni della rete NN1 sulle immagini adversarial generate dall'attacco PGD, sono state tracciate diverse security evaluation curve. Nello specifico, è stata calcolata una curva per ciascuno dei principali parametri dell'attacco, calcolando sia l'*accuracy* che la *targeted_accuracy* al variare di esso. Questo approccio consente di valutare l'impatto specifico di ciascun parametro dell'attacco sulle prestazioni della rete e di avere un'analisi dettagliata sulla robustezza della rete in funzione della forza dell'attacco. In particolare, verranno tracciati i seguenti plot al variare dei seguenti parametri:

Plot	<i>epsilon</i>	<i>epsilon_step_value</i>	<i>max_iter</i>
1	[0.01, 0.02, 0.04, 0.06, 0.08, 0.1]	0.01	10
2	0.1	[0.01, 0.02, 0.03, 0.04, 0.05]	10
3	0.1	0.01	[1, 3, 5, 7, 10]

2.5.1 PGD untargeted

- **Plot 1:** il grafico in Figura 2.9 mostra che l'attacco è altamente efficace anche con valori di *epsilon* molto piccoli. Infatti, già con *epsilon*=0.02, l'accuratezza crolla a zero. Questo indica una bassa robustezza del modello agli attacchi BIM con questa configurazione.

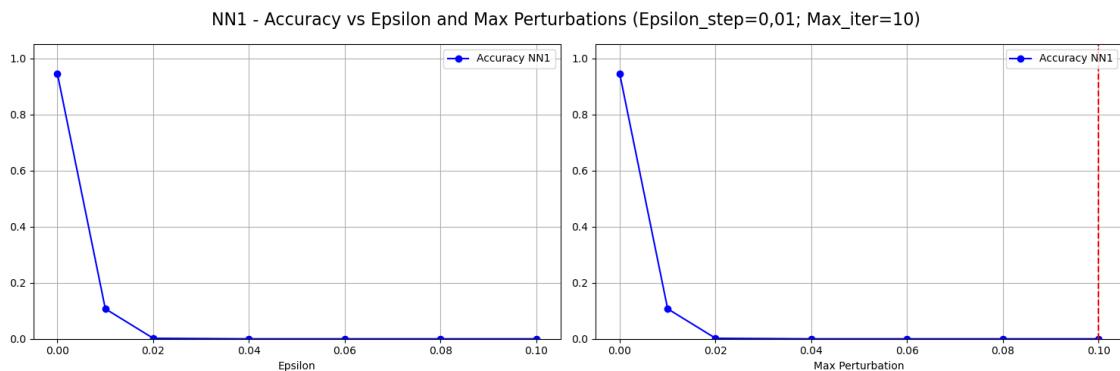


Figure 2.9: NN1 - PGD untargeted - Plot 1

- **Plot 2:** il grafico in Figura 2.10 mostra che anche un *epsilon_step* piccolo (come *epsilon_step*=0.01) è già sufficiente a compromettere la classificazione del modello, grazie al fatto che viene utilizzato un numero di iterazioni (10) sufficiente a far arrivare la perturbazione massima al limite consentito.

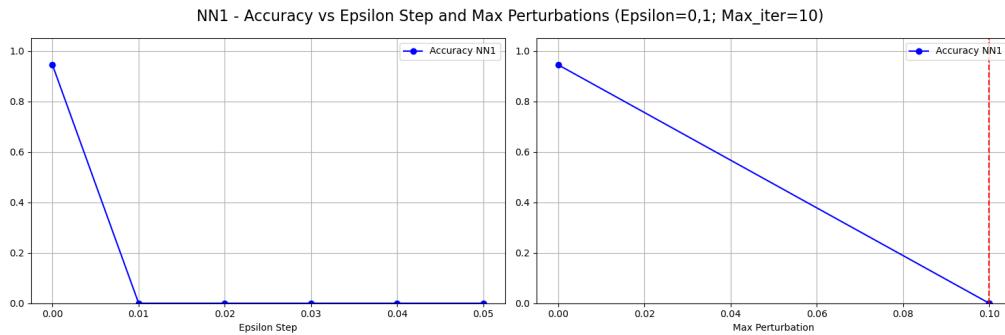


Figure 2.10: NN1 - PGD untargeted - Plot 2

- **Plot 3:** il grafico in Figura 2.11 mostra che anche un numero di iterazioni basso (come *max_iter*=3) è già in grado di abbattere completamente l'accuratezza del modello. Tuttavia, usando *epsilon_step*=0.01, è preferibile usare *max_iter*= 10, in maniera tale da sfruttare la massima perturbazione consentita (0.1).

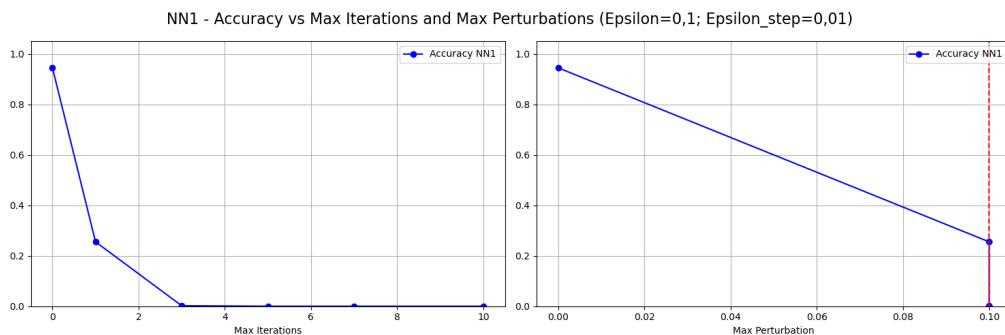


Figure 2.11: NN1 - PGD untargeted - Plot 3

2.5.2 PGD targeted

- **Plot 1:** il grafico in Figura 2.6 mostra che il modello è altamente vulnerabile all'attacco targeted. Infatti, anche una perturbazione minima (come *epsilon*= 0.04) è sufficiente per far sì che il modello assegni sistematicamente l'input alla classe target.

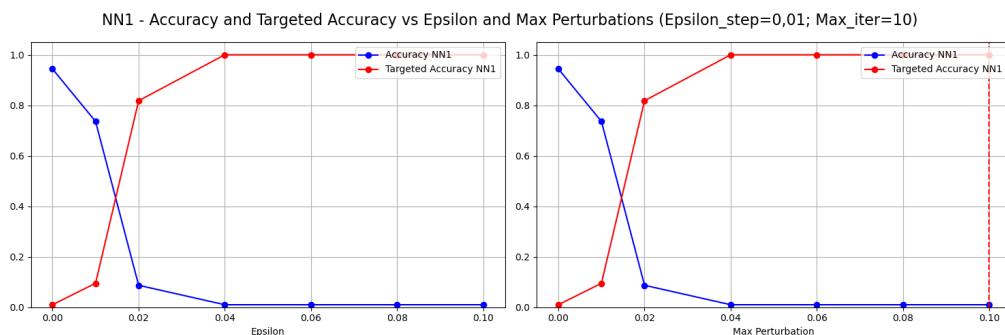


Figure 2.12: NN1 - PGD targeted - Plot 1

- **Plot 2:** il grafico in Figura 2.13 mostra che anche un *epsilon_step* piccolo (come *epsilon_step*=0.01) è già sufficiente a compromettere la classificazione del modello, grazie al fatto che viene utilizzato un numero di iterazioni (10) sufficiente a far arrivare la perturbazione massima al limite consentito.

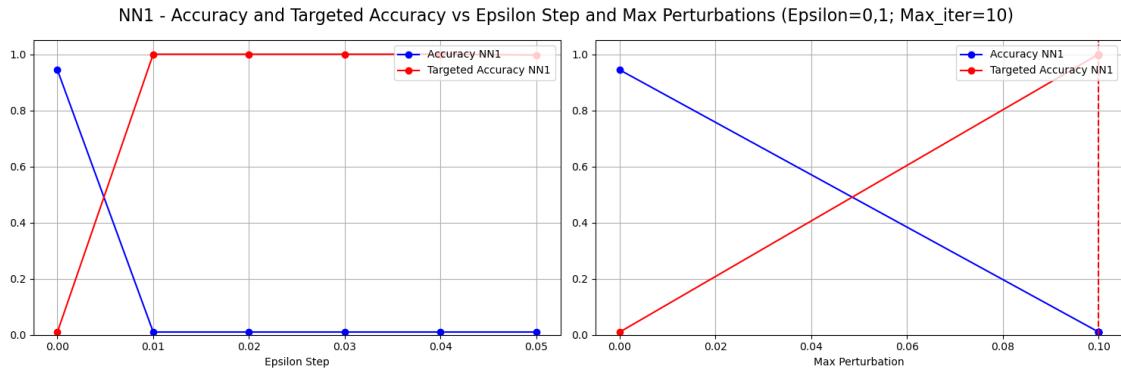


Figure 2.13: NN1 - PGD targeted - Plot 2

- **Plot 3:** il grafico in Figura 2.14 mostra che anche un numero di iterazioni basso (come *max_iter*= 5) è già sufficiente per far sì che il modello assegna sistematicamente l'input alla classe target. Tuttavia, usando *epsilon_step*= 0.01, è preferibile usare *max_iter*= 10, in maniera tale da sfruttare la massima perturbazione consentita (0.1).

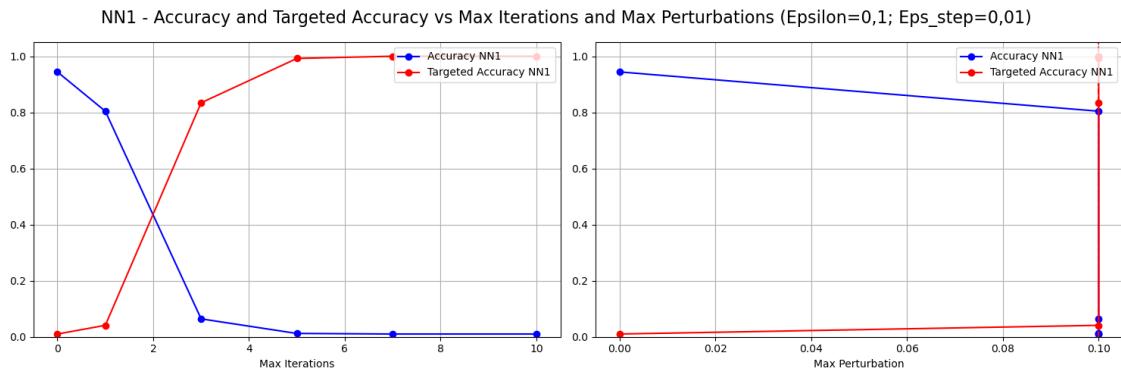


Figure 2.14: NN1 - PGD targeted - Plot 3

2.6 Prestazioni della rete sulle immagini adversarial DF

Per valutare le prestazioni della rete NN1 sulle immagini adversarial generate dall'attacco DF, sono state tracciate diverse security evaluation curve. Nello specifico, è stata calcolata una curva per ciascuno dei principali parametri dell'attacco, calcolando l'*accuracy* al variare di esso. Questo approccio consente di valutare l'impatto specifico di ciascun parametro dell'attacco sulle prestazioni della rete e di avere un'analisi dettagliata sulla robustezza della rete in funzione della forza dell'attacco. In particolare, verranno tracciati i seguenti plot al variare dei seguenti parametri:

Plot	<i>epsilon</i>	<i>nb_grads</i>	<i>max_iter</i>
1	[$1^{-5}, 1^{-4}, 1^{-3}, 1^{-2}, 1^{-1}, 1$]	10	10
2	1^{-2}	[5, 10, 20, 50]	10
3	1^{-2}	10	[1, 3, 5, 7, 10]

2.6.1 DF untargeted

- **Plot 1:** il grafico in Figura 2.15 mostra che l'attacco risulta essere estremamente efficace per ogni valore di ϵ considerato, riuscendo a portare l'accuracy del modello a zero. Tuttavia, all'aumentare di ϵ , cresce anche la perturbazione introdotta in termini di norma Linf. Poiché l'attacco non è progettato per ottimizzare direttamente questa metrica, non è mai stato possibile rientrare nei vincoli di progetto.

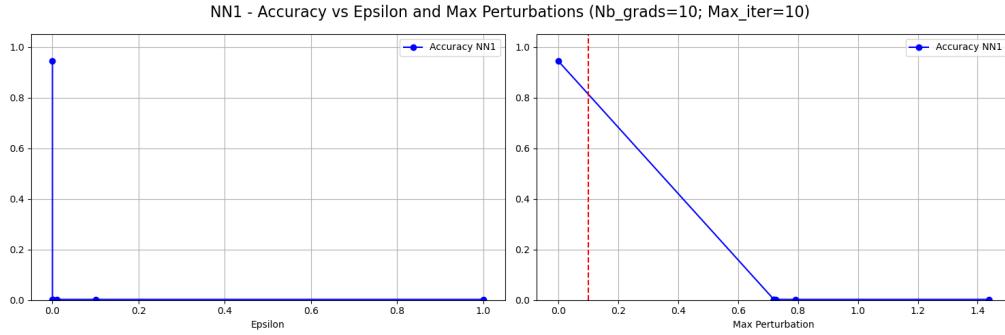


Figure 2.15: NN1 - DF untargeted - Plot 1

- **Plot 2:** il grafico in Figura 2.16 mostra che l'attacco riesce a portare l'accuracy del modello a zero per la maggior parte dei valori di nb_grads considerati, confermando l'efficacia. Si osserva inoltre che, all'aumentare di nb_grads , la perturbazione Linf tende a diminuire, suggerendo che il metodo riesce a trovare soluzioni sempre più ottimali. Tuttavia, anche in questo caso, i valori ottenuti non rientrano nei vincoli del progetto.

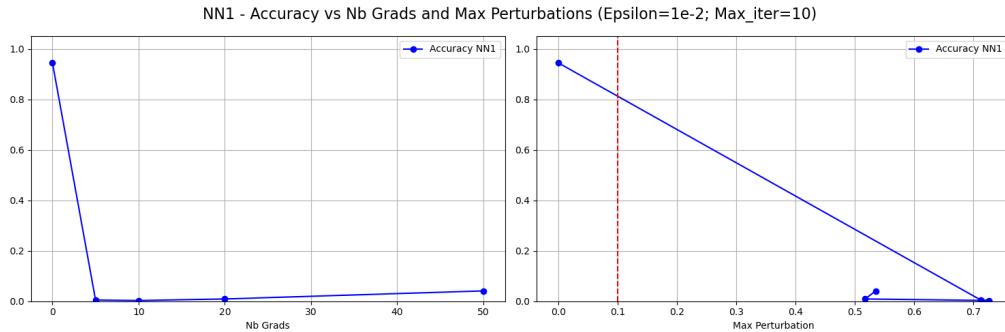


Figure 2.16: NN1 - DF untargeted - Plot 2

- **Plot 3:** il grafico in Figura 2.17 mostra che anche un numero di iterazioni basso (come $max_iter=3$) è già in grado di abbattere completamente l'accuratezza del modello. Anche in questo caso, i valori ottenuti non rientrano nei vincoli del progetto.

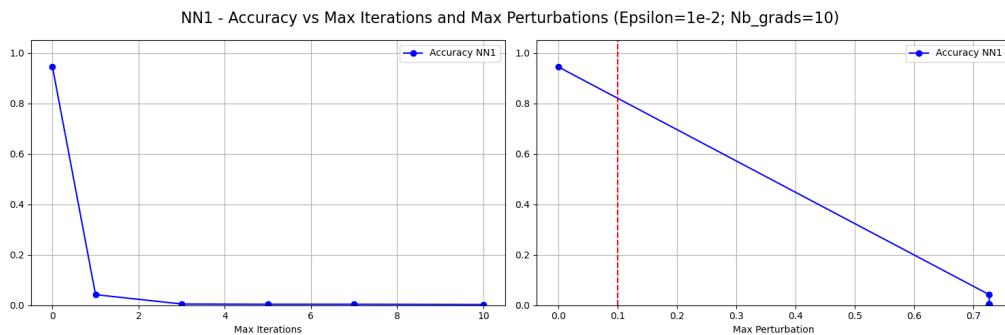


Figure 2.17: NN1 - DF untargeted - Plot 3

Considerazioni sul mancato rispetto dei vincoli

L'attacco *DeepFool* si è dimostrato altamente efficace in termini di riduzione dell'*accuracy* del modello, raggiungendo valori prossimi allo zero in tutte le configurazioni di parametri analizzate. Tuttavia, siccome l'attacco non ottimizza la riduzione della norma Linf , nessuna combinazione testata ha consentito di rientrare nei vincoli del progetto. Si conclude quindi che, nelle condizioni progettuali considerate, **l'attacco DeepFool non è utilizzabile**.

Tuttavia, gli istogrammi della distribuzione delle perturbazioni massime dell'attacco DF (un esempio è mostrato in Figura 2.18) evidenziano che una porzione non trascurabile di immagini presenta perturbazioni inferiori al vincolo progettuale imposto. Questo suggerisce che, sebbene l'attacco non rispetti globalmente i vincoli, esistono comunque casi in cui esso risulta conforme. Pertanto, si è ritenuto opportuno considerare questo attacco anche nelle sezioni successive del progetto.

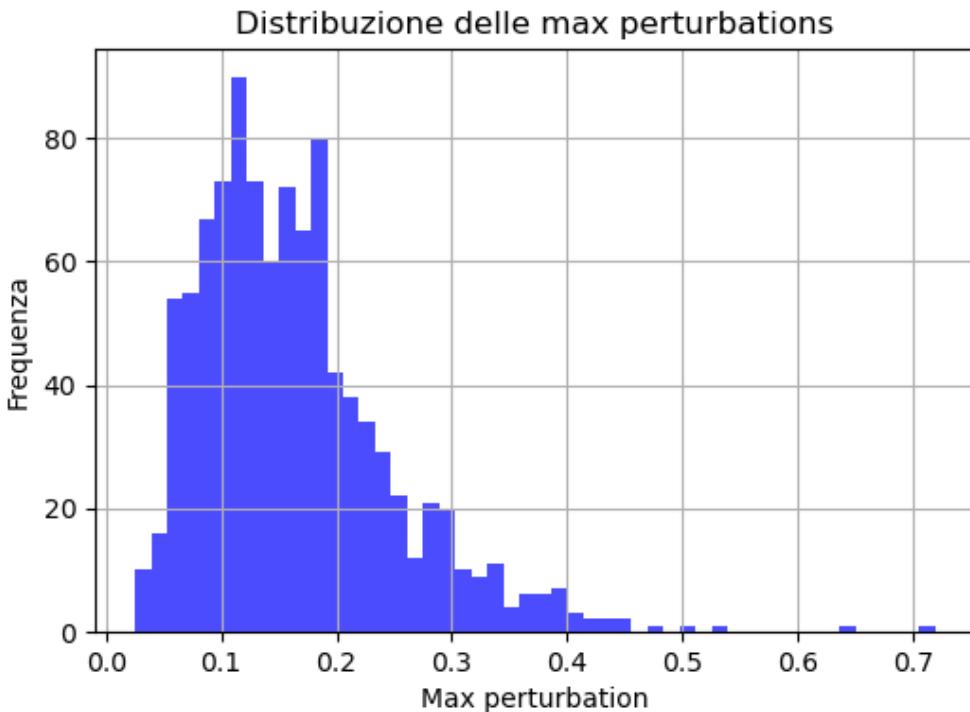


Figure 2.18: Esempio di istogramma delle perturbazioni massime dell'attacco DF.

2.7 Prestazioni della rete sulle immagini adversarial CW

Per valutare le prestazioni della rete NN1 sulle immagini adversarial generate dall'attacco CW, sono state tracciate diverse security evaluation curve. Nello specifico, è stata calcolata una curva per ciascuno dei principali parametri dell'attacco, calcolando sia l'*accuracy* che la *targeted_accuracy* al variare di esso. Questo approccio consente di valutare l'impatto specifico di ciascun parametro dell'attacco sulle prestazioni della rete e di avere un'analisi dettagliata sulla robustezza della rete in funzione della forza dell'attacco. In particolare, verranno tracciati i seguenti plot al variare dei seguenti parametri:

Plot	confidence	learning_rate	max_iter
1	[0.01, 0.1, 1]	0.01	3
2	0.1	[0.01, 0.05, 0.1]	3
3	0.1	0.01	[1, 3, 5]

2.7.1 CW untargeted

- **Plot 1:** il grafico in Figura 2.19 mostra come l'attacco proposto sia estremamente efficace, portando l'accuratezza del modello a zero per tutti i valori di *confidence* testati. Le perturbazioni introdotte rispettano i vincoli imposti sulla perturbazione massima.

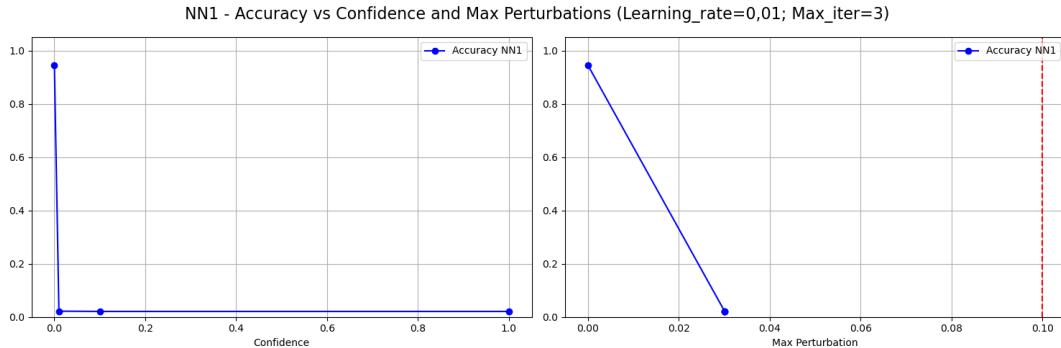


Figure 2.19: NN1 - CW untargeted - Plot 1

- **Plot 2:** il grafico in Figura 2.20 mostra come l'attacco proposto sia estremamente efficace, portando l'accuratezza del modello a zero per tutti i valori di *learning_rate* testati. E' possibile notare un incremento della perturbazione massima all'aumentare del *learning_rate*, rimanendo comunque nei vincoli imposti.

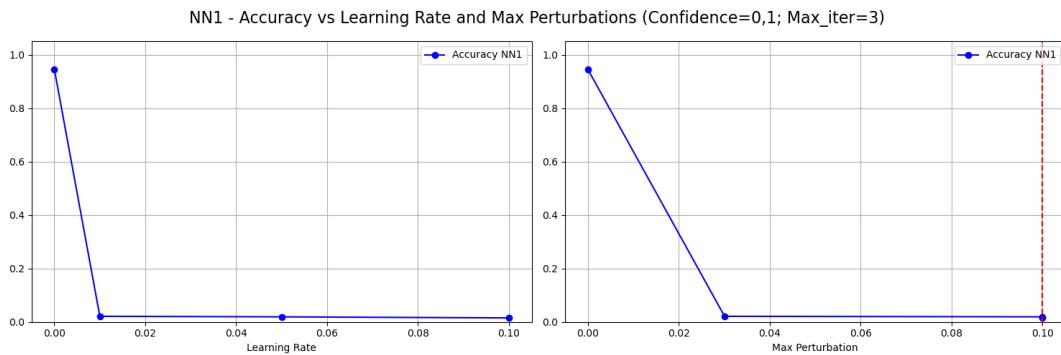


Figure 2.20: NN1 - CW untargeted - Plot 2

- **Plot 3:** il grafico in Figura 2.21 mostra che anche un numero di iterazioni basso (come *max_iter*= 3) è già in grado di abbattere completamente l'accuratezza del modello. Le perturbazioni introdotte rispettano i vincoli imposti sulla perturbazione massima.

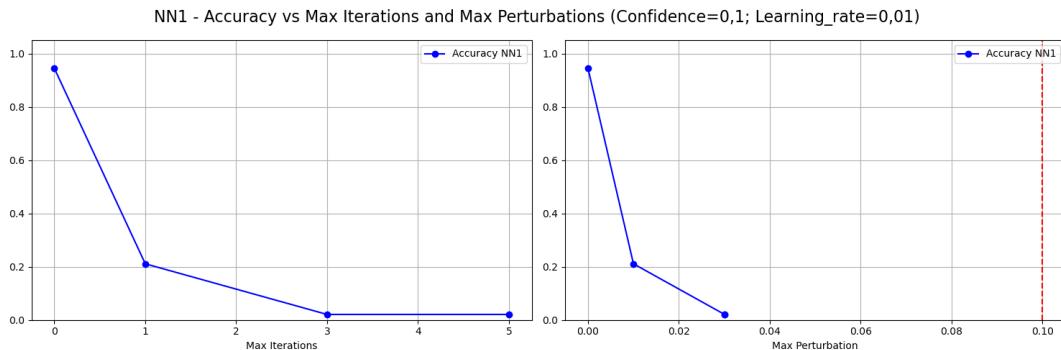


Figure 2.21: NN1 - CW untargeted - Plot 3

2.7.2 CW targeted

- **Plot 1:** il grafico in Figura 2.22 mostra che l'efficacia dell'attacco *targeted* è molto limitata per tutti i valori di *confidence* testati: l'*accuracy* del modello non decresce oltre 0.4, mentre la *targeted_accuracy* resta praticamente invariata a 0.01, segnalando l'incapacità dell'attacco di convertire correttamente le immagini verso la classe obiettivo. Le perturbazioni introdotte rispettano i vincoli imposti sulla perturbazione massima.

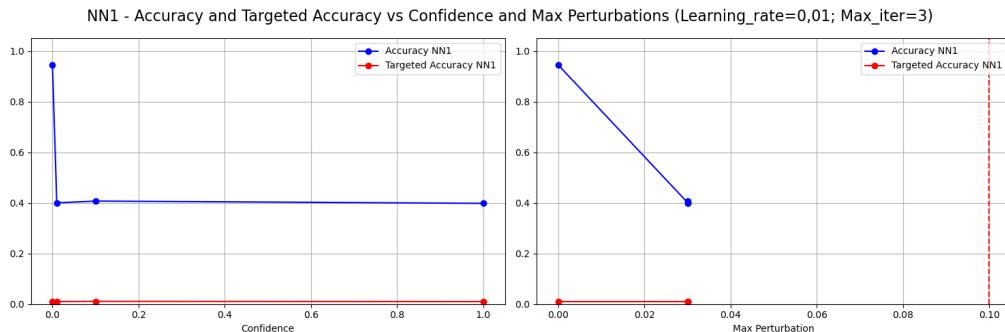


Figure 2.22: NN1 - CW targeted - Plot 1

- **Plot 2:** il grafico in Figura 2.23 mostra che l'efficacia dell'attacco *targeted* è molto limitata anche al variare del *learning_rate*. Inoltre, all'aumentare di quest'ultimo, si osserva un aumento della perturbazione massima (che non rispetta i vincoli imposti) e dell'*accuracy* (indice di un peggioramento delle prestazioni dell'attacco). Questo suggerisce l'utilizzo di un valore basso per il *learning_rate*.

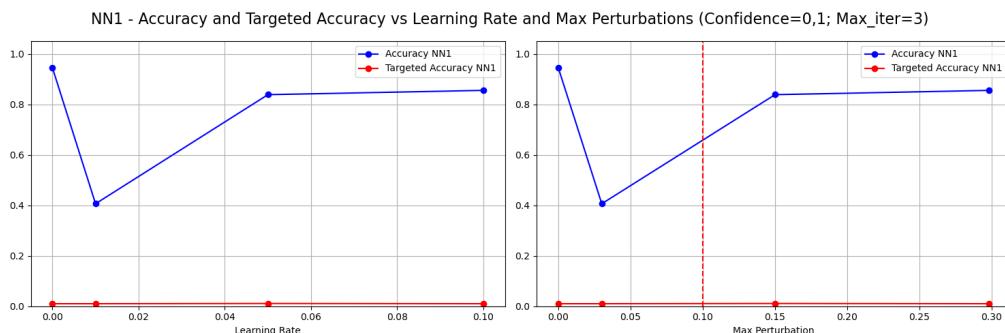


Figure 2.23: NN1 - CW targeted - Plot 2

- **Plot 3:** il grafico in Figura 2.24 mostra che l'efficacia dell'attacco *targeted* è molto limitata anche al variare di *max_iter*. Dal grafico si evince che il valore ottimale è *max_iter*= 3. Le perturbazioni introdotte rispettano i vincoli imposti sulla perturbazione massima.

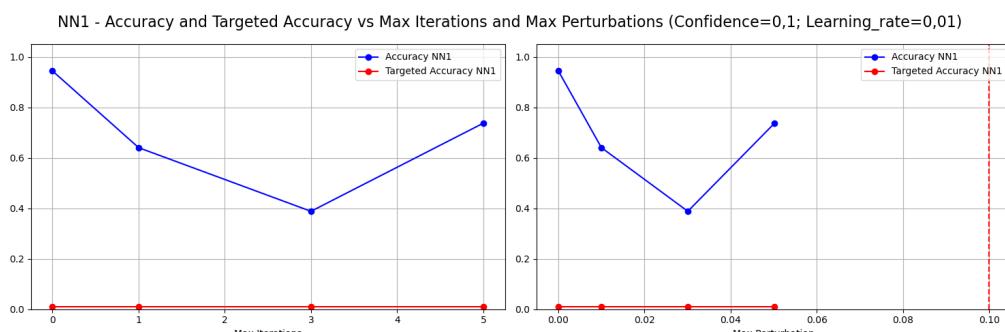


Figure 2.24: NN1 - CW targeted - Plot 3

Considerazioni sulla mancata efficacia dell'attacco target

L'attacco *Carlini-Wagner* nella sua variante *targeted* si è dimostrato significativamente meno efficace rispetto alla controparte *untargeted*. In nessuna configurazione testata è stato possibile ottenere un aumento apprezzabile della *targeted_accuracy*, per cui l'attacco non riesce a raggiungere il suo obiettivo principale, ovvero forzare la classificazione verso una classe specifica. Si conclude pertanto che, nelle condizioni imposte dal progetto, l'attacco **Carlini-Wagner targeted non è utilizzabile** in modo efficace, motivo per il quale si è ritenuto opportuno non considerare questo attacco nelle sezioni successive del progetto.

2.8 Riepilogo sull'efficacia degli attacchi sulla rete NN1

In questa sezione viene presentato un breve riepilogo sull'efficacia degli attacchi sulla rete NN1. Per ciascun tipo di attacco viene riportata l'**accuracy minima** (ed eventualmente la **targeted accuracy massima**) ottenuta, insieme alla **relativa perturbazione massima** osservata (nel caso in cui il valore massimo dell'accuracy sia stato ottenuto più volte con perturbazioni massime diverse, verrà considerata la perturbazione massima più piccola).

Attacco	Accuracy _{min} (\approx)	Targeted Accuracy _{max} (\approx)	L _{inf} (\approx)
FGSM (untargeted)	0.0	/	0.06
FGSM (targeted)	0.1	0.45	0.08
BIM (untargeted)	0.0	/	0.02
BIM (targeted)	0.0	1.0	0.04
PGD (untargeted)	0.0	/	0.02
PGD (targeted)	0.0	1.0	0.04
DF (untargeted)	0.0	/	0.70
CW (untargeted)	0.0	/	0.03
CW (targeted)	0.4	0.01	0.03

Dall'analisi dei risultati riportati in tabella emerge che gli attacchi *untargeted* sono generalmente più efficaci rispetto a quelli *targeted* nel ridurre l'accuratezza del modello NN1. Infatti, entrambi i tipi di attacco riescono in alcuni casi ad azzerare l'*accuracy*, ma gli attacchi *untargeted* lo fanno con perturbazioni massime inferiori. In particolare, BIM, PGD e CW nella modalità *untargeted* si distinguono per la loro elevata efficacia, raggiungendo un'accuracy minima pari a circa 0.0 con perturbazioni contenute. Infine, per quanto riguarda gli attacchi *targeted*, solo BIM e PGD riescono a ottenere una *targeted accuracy* massima pari a circa 1.0, risultando i più efficaci all'interno di questa categoria.

CHAPTER 3

ANALISI DELLA TRASFERIBILITÀ DEGLI ATTACCHI

In questo capitolo verranno trattate le informazioni, utili ai fini del progetto, sulla rete NN2 e verrà effettuata un'analisi delle performance della rete, in termini di *accuracy* e *targeted accuracy*, sia su dati clean che su dati adversarial. Quest'ultimi sono stati generati con attacchi *Fast Gradient Sign Method (FGSM)*, *Basic Iterative Method (BIM)*, *Projected Gradient Descent (PGD)*, *DeepFool (DF)* e *Carlini-Wagner (CW)*, per ingannare la rete NN1.

Dunque, l'obiettivo di questo capitolo è valutare la trasferibilità degli attacchi generati sulla rete NN1. Per farlo, verranno confrontate le performance delle rete NN2 con le performance della rete NN1. Infatti, per trasferibilità si intende la capacità di un'immagine avversaria, generata per ingannare una specifica rete neurale (in questo caso NN1), di compromettere anche le prestazioni di un'altra rete (in questo caso NN2) addestrata per lo stesso compito.

3.1 Informazioni sulla rete NN2: SENet50

La rete neurale NN2 è basata sull'architettura **Squeeze-and-Excitation Network 50 (SENet50)**, una variante del modello *ResNet50* in cui vengono integrati i blocchi *Squeeze-and-Excitation (SE)*, progettati per migliorare la rappresentazione dei canali di feature attraverso un meccanismo di attenzione.

Il modello utilizzato è stato pre-addestrato sul dataset **VGGFace2**, ed è disponibile pubblicamente al seguente repository GitHub: <https://github.com/cydonia999/VGGFace2-pytorch>.

Il modello attende in input immagini con risoluzione 224x224 e con 3 canali (BGR), e restituisce in output un vettore di dimensione 8631 (numero di classi), che corrisponde al numero di identità presenti nel dataset di addestramento.

3.2 Preprocessing delle immagini

Al fine di valutare la trasferibilità degli attacchi, generati dalla rete NN1, sulla rete NN2, è necessario adattare le immagini avversarie prodotte al tipo di immagini richieste in input dalla rete NN2. Infatti, le due reti utilizzano convenzioni differenti nella rappresentazione delle immagini in input: la rete NN1 richiede immagini nell'intervallo di rappresentazione $[-1.0, 1.0]$ in formato RGB, mentre la rete NN2 richiede immagini nell'intervallo di rappresentazione $[0.0, 255.0]$ in formato *BGR*, che vengono poi normalizzate rispetto alla media del dataset VGGFace2 (come si evince dal repository GitHub di riferimento).

Per questo motivo, prima di fornire le immagini avversarie in input alla rete NN2, è necessario applicare le seguenti trasformazioni di preprocessing (tali operazioni non modificano il contenuto semantico delle immagini, né alterano le perturbazioni avversarie apprese durante l'ottimizzazione, ma rappresentano semplicemente un adattamento formale ai requisiti architettonici della rete NN2):

1. **Conversione dell'intervallo di rappresentazione:** le immagini vengono convertite dall'intervallo $[-1.0, 1.0]$ all'intervallo $[0.0, 255.0]$ attraverso la seguente trasformazione:

$$\text{image} = (\text{image} + 1.0) \times \frac{255.0}{2}$$

2. **Conversione dal formato:** le immagini vengono convertite dal formato *RGB* al formato *BGR* attraverso la seguente trasformazione:

$$\text{image}_{[0,1,2]} = \text{image}_{[2,1,0]}$$

3. **Normalizzazione dei valori su ciascun canale:** le immagini vengono normalizzate sottraendo, da ciascun canale, il relativo valore medio calcolato sul dataset VGGFace2:

$$\mu_{\text{BGR}} = \begin{bmatrix} 91.4953 \\ 103.8827 \\ 131.0912 \end{bmatrix}$$

Al termine di questa sequenza di operazioni, le immagini risultano essere pienamente compatibili con la rete NN2 e possono essere utilizzate per testare la trasferibilità degli attacchi generati su NN1. È importante sottolineare che questo processo di adattamento non introduce alcuna perdita informativa, né degrada le caratteristiche avversarie delle immagini. Le trasformazioni effettuate sono reversibili e agiscono esclusivamente sulla forma rappresentazionale dei dati.

3.3 Prestazioni della rete NN2 sulle immagini clean

Sul test set costituito da immagini clean, il classificatore NN2 ha ottenuto un'**accuracy** pari al **91.6%**, classificando correttamente 916 immagini su un totale di 1000.

La **targeted accuracy** sullo stesso test set è risultata pari allo **0.01%**, con 10 immagini su 1000 classificate come appartenenti alla classe target utilizzata (*Cristiano_Ronaldo*). Tali immagini corrispondono esattamente alle 10 istanze reali associate a quella classe presenti nel test set.

3.4 Prestazioni della rete sulle immagini adversarial FGSM

Per valutare le prestazioni della rete NN2 sulle immagini adversarial generate dall'attacco FGSM, sono state tracciate diverse security evaluation curve. Nello specifico, è stata calcolata una curva per ciascuno dei principali parametri dell'attacco, calcolando sia l'*accuracy* che la *targeted_accuracy* al variare di esso. Questo approccio consente di valutare l'impatto specifico di ciascun parametro dell'attacco sulle prestazioni della rete e di avere un'analisi dettagliata sulla robustezza della rete in funzione della forza dell'attacco. In particolare, verranno tracciati i seguenti plot al variare dei seguenti parametri:

Plot	<i>epsilon</i>
1	[0.01, 0.02, 0.04, 0.06, 0.08, 0.1]

3.4.1 FGSM untargeted

- **Plot 1:** il grafico in Figura 3.1 mostra la trasferibilità dell'attacco FGSM untargeted, al variare del parametro *epsilon*: come atteso, l'accuracy del classificatore NN2 è soggetta a una decrescita minore rispetto a quella del classificatore NN1. Tuttavia, l'attacco riesce comunque a ridurre l'accuratezza fino a circa 0.35 (per *epsilon*= 0.1), dimostrando una certa efficacia anche nei confronti della rete NN2. Dunque, è possibile affermare che l'attacco gode di una trasferibilità di medio livello.

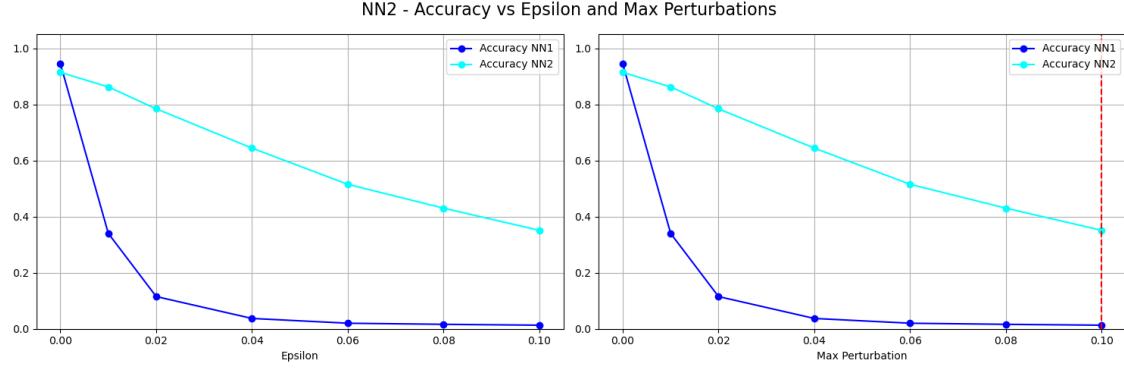


Figure 3.1: NN2 - FGSM untargeted - Plot 1

3.4.2 FGSM targeted

- **Plot 1:** il grafico in Figura 3.2 mostra la trasferibilità dell'attacco FGSM targeted, al variare del parametro *epsilon*: come atteso, l'accuracy del classificatore NN2 è soggetta a una decrescita minore rispetto a quella del classificatore NN1, ma l'attacco riesce comunque a ridurre l'accuratezza fino a circa 0.55 (per *epsilon*= 0.1). Tuttavia, non è possibile notare un aumento significativo della *targeted_accuracy* della rete NN2, indice dell'incapacità dell'attacco di convertire correttamente le immagini verso la classe target. Dunque, è possibile affermare che l'attacco non è trasferibile.

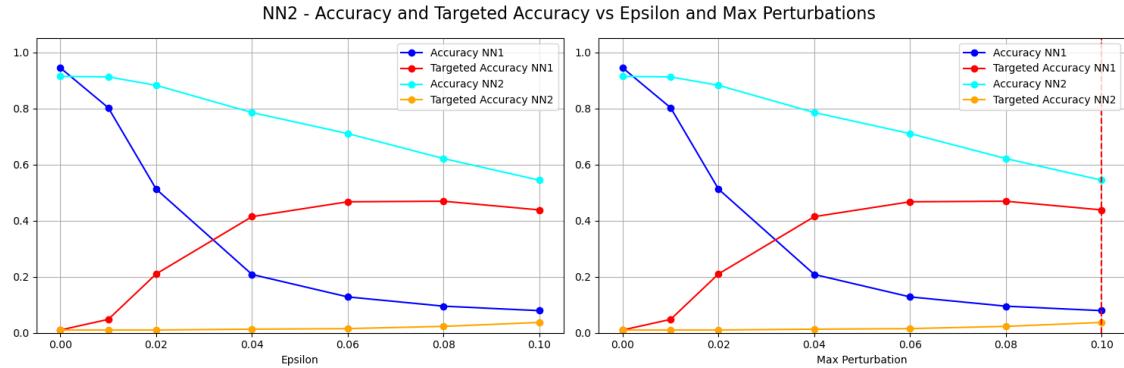


Figure 3.2: NN2 - FGSM targeted - Plot 1

3.5 Prestazioni della rete sulle immagini adversarial BIM

Per valutare le prestazioni della rete NN2 sulle immagini adversarial generate dall'attacco BIM, sono state tracciate diverse security evaluation curve. Nello specifico, è stata calcolata una curva per ciascuno dei principali parametri dell'attacco, calcolando sia l'*accuracy* che la *targeted_accuracy* al variare di esso. Questo approccio consente di valutare l'impatto specifico di ciascun parametro dell'attacco sulle prestazioni della rete e di avere un'analisi dettagliata sulla robustezza della rete in funzione della forza dell'attacco. In particolare, verranno tracciati i seguenti plot al variare dei seguenti parametri:

Plot	<i>epsilon</i>	<i>epsilon_step_value</i>	<i>max_iter</i>
1	[0.01, 0.02, 0.04, 0.06, 0.08, 0.1]	0.01	10
2	0.1	[0.01, 0.02, 0.03, 0.04, 0.05]	10
3	0.1	0.01	[1, 3, 5, 7, 10]

3.5.1 BIM untargeted

- **Plot 1:** il grafico in Figura 3.3 mostra la trasferibilità dell'attacco BIM untargeted, al variare del parametro ϵ : come atteso, l'accuracy del classificatore NN2 è soggetta a una decrescita minore rispetto a quella del classificatore NN1. Tuttavia, l'attacco riesce comunque a ridurre l'accuratezza fino a circa 0.25 (per $\epsilon=0.1$), dimostrando una certa efficacia anche nei confronti della rete NN2. Dunque, è possibile affermare che l'attacco gode di un'alta trasferibilità.

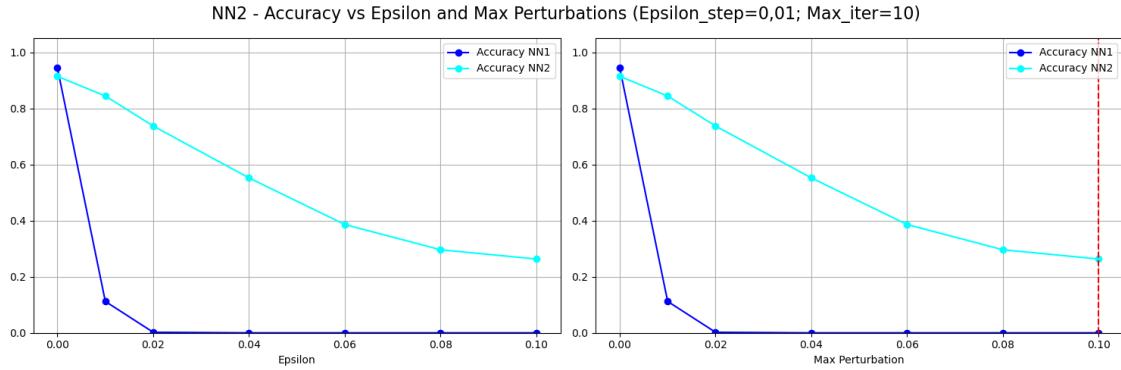


Figure 3.3: NN2 - BIM untargeted - Plot 1

- **Plot 2:** il grafico in Figura 3.4 mostra la trasferibilità dell'attacco BIM untargeted, al variare del parametro ϵ_{step} : come atteso, l'accuracy del classificatore NN2 è soggetta a una decrescita minore rispetto a quella del classificatore NN1. Tuttavia, l'attacco riesce comunque a ridurre l'accuratezza fino a circa 0.2 (per $\epsilon_{step}=0.02$), dimostrando una certa efficacia anche nei confronti della rete NN2. Dunque, è possibile affermare che l'attacco gode di una un'alta trasferibilità.

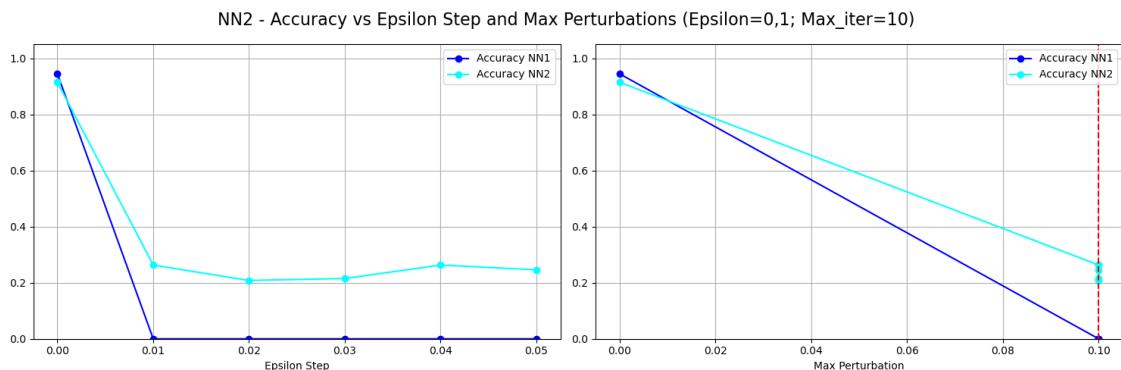


Figure 3.4: NN2 - BIM untargeted - Plot 2

- **Plot 3:** il grafico in Figura 3.5 mostra la trasferibilità dell'attacco BIM untargeted, al variare del parametro max_iter : come atteso, l'accuracy del classificatore NN2 è soggetta a una decrescita minore rispetto a quella del classificatore NN1. Tuttavia, l'attacco riesce comunque a ridurre l'accuratezza fino a circa 0.25 (per $max_iter=10$), dimostrando una certa efficacia anche nei confronti della rete NN2. Dunque, è possibile affermare che l'attacco gode di un'alta trasferibilità.

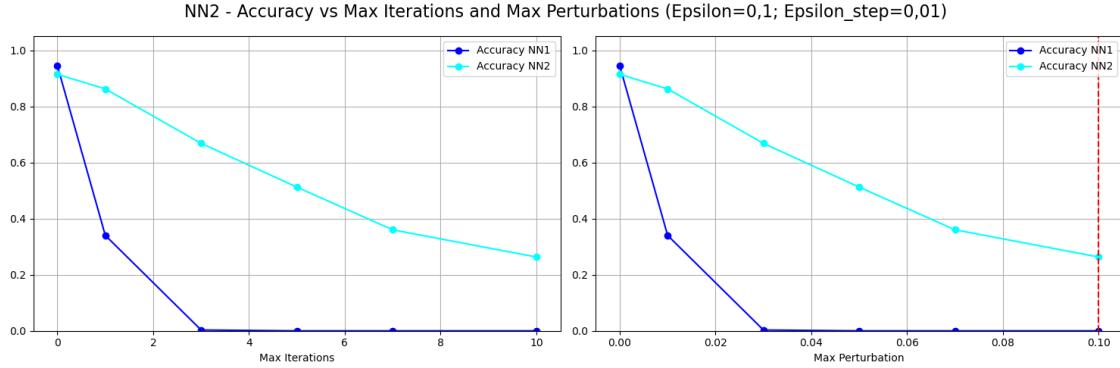


Figure 3.5: NN2 - BIM untargeted - Plot 3

3.5.2 BIM targeted

- **Plot 1:** il grafico in Figura 3.6 mostra la trasferibilità dell’attacco BIM targeted, al variare del parametro ϵ : come atteso, l’accuracy del classificatore NN2 è soggetta a una decrescita minore rispetto a quella del classificatore NN1, ma l’attacco riesce comunque a ridurre l’accuratezza fino a circa 0.7 (per $\epsilon=0.1$). Tuttavia, non è possibile notare un aumento significativo della *targeted_accuracy* della rete NN2, indice dell’incapacità dell’attacco di convertire correttamente le immagini verso la classe target. Dunque, è possibile affermare che l’attacco non è trasferibile.

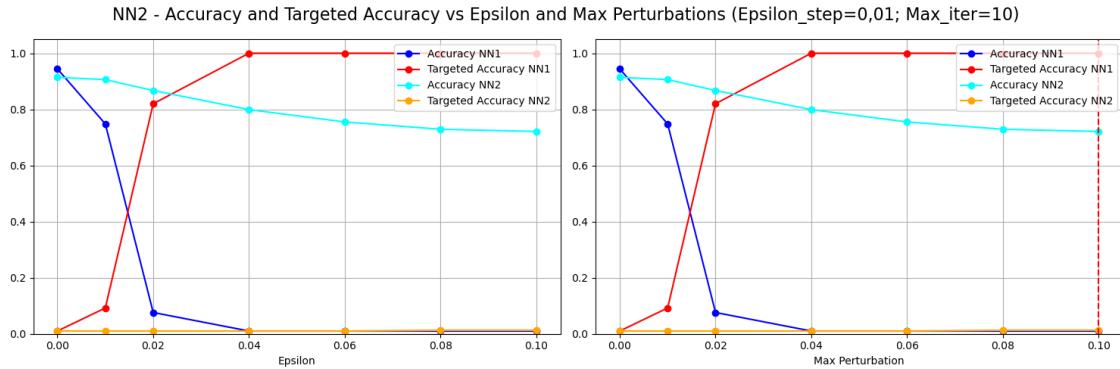


Figure 3.6: NN2 - BIM targeted - Plot 1

- **Plot 2:** il grafico in Figura 3.7 mostra la trasferibilità dell’attacco BIM targeted, al variare del parametro ϵ_{step} : come atteso, l’accuracy del classificatore NN2 è soggetta a una decrescita minore rispetto a quella del classificatore NN1, ma l’attacco riesce comunque a ridurre l’accuratezza fino a circa 0.65 (per $\epsilon_{step}=0.03$). Tuttavia, non è possibile notare un aumento significativo della *targeted_accuracy* della rete NN2, indice dell’incapacità dell’attacco di convertire correttamente le immagini verso la classe target. Dunque, è possibile affermare che l’attacco non è trasferibile.

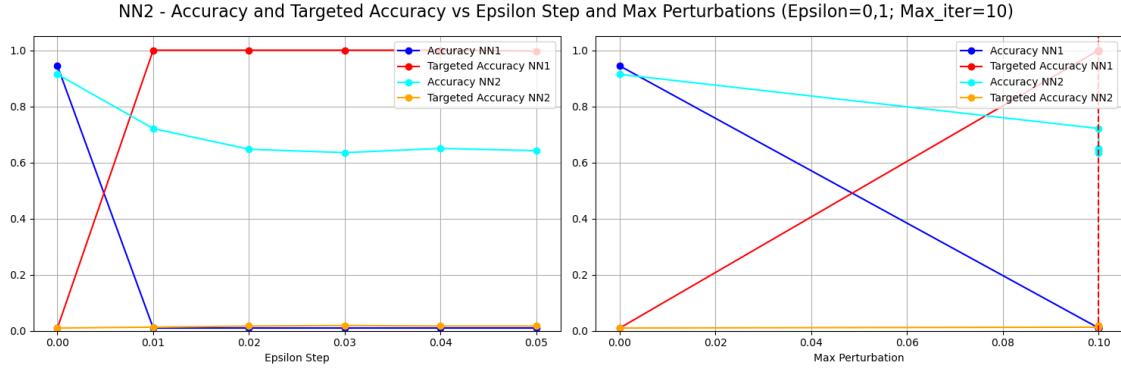


Figure 3.7: NN2 - BIM targeted - Plot 2

- Plot 3:** il grafico in Figura 3.8 mostra la trasferibilità dell'attacco BIM targeted, al variare del parametro *max_iter*: come atteso, l'accuracy del classificatore NN2 è soggetta a una decrescita minore rispetto a quella del classificatore NN1, ma l'attacco riesce comunque a ridurre l'accuratezza fino a circa 0.7 (per *max_iter*= 10). Tuttavia, non è possibile notare un aumento significativo della *targeted_accuracy* della rete NN2, indice dell'incapacità dell'attacco di convertire correttamente le immagini verso la classe target. Dunque, è possibile affermare che l'attacco non è trasferibile.

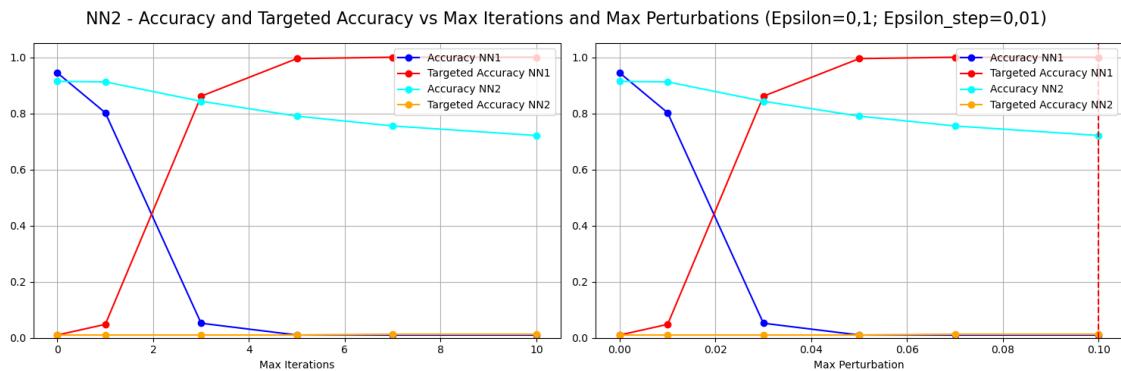


Figure 3.8: NN2 - BIM targeted - Plot 3

3.6 Prestazioni della rete sulle immagini adversarial PGD

Per valutare le prestazioni della rete NN2 sulle immagini adversarial generate dall'attacco PGD, sono state tracciate diverse security evaluation curve. Nello specifico, è stata calcolata una curva per ciascuno dei principali parametri dell'attacco, calcolando sia l'*accuracy* che la *targeted_accuracy* al variare di esso. Questo approccio consente di valutare l'impatto specifico di ciascun parametro dell'attacco sulle prestazioni della rete e di avere un'analisi dettagliata sulla robustezza della rete in funzione della forza dell'attacco. In particolare, verranno tracciati i seguenti plot al variare dei seguenti parametri:

Plot	epsilon	epsilon_step_value	max_iter
1	[0.01, 0.02, 0.04, 0.06, 0.08, 0.1]	0.01	10
2	0.1	[0.01, 0.02, 0.03, 0.04, 0.05]	10
3	0.1	0.01	[1, 3, 5, 7, 10]

3.6.1 PGD untargeted

- **Plot 1:** il grafico in Figura 3.9 mostra la trasferibilità dell'attacco PGD untargeted, al variare del parametro *epsilon*: come atteso, l'accuracy del classificatore NN2 è soggetta a una decrescita minore rispetto a quella del classificatore NN1. Tuttavia, l'attacco riesce comunque a ridurre l'accuratezza fino a circa 0.4 (per *epsilon*= 0.1), dimostrando una certa efficacia anche nei confronti della rete NN2. Dunque, è possibile affermare che, per questa configurazione, l'attacco gode di una buona trasferibilità.

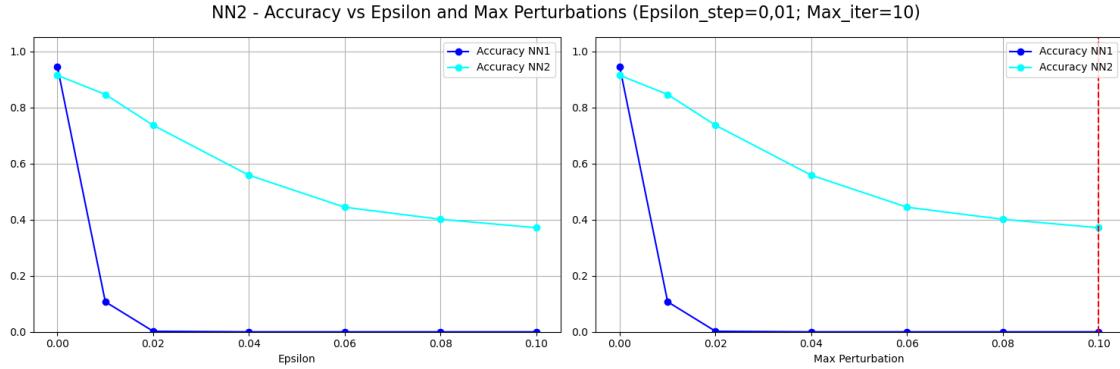


Figure 3.9: NN2 - PGD untargeted - Plot 1

- **Plot 2:** il grafico in Figura 3.10 mostra la trasferibilità dell'attacco PGD untargeted, al variare del parametro *epsilon_step*: come atteso, l'accuracy del classificatore NN2 è soggetta a una decrescita minore rispetto a quella del classificatore NN1. Tuttavia, l'attacco riesce comunque a ridurre l'accuratezza fino a circa 0.25 per diversi valori di *epsilon_step*, dimostrando una certa efficacia anche nei confronti della rete NN2. Dunque, è possibile affermare che, per alcuni valori di *epsilon_step*, l'attacco gode di un'alta trasferibilità.

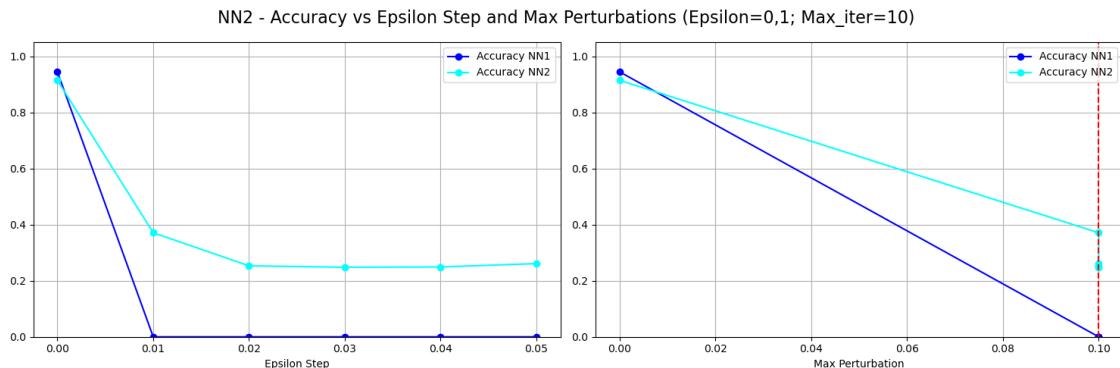


Figure 3.10: NN2 - PGD untargeted - Plot 2

- **Plot 3:** il grafico in Figura 3.11 mostra la trasferibilità dell'attacco PGD untargeted, al variare del parametro *max_iter*: come atteso, l'accuracy del classificatore NN2 è soggetta a una decrescita minore rispetto a quella del classificatore NN1. Tuttavia, l'attacco riesce comunque a ridurre l'accuratezza fino a circa 0.4 (per *max_iter*= 10), dimostrando una certa efficacia anche nei confronti della rete NN2. Dunque, è possibile affermare che, per questa configurazione, l'attacco gode di una buona trasferibilità.

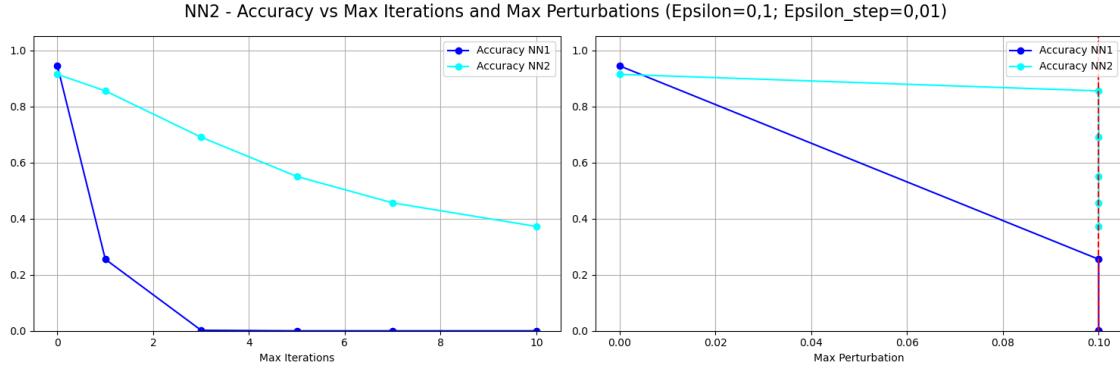


Figure 3.11: NN2 - PGD untargeted - Plot 3

3.6.2 PGD targeted

- Plot 1:** il grafico in Figura 3.12 mostra la trasferibilità dell'attacco PGD targeted, al variare del parametro ϵ : come atteso, l'accuracy del classificatore NN2 è soggetta a una decrescita minore rispetto a quella del classificatore NN1, ma l'attacco riesce comunque a ridurre l'accuratezza fino a circa 0.75 (per $\epsilon = 0.1$). Tuttavia, non è possibile notare un aumento significativo della *targeted_accuracy* della rete NN2, indice dell'incapacità dell'attacco di convertire correttamente le immagini verso la classe target. Dunque, è possibile affermare che l'attacco non è trasferibile.

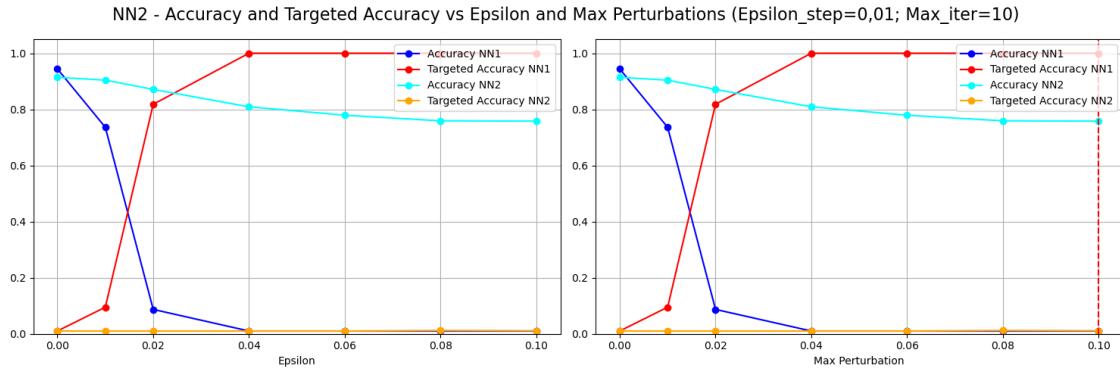


Figure 3.12: NN2 - PGD targeted - Plot 1

- Plot 2:** il grafico in Figura 3.13 mostra la trasferibilità dell'attacco PGD targeted, al variare del parametro ϵ_{step} : come atteso, l'accuracy del classificatore NN2 è soggetta a una decrescita minore rispetto a quella del classificatore NN1, ma l'attacco riesce comunque a ridurre l'accuratezza fino a circa 0.65 (per $\epsilon_{step} = 0.05$). Tuttavia, non è possibile notare un aumento significativo della *targeted_accuracy* della rete NN2, indice dell'incapacità dell'attacco di convertire correttamente le immagini verso la classe target. Dunque, è possibile affermare che l'attacco non è trasferibile.

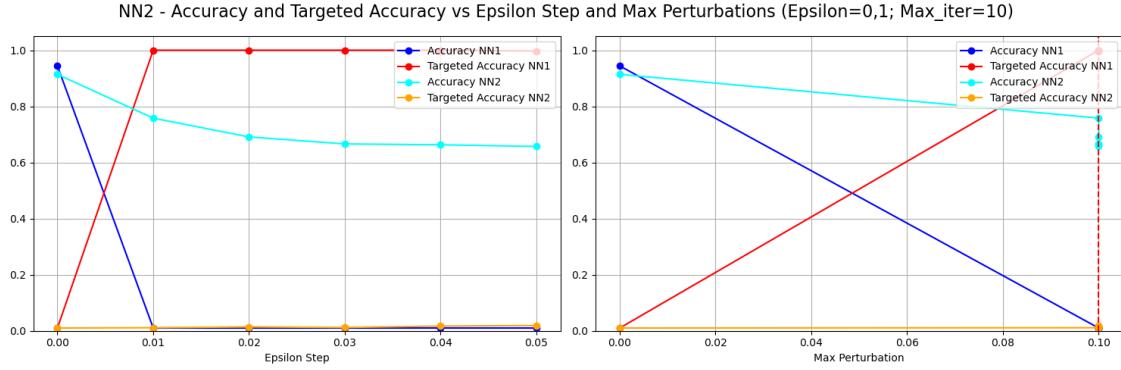


Figure 3.13: NN2 - PGD targeted - Plot 2

- Plot 3:** il grafico in Figura 3.14 mostra la trasferibilità dell'attacco PGD targeted, al variare del parametro *max_iter*: come atteso, l'accuracy del classificatore NN2 è soggetta a una decrescita minore rispetto a quella del classificatore NN1, ma l'attacco riesce comunque a ridurre l'accuratezza fino a circa 0.75 (per *max_iter*= 10). Tuttavia, non è possibile notare un aumento significativo della *targeted_accuracy* della rete NN2, indice dell'incapacità dell'attacco di convertire correttamente le immagini verso la classe target. Dunque, è possibile affermare che l'attacco non è trasferibile.

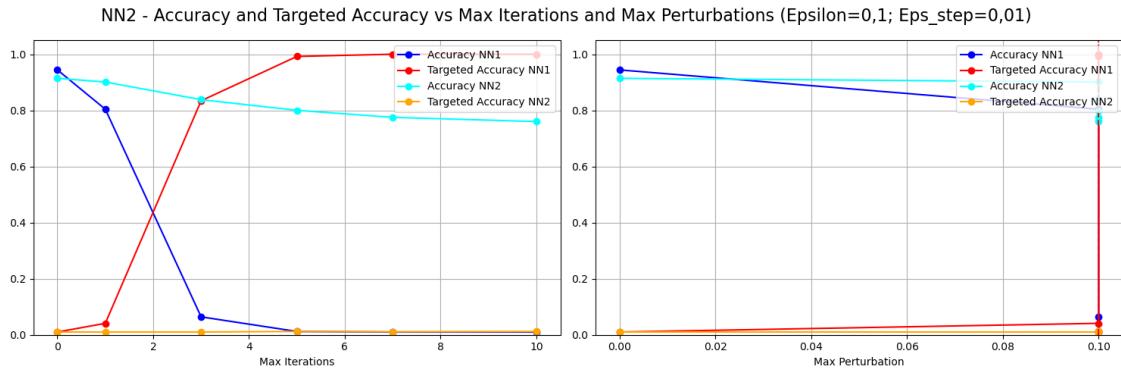


Figure 3.14: NN2 - PGD targeted - Plot 3

3.7 Prestazioni della rete sulle immagini adversarial DF

Per valutare le prestazioni della rete NN2 sulle immagini adversarial generate dall'attacco DF, sono state tracciate diverse security evaluation curve. Nello specifico, è stata calcolata una curva per ciascuno dei principali parametri dell'attacco, calcolando l'*accuracy* al variare di esso. Questo approccio consente di valutare l'impatto specifico di ciascun parametro dell'attacco sulle prestazioni della rete e di avere un'analisi dettagliata sulla robustezza della rete in funzione della forza dell'attacco. In particolare, verranno tracciati i seguenti plot al variare dei seguenti parametri:

Plot	<i>epsilon</i>	<i>nb_grads</i>	<i>max_iter</i>
1	$[1^{-5}, 1^{-4}, 1^{-3}, 1^{-2}, 1^{-1}, 1]$	10	10
2	1^{-2}	$[5, 10, 20, 50]$	10
3	1^{-2}	10	$[1, 3, 5, 7, 10]$

3.7.1 DF untargeted

- **Plot 1:** il grafico in Figura 3.15 mostra la trasferibilità dell'attacco DF untargeted, al variare del parametro *epsilon*: come atteso, l'accuracy del classificatore NN2 è soggetta a una decrescita minore rispetto a quella del classificatore NN1. In particolare, mentre sulla rete NN1 il valore del parametro *epsilon* non fa differenza, sulla rete NN2 l'attacco riesce a ridurre l'accuratezza fino a circa 0.65 solo per *epsilon*= 1.0, dimostrando una discreta efficacia nei confronti della rete NN2. Dunque, è possibile affermare che l'attacco gode di una bassa trasferibilità.

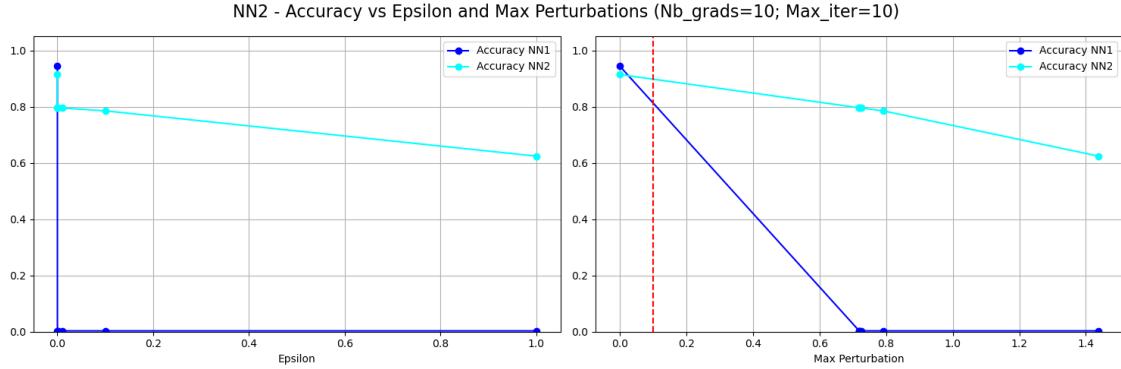


Figure 3.15: NN2 - DF untargeted - Plot 1

- **Plot 2:** il grafico in Figura 3.16 mostra la trasferibilità dell'attacco DF untargeted, al variare del parametro *nd_grads*: come atteso, l'accuracy del classificatore NN2 è soggetta a una decrescita minore rispetto a quella del classificatore NN1. In particolare, l'attacco riesce a ridurre l'accuratezza fino a circa 0.75 (per *nd_grads*= 5), dimostrando poca efficacia nei confronti della rete NN2. Dunque, è possibile affermare che l'attacco gode di una bassa trasferibilità.

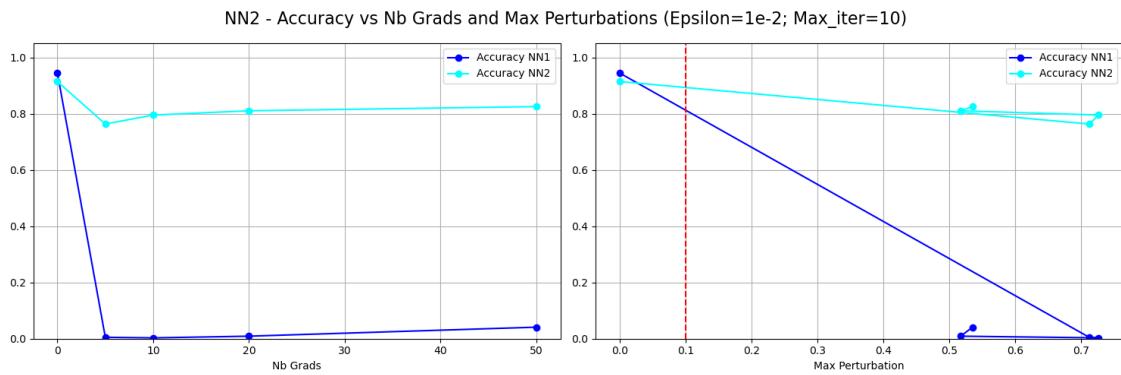


Figure 3.16: NN2 - DF untargeted - Plot 2

- **Plot 3:** il grafico in Figura 3.17 mostra la trasferibilità dell'attacco DF untargeted, al variare del parametro *max_iter*: come atteso, l'accuracy del classificatore NN2 è soggetta a una decrescita minore rispetto a quella del classificatore NN1. In particolare, l'attacco riesce a ridurre l'accuratezza fino a circa 0.8, dimostrando poca efficacia nei confronti della rete NN2. Dunque, è possibile affermare che l'attacco gode di una bassa trasferibilità.

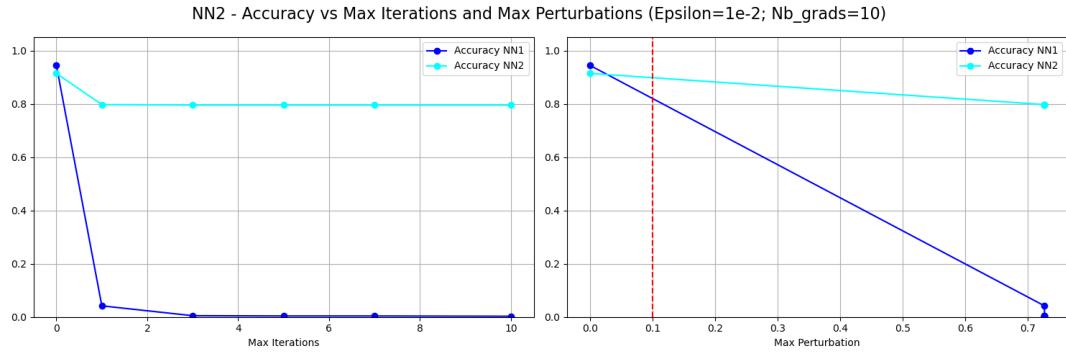


Figure 3.17: NN2 - DF untargeted - Plot 3

3.8 Prestazioni della rete sulle immagini adversarial CW

Per valutare le prestazioni della rete NN2 sulle immagini adversarial generate dall'attacco CW, sono state tracciate diverse security evaluation curve. Nello specifico, è stata calcolata una curva per ciascuno dei principali parametri dell'attacco, calcolando l'*accuracy* al variare di esso (non è stata calcolata anche la *targeted_accuracy* perché l'attacco target non si è rivelato efficace sulla rete NN1). Questo approccio consente di valutare l'impatto specifico di ciascun parametro dell'attacco sulle prestazioni della rete e di avere un'analisi dettagliata sulla robustezza della rete in funzione della forza dell'attacco. In particolare, verranno tracciati i seguenti plot al variare dei seguenti parametri:

Plot	confidence	learning_rate	max_iter
1	[0.01, 0.1, 1]	0.01	3
2	0.1	[0.01, 0.05, 0.1]	3
3	0.1	0.01	[1, 3, 5]

Nota: per l'attacco CW è stato deciso di analizzare esclusivamente la trasferibilità dell'attacco *untargeted*, in quanto dall'analisi effettuata sulla rete NN1 è emerso che la versione *targeted* dell'attacco non è in grado di compromettere efficacemente il modello, per cui si è valutato superfluo valutare la trasferibilità di un attacco inefficace.

3.8.1 CW untargeted

- **Plot 1:** il grafico in Figura 3.18 mostra la trasferibilità dell'attacco CW untargeted, al variare del parametro *confidence*: mentre l'accuracy del classificatore NN1 viene completamente abbattuta per tutti i valori di *confidence* testati, l'accuracy del classificatore NN2 subisce solamente un piccolissimo calo rispetto al caso "clean" (resta all'incirca 0.9). Dunque, è possibile affermare che, per questa configurazione, l'attacco gode di una trasferibilità praticamente nulla.

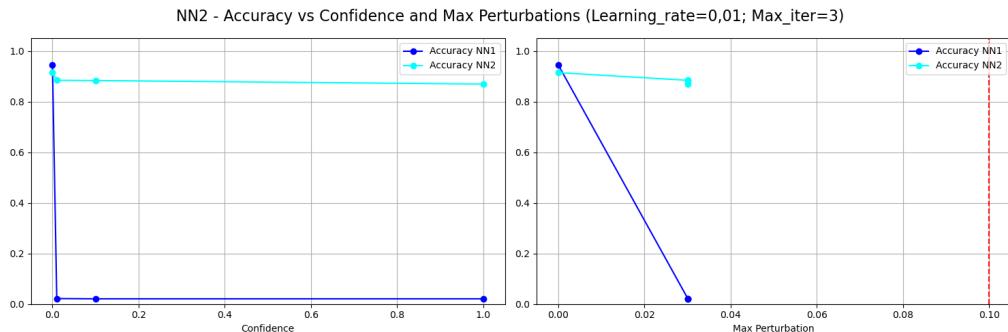


Figure 3.18: NN2 - CW untargeted - Plot 1

- **Plot 2:** il grafico in Figura 3.19 mostra la trasferibilità dell'attacco CW untargeted, al variare del parametro *learning_rate*: come atteso, l'accuracy del classificatore NN2 è soggetta a una decrescita minore rispetto a quella del classificatore NN1. In particolare, mentre sulla rete NN1 il valore del parametro *learning_rate* non fa differenza, sulla rete NN2 l'attacco riesce a ridurre l'accuratezza fino a circa 0.4 per *learning_rate*= 0.1, dimostrando una certa efficacia nei confronti della rete NN2. Dunque, è possibile affermare che, per valori *learning_rate* alti, l'attacco gode di una trasferibilità di medio livello.

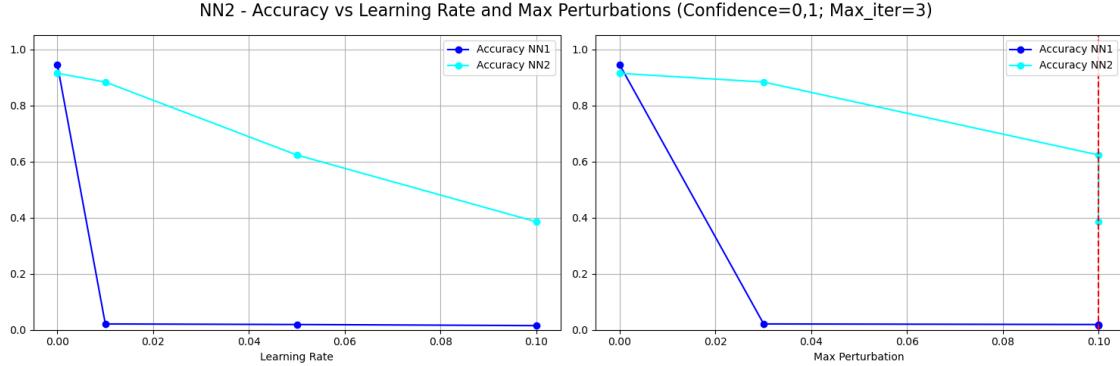


Figure 3.19: NN2 - CW untargeted - Plot 2

- **Plot 3:** il grafico in Figura 3.20 mostra la trasferibilità dell'attacco CW untargeted, al variare del parametro *max_iter*: mentre l'accuracy del classificatore NN1 viene completamente abbattuta per $\text{max_iter} \geq 3$, l'accuracy del classificatore NN2 subisce solamente un piccolissimo calo rispetto al caso "clean" (resta all'incirca 0.9). Dunque, è possibile affermare che, per questa configurazione, l'attacco gode di una trasferibilità praticamente nulla.

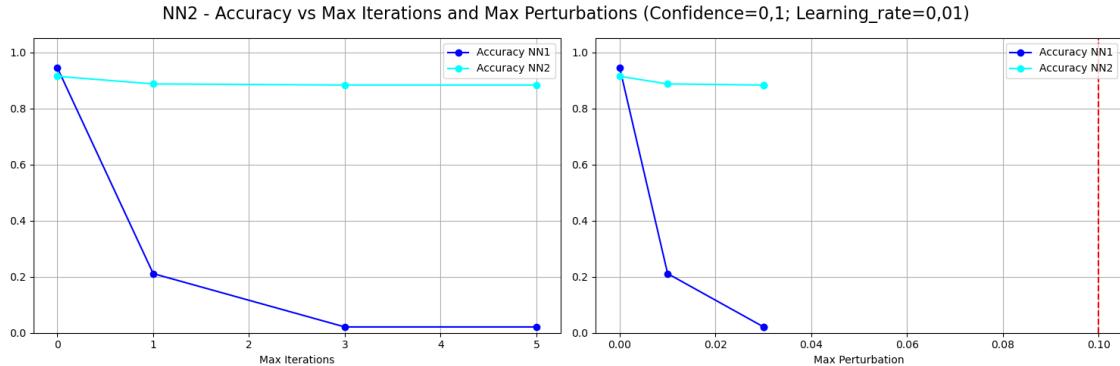


Figure 3.20: NN2 - CW untargeted - Plot 3

3.9 Riepilogo sull’analisi della trasferibilità degli attacchi

In questa sezione viene presentato un breve riepilogo sull’efficacia degli attacchi sulla rete NN2. Per ciascun tipo di attacco viene riportata l’**accuracy minima** (ed eventualmente la **targeted accuracy massima**) ottenuta, insieme alla **relativa perturbazione massima** osservata (nel caso in cui il valore massimo dell’accuracy sia stato ottenuto più volte con perturbazioni diverse, verrà considerata la perturbazione massima più piccola). Inoltre, per ogni attacco, viene riportato il **grado di trasferibilità**, che può essere: nulla(*), bassa(**), media(***), alta(****).

Attacco	Accuracy _{min} (\approx)	Targeted Accuracy _{max} (\approx)	L _{inf} (\approx)	Trasferibilità
FGSM (untargeted)	0.35	/	0.1	***
FGSM (targeted)	0.55	0.03	0.1	*
BIM (untargeted)	0.20	/	0.1	****
BIM (targeted)	0.65	0.01	0.1	*
PGD (untargeted)	0.25	/	0.1	****
PGD (targeted)	0.65	0.01	0.1	*
DF (untargeted)	0.65	/	1.4	**
CW (untargeted)	0.40	/	0.1	***

Dall’analisi dei risultati riportati in tabella, si osserva come gli attacchi *targeted* risultino meno trasferibili rispetto a quelli *untargeted*, in termini di capacità di ridurre l’accuracy del modello. In particolare, la loro efficacia si limita a un calo marginale dell’accuracy complessiva, che rimane significativamente più elevata rispetto a quella degli attacchi *untargeted*. Inoltre, la targeted accuracy resta in linea con quella calcolata sui dati clean per quasi tutti gli attacchi *targeted* (solo l’attacco FGSM riesce ad ottenere un leggerissimo miglioramento), evidenziando l’incapacità di forzare con successo la rete NN2 a classificare i campioni come classe obiettivo.

CHAPTER 4

ANALISI DEL SISTEMA DI DIFESA IMPLEMENTATO

In questo capitolo verrà analizzato il sistema di difesa implementato per la rete NN1, basato sull'utilizzo di detectors in grado di riconoscere campioni adversariali, e verrà effettuata un'analisi delle performance del sistema complessivo NN1+Detectors, in termini di *accuracy* e *targeted accuracy*, sia su dati clean che su dati adversariali. Quest'ultimi sono stati generati con attacchi *Fast Gradient Sign Method (FGSM)*, *Basic Iterative Method (BIM)*, *Projected Gradient Descent (PGD)*, *DeepFool (DF)* e *Carlini-Wagner (CW)*, per ingannare la rete NN1.

Dunque, l'obiettivo di questo capitolo è valutare l'efficacia del sistema di difesa implementato. Per farlo, verranno confrontate le performance del sistema NN1+Detectors con le performance della rete NN1.

4.1 Sistema di difesa implementato

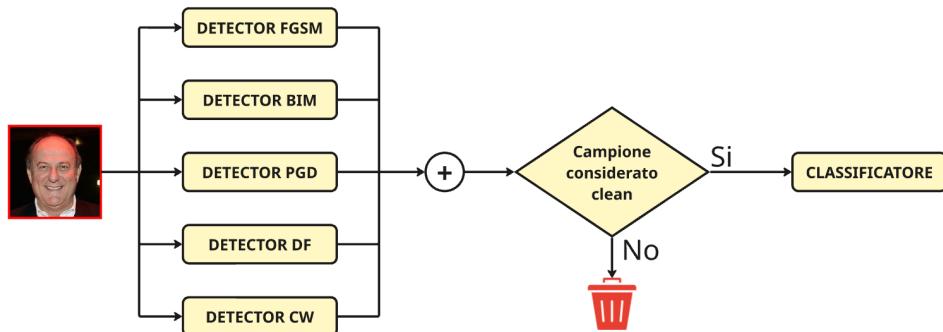


Figure 4.1: Architettura del sistema di difesa implementato.

La strategia di difesa adottata in questo progetto si basa sull'impiego di un sistema di **detectors** di campioni adversariali. In particolare, sono stati implementati cinque detector, ognuno dei quali specificamente addestrato per riconoscere immagini perturbate da una delle cinque tipologie di attacco considerate (FGSM, BIM, PGD, DF, CW). Ogni detector agisce come un classificatore binario, la cui funzione è distinguere tra immagini *clean* e *adversarial*.

Nota: il detector relativo all'attacco DeepFool è stato incluso nel sistema di difesa, nonostante tale attacco non rispetti i vincoli imposti. Questo perché analisi preliminari hanno evidenziato che alcune immagini perturbate tramite DeepFool possono comunque rientrare nei vincoli previsti, risultando potenzialmente efficaci per compromettere il sistema. L'integrazione del relativo detector consente quindi di estendere la capacità di rilevazione anche a tali casi particolari.

E' stato previsto che un'immagine, prima di essere classificata dalla rete principale NN1, debba superare positivamente il controllo di tutti i detectors. Dunque, un'immagine viene considerata "adversarial" e bloccata se **almeno** uno dei detectors la classifica come tale (viene effettuato un OR logico tra gli output dei detectors), per cui solo le immagini che vengono riconosciute come "clean" da tutti i detectors vengono poi elaborate dalla rete di riconoscimento facciale per la classificazione.

Nota: si è preferito addestrare cinque detector specializzati, ciascuno su un singolo tipo di attacco, piuttosto che un unico detector generale addestrato su tutte le tipologie, al fine di ottenere prestazioni migliori in termini di accuratezza nella rilevazione delle perturbazioni. Questa scelta, tuttavia, comporta uno svantaggio in termini di costo computazionale, poiché ogni immagine deve essere elaborata da tutti e cinque i detector.

4.2 Architettura dei singoli detector

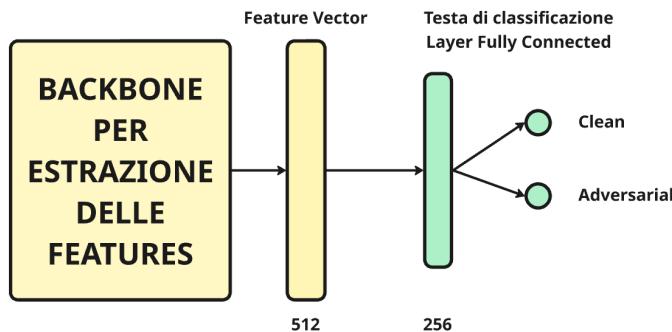


Figure 4.2: Architettura di ogni singolo detector

L'architettura di ogni singolo detector segue un approccio modulare ed è composta dai seguenti elementi:

1. **Backbone:** è costituita dal modello *InceptionResnetV1*, impiegato per estrarre rappresentazioni compatte e significative dell'immagine in input. Questo modulo consente di ottenere un embedding ricco in termini semantici, utile per distinguere efficacemente tra immagini clean e adversarial.
2. **Classificatore:** le feature generate dalla backbone vengono passate a un classificatore costituito da una rete neurale fully connected a due strati. Il primo layer riduce la dimensionalità delle feature da 512 a 256 e applica una funzione di attivazione ReLU, mentre il secondo layer proietta il vettore a 256 dimensioni in uno spazio bidimensionale, corrispondente alle classi *clean* (0) e *adversarial* (1). Dunque, l'output del classificatore è un vettore di due logits, ovvero valori non normalizzati che rappresentano la propensione del modello verso ciascuna classe. L'etichetta finale viene assegnata selezionando il logit col valore maggiore. Questa struttura è stata adottata per garantire la compatibilità con la classe *BinaryInputDetector* della libreria ART (Adversarial Robustness Toolbox), la quale richiede esplicitamente un nodo di output per ciascuna classe.

4.3 Addestramento dei singoli detectors

Al fine di adattare i modelli al dominio specifico e renderli più sensibili alle manipolazioni avversarie, l'addestramento dei detectors ha coinvolto l'intera architettura della rete neurale, inclusa la componente *backbone*. In particolare, l'addestramento di ciascun detector è stato condotto utilizzando i seguenti parametri:

- **loss function:** *CrossEntropyLoss* (necessaria perché la classe *PytorchClassifier*, utilizzata dalla classe *BinaryInputDetector*, richiede un output per ciascuna delle due classi, per cui non è possibile utilizzare la *BinaryCrossEntropyLoss* che richiede un output singolo);
- **ottimizzatore:** Adam;

- **learning rate:** $1e - 4$;
- **numero di epoch:** 30;
- **dimensione del batch:** 16.

4.3.1 Costruzione del train set

Ciascun detector è stato addestrato individualmente utilizzando un **train set bilanciato**, composto da 1000 campioni *clean*, estratti in maniera casuale dal dataset VGGFace2, e da 1000 campioni *adversarial*, generati sulla rete NN1 dallo specifico tipo di attacco che il detector deve rilevare.

- Per quanto riguarda gli attacchi **FGSM**, **BIM** e **PGD**, gli esempi adversarial sono stati bilanciati in modo da includere 500 campioni generati tramite attacchi untargeted e 500 campioni generati tramite attacchi targeted (definendo, per ogni campione, la classe target in maniera casuale), al fine di garantire una maggiore diversità nei dati di addestramento e migliorare la robustezza del detector.
- Tale suddivisione non è stata possibile per l'attacco **DF**, in quanto la libreria ART non supporta la modalità targeted, pertanto i campioni adversarial sono stati generati tutti in modalità untargeted.
- Infine, per l'attacco **CW** è stato deciso di utilizzare esclusivamente esempi untargeted, in quanto dall'analisi effettuata sulla rete NN1 è emerso che la versione targeted dell'attacco non è in grado di compromettere efficacemente il modello, per cui si è valutato superfluo addestrare il detector su un attacco inefficace.

I campioni adversarial utilizzati sono stati generati in modo da ricoprire uniformemente l'intervallo di intensità dell'attacco concesso dai vincoli progettuali. Questo approccio consente di esporre i detectors a una gamma completa di perturbazioni, facilitando l'apprendimento delle diverse manifestazioni dell'attacco. In particolare, la generazione dei campioni per ciascun tipo di attacco è avvenuta utilizzando i seguenti parametri:

- **FGSM:** `eps=random.uniform(0.01, 0.1)`. Per ogni campione viene scelto un valore di *epsilon* in modo casuale da un intervallo uniforme. Questa scelta consente di esporre il detector a tutto lo spettro di valori di *epsilon* previsti dai vincoli progettuali, in modo da addestrarlo a riconoscere perturbazioni di diversa intensità, incluse quelle potenzialmente più subdole.
- **BIM:** `eps=random.uniform(0.01, 0.1), epsilon_step=0.01, max_iter=10`. Per ogni campione viene scelto un valore di *epsilon* in modo casuale da un intervallo uniforme, mentre i parametri *epsilon_step* e *max_iter* sono fissati in maniera tale da conferire all'attacco la massima potenza possibile nel trovare la perturbazione ottimale per l'*epsilon* estratto: infatti, un *epsilon_step* piccolo e un *max_iter* elevato permettono una maggiore esplorazione e precisione nel processo iterativo.
- **PGD:** `eps=random.uniform(0.01, 0.1), epsilon_step=0.01, max_iter=10`. Per ogni campione viene scelto un valore di *epsilon* in modo casuale da un intervallo uniforme, mentre i parametri *epsilon_step* e *max_iter* sono fissati in maniera tale da conferire all'attacco la massima potenza possibile nel trovare la perturbazione ottimale per l'*epsilon* estratto: infatti, un *epsilon_step* piccolo e un *max_iter* elevato permettono una maggiore esplorazione e precisione nel processo iterativo.
- **DeepFool (DF):** `eps=random.uniform(1e-5, 1.0), nb_grads=10, max_iter=10`. I valori sono stati scelti per assicurare un buon compromesso tra varietà dei campioni ed efficienza dell'attacco. Infatti, l'intervallo di *epsilon* scelto è molto ampio e consente di generare attacchi da quasi impercettibili a molto evidenti, mentre i valori scelti di *nb_grads* e *max_iter* rappresentano un buon compresso tra prestazioni ed efficienza.
- **Carlini-Wagner (CW):** `confidence=random.uniform(0.01, 1.0), learning_rate=0.01, max_iter=3`. I valori sono stati scelti per assicurare un buon compromesso tra varietà dei campioni ed efficienza dell'attacco. Infatti, l'intervallo di *confidence* scelto è molto ampio e consente di generare attacchi da quasi impercettibili a molto evidenti, mentre i valori scelti di *nb_grads* e *max_iter* rappresentano un buon compresso tra prestazioni ed efficienza.

4.4 Testing dei singoli detectors

Ciascun detector è stato testato individualmente utilizzando un **test set bilanciato**, composto da 1000 campioni *clean*, appartenenti al test set di partenza, e da 1000 campioni *adversarial*, generati sulla rete NN1 dallo specifico tipo di attacco che il detector deve rilevare.

- Per quanto riguarda gli attacchi **FGSM**, **BIM** e **PGD**, gli esempi adversarial sono stati bilanciati in modo da includere 500 campioni generati tramite attacchi untarged e 500 tramite attacchi targeted, al fine di garantire una maggiore diversità nei dati di test.
- Tale suddivisione non è stata possibile per l'attacco **DF**, in quanto la libreria ART non supporta la modalità targeted, pertanto i campioni adversarial sono stati generati tutti in modalità untarged.
- Infine, per l'attacco **CW** è stato deciso di utilizzare esclusivamente esempi untarged, in quanto dall'analisi effettuata sulla rete NN1 è emerso che la versione targeted dell'attacco non è in grado di compromettere efficacemente il modello, per cui si è valutato superfluo testare il detector su un attacco inefficace.

I campioni adversarial utilizzati sono stati generati in modo da ricoprire l'intero l'intervallo di intensità dell'attacco concesso dai vincoli progettuali, in maniera tale da testare i detectors su una gamma completa di perturbazioni. In particolare, per ogni tipo di attacco, sono stati selezionati in maniera casuale 1000 campioni generati con i parametri corrispondenti alle configurazioni del **plot1**, descritte nei capitoli precedenti.

Nel corso degli esperimenti, come valore di soglia di decisione (**threshold**) è stato utilizzato il valore 0,5, ovvero un campione viene considerato avversario se la probabilità calcolata dal detector supera questo valore. Tuttavia, la soglia può essere modificata in base all'applicazione specifica del sistema: è consigliato aumentarla in scenari in cui è prioritario ridurre i falsi positivi (sistemi sottoposti a un numero limitato di attacchi), mentre è consigliato ridurla in scenari in cui è prioritario ridurre i falsi negativi (sistemi sottoposti a un numero elevato di attacchi).

Per ciascun detector (classificatore binario), le metriche di valutazione utilizzate sono state:

- **True Negative (TN)**: numero di campioni *clean* correttamente classificati come tali.
- **False Positive (FP)**: numero di campioni *clean* erroneamente classificati come *adversarial*.
- **False Negative (FN)**: numero di campioni *adversarial* erroneamente classificati come *clean*.
- **True Positive (TP)**: numero di campioni *adversarial* correttamente classificati come tali.
- **Accuracy**:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- **Precision**:

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall**:

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1-score**:

$$F1 - score = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Curva ROC**: rappresenta la relazione tra il *True Positive Rate (TPR)* (corrisponde alla Recall) e il *False Positive Rate (FPR)* (definito come $FPR = \frac{FP}{TP+FN}$), al variare della soglia di decisione (threshold). Un classificatore efficace si caratterizza per una curva che si avvicina all'angolo superiore sinistro del grafico, corrispondente ad un elevato TPR e a un basso FPR. Al contrario, una curva prossima alla diagonale indica una classificazione casuale. L'area sottesa dalla curva (AUC, Area Under the Curve) viene comunemente impiegata come misura riassuntiva delle prestazioni del modello.

Nelle sezioni seguenti sono riportati i risultati dei test ottenuti per ciascun detector.

4.4.1 FGSM Detector

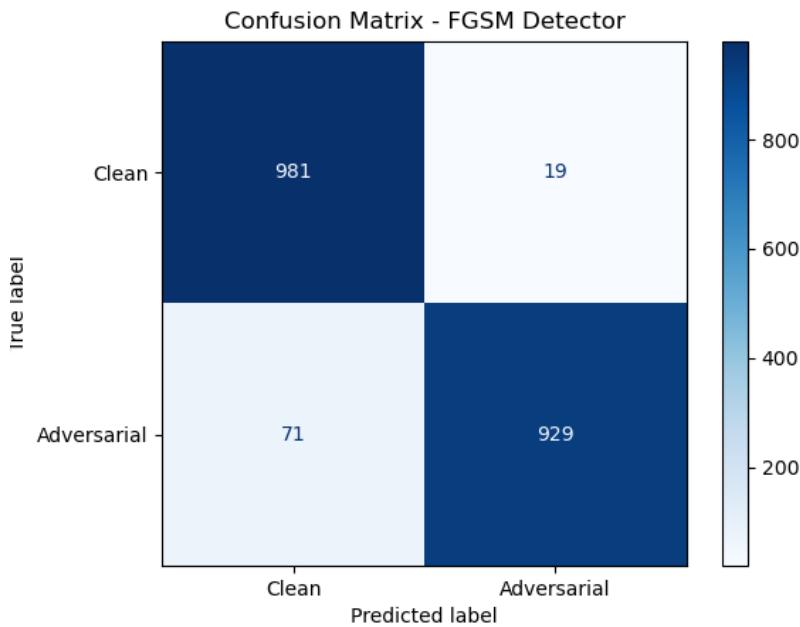


Figure 4.3: Matrice di confusione FGSM Detector

Metrica	Valore
Accuracy	0.9550
Precision	0.9800
Recall	0.9290
F1-score	0.9538

Table 4.1: Metriche di valutazione FGSM Detector

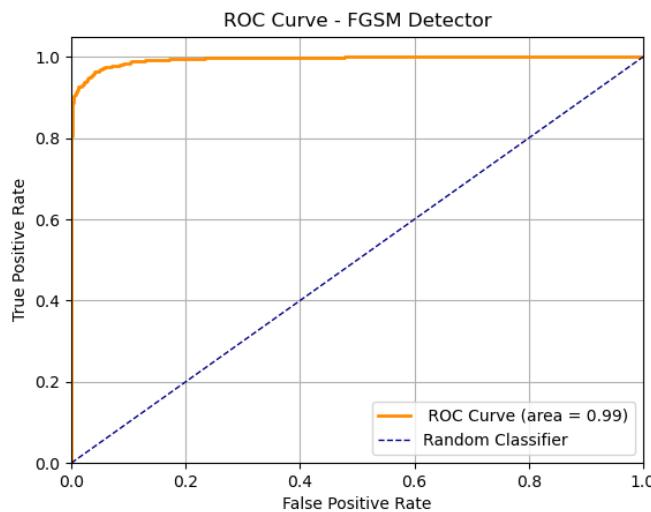


Figure 4.4: Curva ROC FGSM Detector

4.4.2 BIM Detector

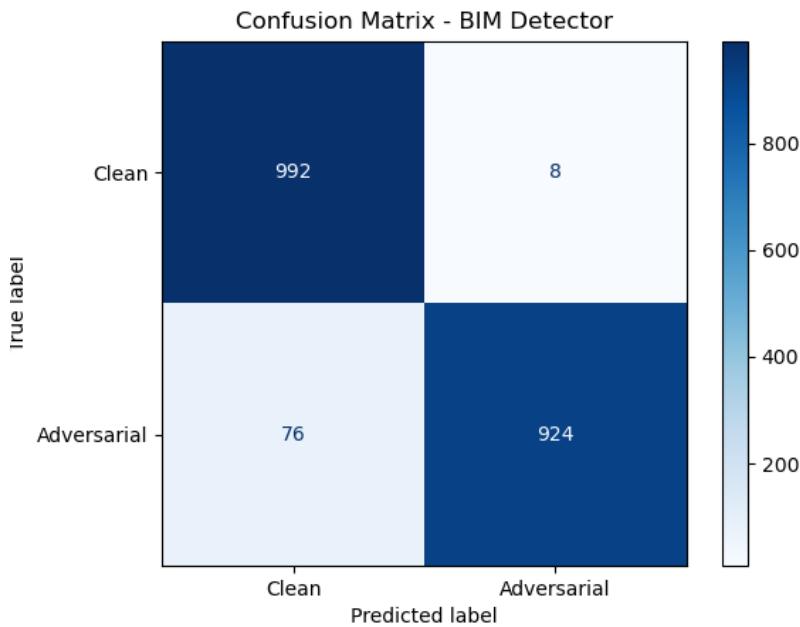


Figure 4.5: Matrice di confusione BIM Detector

Metrica	Valore
Accuracy	0.9580
Precision	0.9914
Recall	0.9240
F1-score	0.9565

Table 4.2: Metriche di valutazione BIM Detector

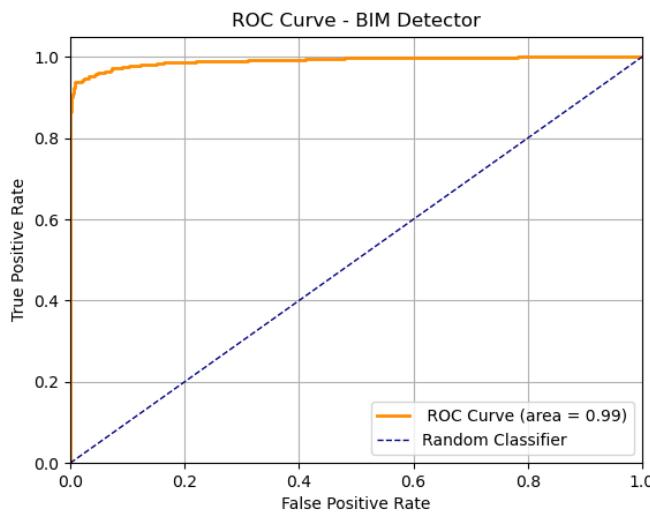


Figure 4.6: Curva ROC BIM Detector

4.4.3 PGD Detector

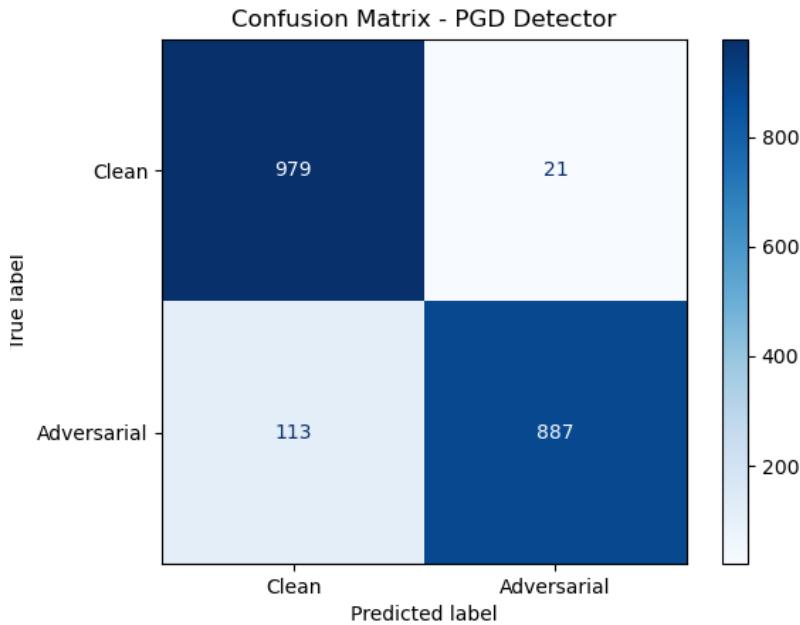


Figure 4.7: Matrice di confusione PGD Detector

Metrica	Valore
Accuracy	0.9330
Precision	0.9769
Recall	0.8870
F1-score	0.9303

Table 4.3: Metriche di valutazione PGD Detector

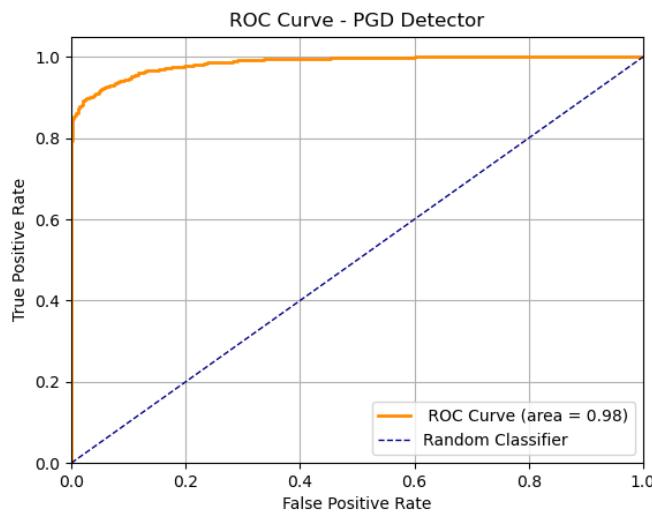


Figure 4.8: Curva ROC PGD Detector

4.4.4 DF Detector

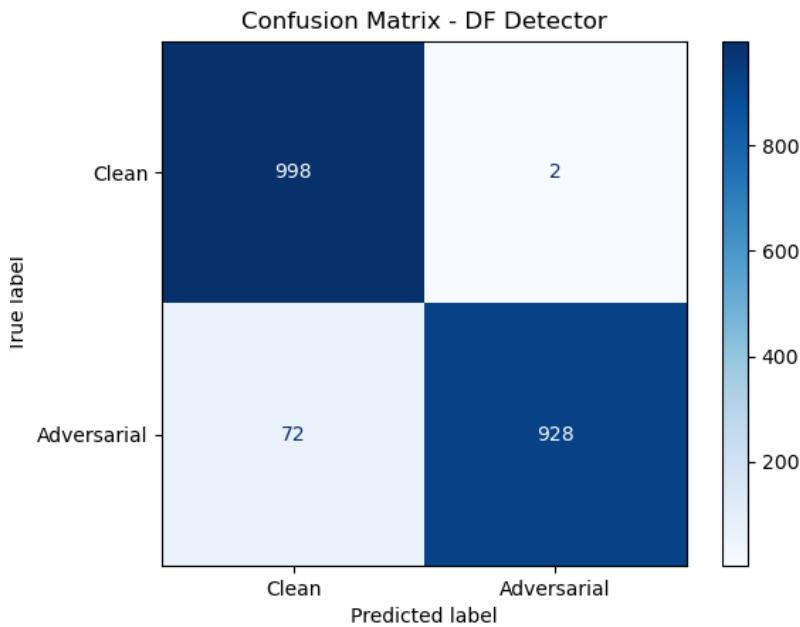


Figure 4.9: Matrice di confusione DF Detector

Metrica	Valore
Accuracy	0.9630
Precision	0.9978
Recall	0.9280
F1-score	0.9617

Table 4.4: Metriche di valutazione DF Detector

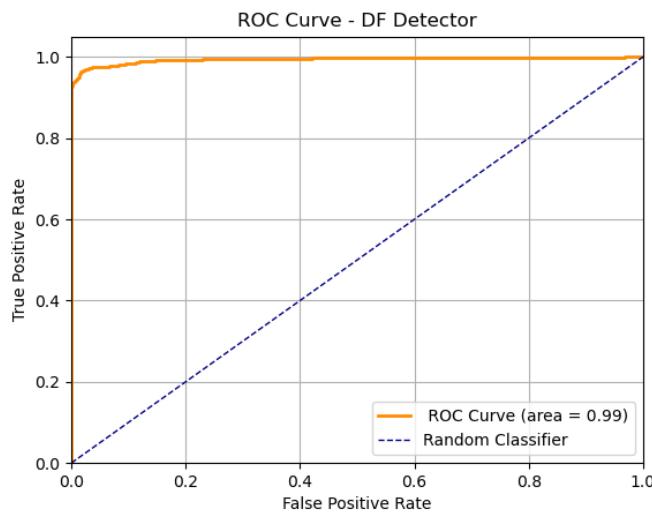


Figure 4.10: Curva ROC DF Detector

4.4.5 CW Detector

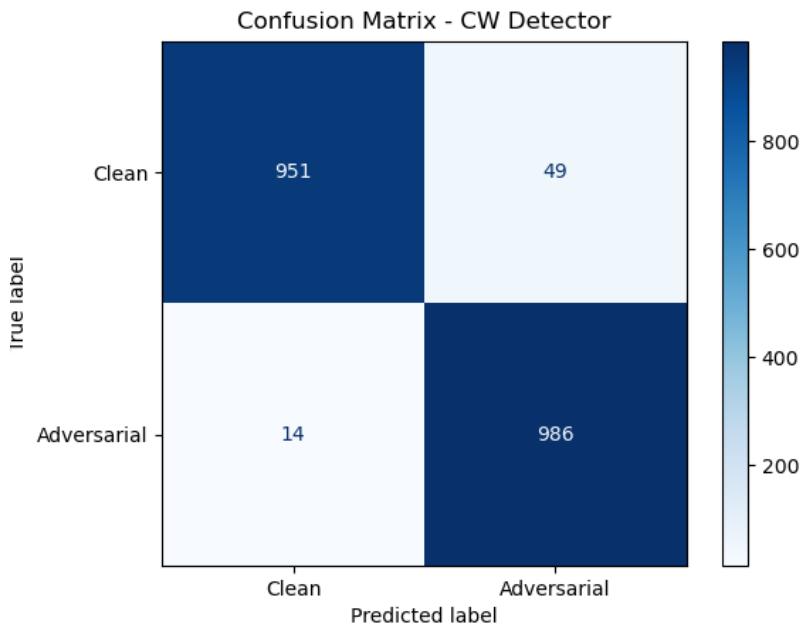


Figure 4.11: Matrice di confusione CW Detector

Metrica	Valore
Accuracy	0.9685
Precision	0.9527
Recall	0.9860
F1-score	0.9690

Table 4.5: Metriche di valutazione CW Detector

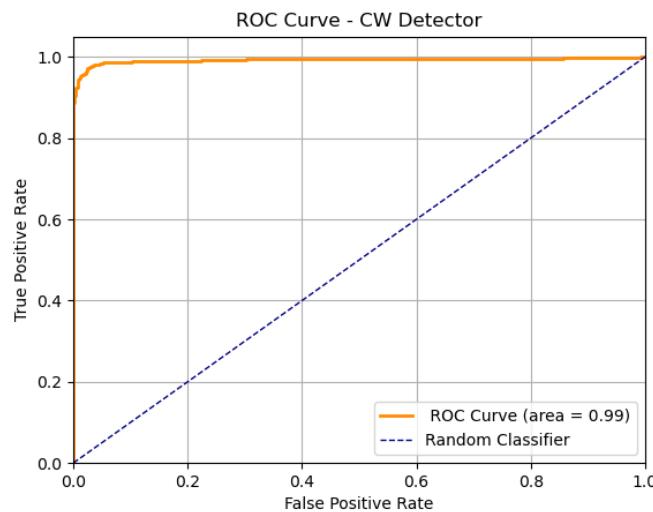


Figure 4.12: Curva ROC CW Detector

4.5 Criteri di valutazione del classificatore NN1+Detectors

L'**accuracy** del sistema complessivo, composto dai detectors e dal classificatore NN1, è stata definita come:

$$\text{Accuracy} = \frac{N_{\text{campioni_correttamente_scartati}} + N_{\text{campioni_correttamente_classificati}}}{N_{\text{totale_campioni}}}$$

- Gli elementi che contribuiscono positivamente a questa metrica sono:
 - i campioni adversarial che vengono correttamente scartati dai detector;
 - i campioni che superano i controlli dei detectors e vengono classificati correttamente.
- Gli elementi che contribuiscono negativamente a questa metrica sono:
 - i campioni clean che vengono erroneamente scartati dai detector;
 - i campioni che superano i controlli dei detectors e vengono classificati erroneamente.

La **targeted accuracy** del sistema complessivo, composto dai detectors e dal classificatore NN1, è stata definita come:

$$\text{Targeted_accuracy} = \frac{N_{\text{campioni_classificati_come_classe_target}}}{N_{\text{totale_campioni}}}$$

- Gli elementi che contribuiscono positivamente a questa metrica sono:
 - i campioni che superano i controlli dei detectors e vengono classificati come classe target.
- Gli elementi che contribuiscono negativamente a questa metrica sono:
 - i campioni (sia clean che adversarial) che vengono scartati dai detectors;
 - i campioni che superano i controlli dei detectors e non vengono classificati come classe target.

4.6 Prestazioni della rete sulle immagini clean

Sul test set costituito da immagini clean, il classificatore NN1+Detectors ha ottenuto un'**accuracy** pari al **87.3%**, classificando correttamente 873 immagini su un totale di 1000.

La **targeted accuracy** sullo stesso test set è risultata pari allo **0.008%**, con 8 immagini su 1000 classificate come appartenenti alla classe target utilizzata (*Cristiano Ronaldo*). Tali immagini corrispondono a 8 delle 10 istanze reali associate a quella classe presenti nel test set.

Si osserva che sia l'accuracy che la targeted accuracy risultano inferiori rispetto a quelle ottenuta dalla rete NN1 in assenza del sistema di difesa (rispettivamente, 94.4% e 0.01%). Questo comportamento è atteso, in quanto l'inserimento dei detectors introduce la possibilità di rilevare dei **falsi positivi**. Dunque, con l'inserimento del sistema di difesa si accetta una riduzione dell'accuratezza sui dati clean, al fine di migliorare la robustezza complessiva del sistema in presenza di attacchi avversari.

Di seguito è riportato il numero di campioni clean scartati da ciascun detector (il totale non corrisponde necessariamente alla somma dei valori singoli perché alcuni campioni possono essere stati scartati da più detectors contemporaneamente):

Detector	Campioni clean scartati
Detector FGSM	19
Detector BIM	8
Detector PGD	21
Detector DeepFool	2
Detector Carlini	49
Totale	80

4.7 Prestazioni della rete sulle immagini adversarial FGSM

Per valutare le prestazioni della rete NN1+detectors sulle immagini adversarial generate dall'attacco FGSM, sono state tracciate diverse security evaluation curve. Nello specifico, è stata calcolata una curva per ciascuno dei principali parametri dell'attacco, calcolando sia l'*accuracy* che la *targeted_accuracy* al variare di esso. Questo approccio consente di valutare l'impatto specifico di ciascun parametro dell'attacco sulle prestazioni della rete e di avere un'analisi dettagliata sulla robustezza della rete in funzione della forza dell'attacco. In particolare, verranno tracciati i seguenti plot al variare dei seguenti parametri:

Plot	<i>epsilon</i>
1	[0.01, 0.02, 0.04, 0.06, 0.08, 0.1]

4.7.1 FGSM untargeted

- **Plot 1:** il grafico in Figura 4.13 mostra l'efficacia del sistema di difesa rispetto all'attacco FGSM untargeted, al variare del parametro *epsilon*: come atteso, l'accuracy del classificatore NN1+Detectors è soggetta ad un netto miglioramento rispetto a quella del classificatore NN1, raggiungendo il valore di circa 1.0 (per tutti i valori di *epsilon*), dovuto al fatto che i detectors scartano quasi tutti i campioni adversarial. Dunque, è possibile affermare che, per questo tipo di attacco, il sistema complessivo NN1+Detectors gode di ottime prestazioni.

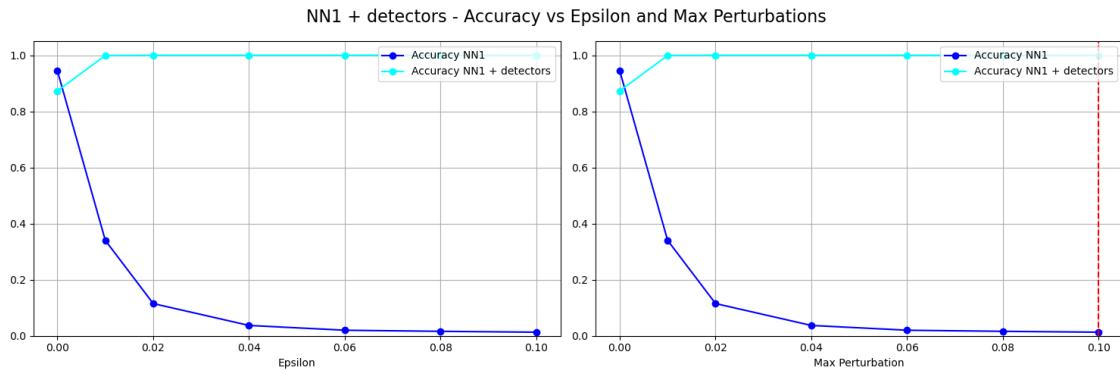


Figure 4.13: NN1+Detectors - FGSM untargeted - Plot 1

4.7.2 FGSM targeted

- **Plot 1:** il grafico in Figura 4.14 mostra l'efficacia del sistema di difesa rispetto all'attacco FGSM targeted, al variare del parametro *epsilon*: come atteso, l'accuracy del classificatore NN1+Detectors è soggetta ad un netto miglioramento rispetto a quella del classificatore NN1, raggiungendo il valore di circa 1.0 (per tutti i valori di *epsilon*), dovuto al fatto che i detectors scartano quasi tutti i campioni adversarial. Inoltre, è possibile notare che la targeted accuracy del classificatore NN1+Detectors raggiunge il valore di circa 0.0 (per tutti i valori di *epsilon*), sottolineando l'inefficacia degli attacchi. Dunque, è possibile affermare che, per questo tipo di attacco, il sistema complessivo NN1+Detectors gode di ottime prestazioni.

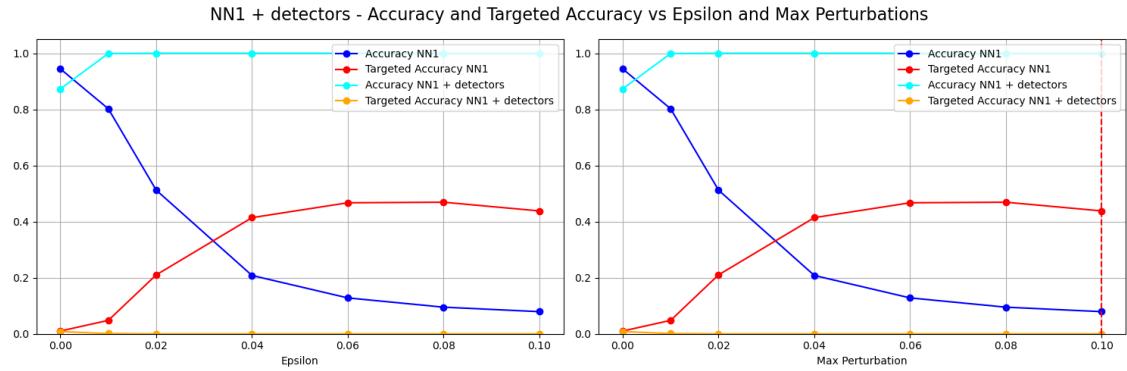


Figure 4.14: NN1+Detectors - FGSM targeted - Plot 1

4.8 Prestazioni della rete sulle immagini adversarial BIM

Per valutare le prestazioni della rete NN1+detectors sulle immagini adversarial generate dall'attacco BIM, sono state tracciate diverse security evaluation curve. Nello specifico, è stata calcolata una curva per ciascuno dei principali parametri dell'attacco, calcolando sia l'*accuracy* che la *targeted_accuracy* al variare di esso. Questo approccio consente di valutare l'impatto specifico di ciascun parametro dell'attacco sulle prestazioni della rete e di avere un'analisi dettagliata sulla robustezza della rete in funzione della forza dell'attacco. In particolare, verranno tracciati i seguenti plot al variare dei seguenti parametri:

Plot	<i>epsilon</i>	<i>epsilon_step_value</i>	<i>max_iter</i>
1	[0.01, 0.02, 0.04, 0.06, 0.08, 0.1]	0.01	10
2	0.1	[0.01, 0.02, 0.03, 0.04, 0.05]	10
3	0.1	0.01	[1, 3, 5, 7, 10]

4.8.1 BIM untargeted

- **Plot 1:** il grafico in Figura 4.15 mostra l'efficacia del sistema di difesa rispetto all'attacco BIM untargeted, al variare del parametro *epsilon*: come atteso, l'accuracy del classificatore NN1+Detectors è soggetta ad un netto miglioramento rispetto a quella del classificatore NN1, raggiungendo il valore di circa 1.0 (per tutti i valori di *epsilon*), dovuto al fatto che i detectors scartano quasi tutti i campioni adversarial. Dunque, è possibile affermare che, per questo tipo di attacco, il sistema complessivo NN1+Detectors gode di ottime prestazioni.

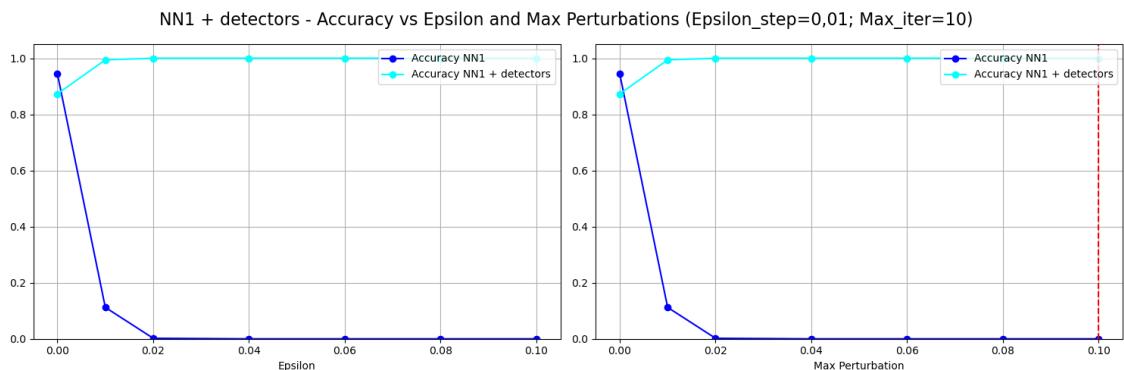


Figure 4.15: NN1+Detector - BIM untargeted - Plot 1

- **Plot 2:** il grafico in Figura 4.16 mostra l'efficacia del sistema di difesa rispetto all'attacco BIM untargeted, al variare del parametro epsilon_step : come atteso, l'accuracy del classificatore NN1+Detectors è soggetta ad un netto miglioramento rispetto a quella del classificatore NN1, raggiungendo il valore di circa 1.0 (per tutti i valori di epsilon_step), dovuto al fatto che i detectors scartano quasi tutti i campioni adversarial. Dunque, è possibile affermare che, per questo tipo di attacco, il sistema complessivo NN1+Detectors gode di ottime prestazioni.

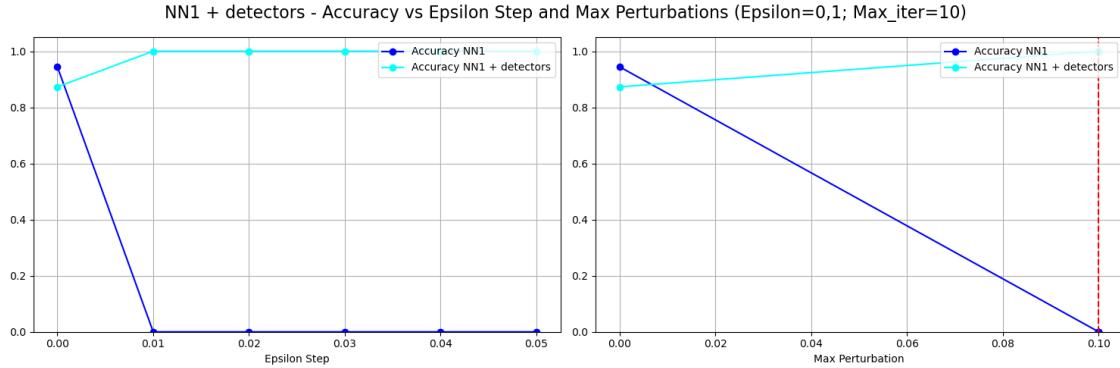


Figure 4.16: NN1+Detectors - BIM untargeted - Plot 2

- **Plot 3:** il grafico in Figura 4.17 mostra l'efficacia del sistema di difesa rispetto all'attacco BIM untargeted, al variare del parametro max_iter : come atteso, l'accuracy del classificatore NN1+Detectors è soggetta ad un netto miglioramento rispetto a quella del classificatore NN1, raggiungendo il valore di circa 1.0 (per tutti i valori di max_iter), dovuto al fatto che i detectors scartano quasi tutti i campioni adversarial. Dunque, è possibile affermare che, per questo tipo di attacco, il sistema complessivo NN1+Detectors gode di ottime prestazioni.

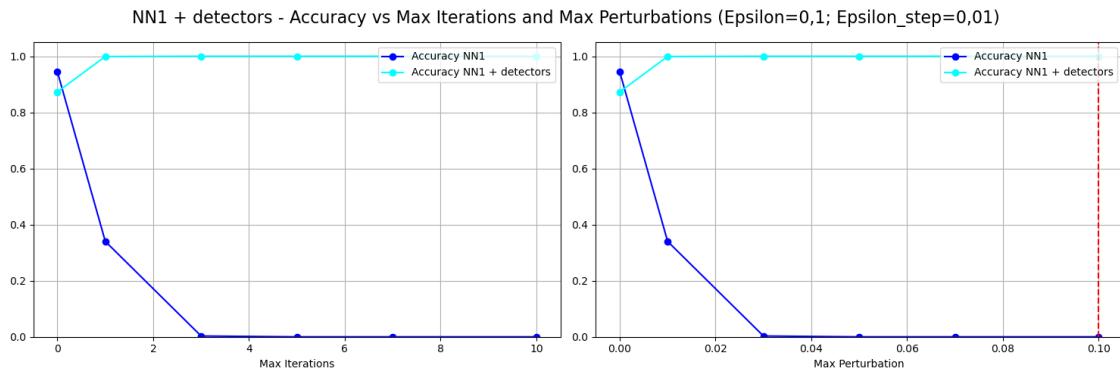


Figure 4.17: NN1+Detectors - BIM untargeted - Plot 3

4.8.2 BIM targeted

- **Plot 1:** il grafico in Figura 4.18 mostra l'efficacia del sistema di difesa rispetto all'attacco BIM targeted, al variare del parametro epsilon : come atteso, l'accuracy del classificatore NN1+Detectors è soggetta ad un netto miglioramento rispetto a quella del classificatore NN1, raggiungendo il valore di circa 1.0 (per tutti i valori di epsilon), dovuto al fatto che i detectors scartano quasi tutti i campioni adversarial. Inoltre, è possibile notare che la targeted accuracy del classificatore NN1+Detectors raggiunge il valore di circa 0.0 (per tutti i valori di epsilon), sottolineando l'inefficacia degli attacchi. Dunque, è possibile affermare che, per questo tipo di attacco, il sistema complessivo NN1+Detectors gode di ottime prestazioni.

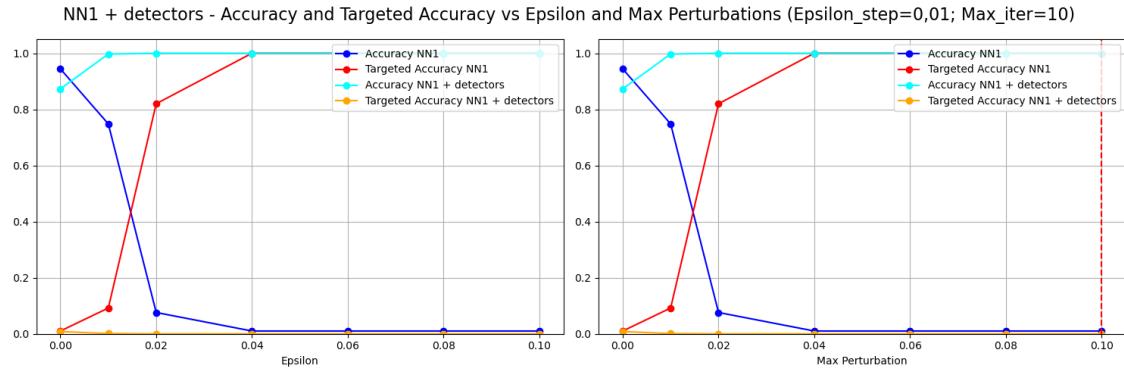


Figure 4.18: NN1+Detectors - BIM targeted - Plot 1

- Plot 2:** il grafico in Figura 4.19 mostra l'efficacia del sistema di difesa rispetto all'attacco BIM targeted, al variare del parametro *epsilon_step*: come atteso, l'accuracy del classificatore NN1+Detectors è soggetta ad un netto miglioramento rispetto a quella del classificatore NN1, raggiungendo il valore di circa 1.0 (per tutti i valori di *epsilon_step*), dovuto al fatto che i detectors scartano quasi tutti i campioni adversariali. Inoltre, è possibile notare che la targeted accuracy del classificatore NN1+Detectors raggiunge il valore di circa 0.0 (per tutti i valori di *epsilon_step*), sottolineando l'inefficacia degli attacchi. Dunque, è possibile affermare che, per questo tipo di attacco, il sistema complessivo NN1+Detectors gode di ottime prestazioni.

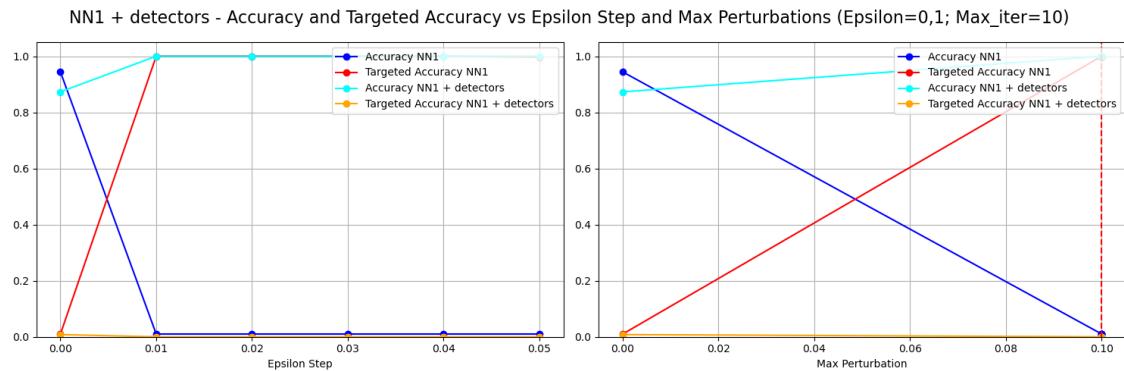


Figure 4.19: NN1+Detectors - BIM targeted - Plot 2

- Plot 3:** il grafico in Figura 4.20 mostra l'efficacia del sistema di difesa rispetto all'attacco BIM targeted, al variare del parametro *max_iter*: come atteso, l'accuracy del classificatore NN1+Detectors è soggetta ad un netto miglioramento rispetto a quella del classificatore NN1, raggiungendo il valore di circa 1.0 (per tutti i valori di *max_iter*), dovuto al fatto che i detectors scartano quasi tutti i campioni adversariali. Inoltre, è possibile notare che la targeted accuracy del classificatore NN1+Detectors raggiunge il valore di circa 0.0 (per tutti i valori di *max_iter*), sottolineando l'inefficacia degli attacchi. Dunque, è possibile affermare che, per questo tipo di attacco, il sistema complessivo NN1+Detectors gode di ottime prestazioni.

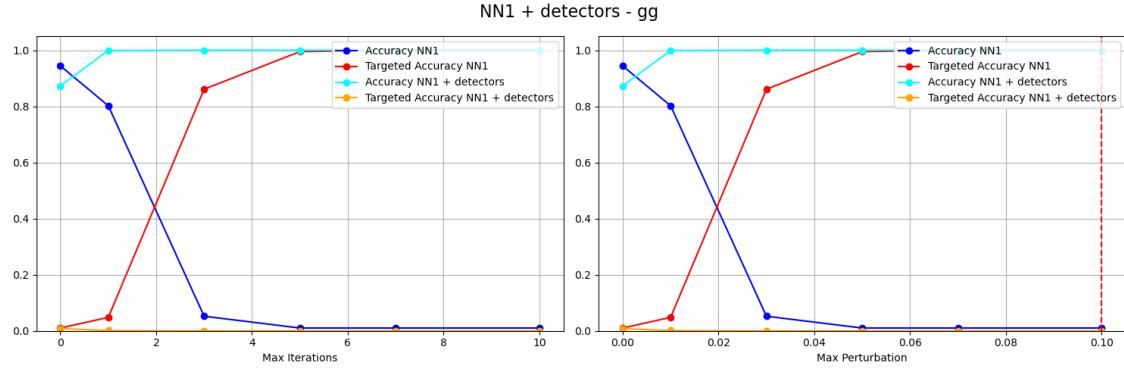


Figure 4.20: NN1+Detectors - BIM targeted - Plot 3

4.9 Prestazioni della rete sulle immagini adversarial PGD

Per valutare le prestazioni della rete NN1+detectors sulle immagini adversarial generate dall'attacco PGD, sono state tracciate diverse security evaluation curve. Nello specifico, è stata calcolata una curva per ciascuno dei principali parametri dell'attacco, calcolando sia l'*accuracy* che la *targeted_accuracy* al variare di esso. Questo approccio consente di valutare l'impatto specifico di ciascun parametro dell'attacco sulle prestazioni della rete e di avere un'analisi dettagliata sulla robustezza della rete in funzione della forza dell'attacco. In particolare, verranno tracciati i seguenti plot al variare dei seguenti parametri:

Plot	<i>epsilon</i>	<i>epsilon_step_value</i>	<i>max_iter</i>
1	[0.01, 0.02, 0.04, 0.06, 0.08, 0.1]	0.01	10
2	0.1	[0.01, 0.02, 0.03, 0.04, 0.05]	10
3	0.1	0.01	[1, 3, 5, 7, 10]

4.9.1 PGD untargeted

- **Plot 1:** il grafico in Figura 4.21 mostra l'efficacia del sistema di difesa rispetto all'attacco PGD untargeted, al variare del parametro *epsilon*: come atteso, l'accuracy del classificatore NN1+Detectors è soggetta ad un netto miglioramento rispetto a quella del classificatore NN1, raggiungendo il valore di circa 1.0 (per tutti i valori di *epsilon*), dovuto al fatto che i detectors scartano quasi tutti i campioni adversarial. Dunque, è possibile affermare che, per questo tipo di attacco, il sistema complessivo NN1+Detectors gode di ottime prestazioni.

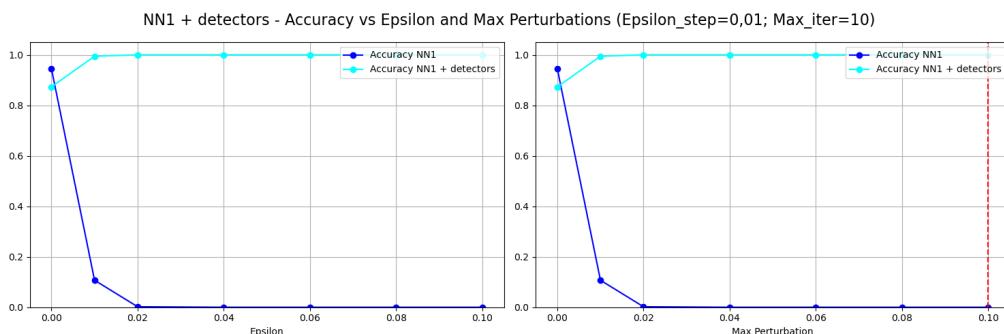


Figure 4.21: NN1+Detectors - PGD untargeted - Plot 1

- **Plot 2:** il grafico in Figura 4.22 mostra l'efficacia del sistema di difesa rispetto all'attacco PGD untargeted, al variare del parametro $epsilon_step$: come atteso, l'accuracy del classificatore NN1+Detectors è soggetta ad un netto miglioramento rispetto a quella del classificatore NN1, raggiungendo il valore di circa 1.0 (per tutti i valori di $epsilon_step$), dovuto al fatto che i detectors scartano quasi tutti i campioni adversarial. Dunque, è possibile affermare che, per questo tipo di attacco, il sistema complessivo NN1+Detectors gode di ottime prestazioni.

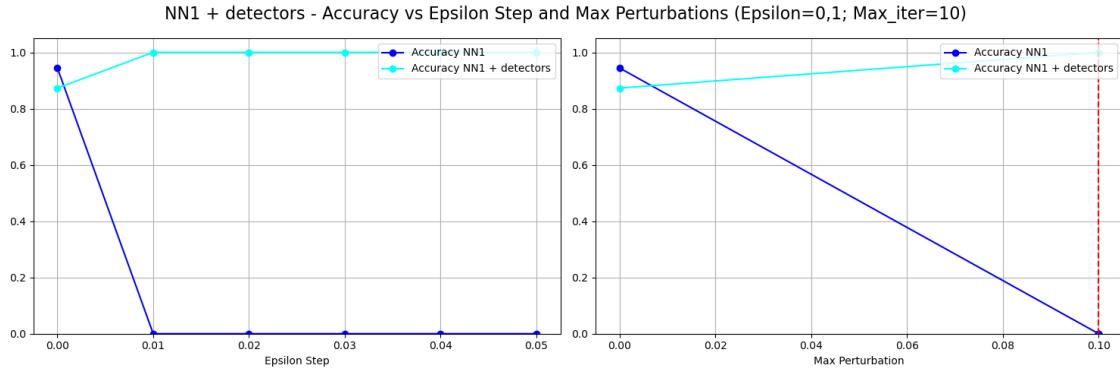


Figure 4.22: NN1+Detectors - PGD untargeted - Plot 2

- **Plot 3:** il grafico in Figura 4.23 mostra l'efficacia del sistema di difesa rispetto all'attacco PGD untargeted, al variare del parametro max_iter : come atteso, l'accuracy del classificatore NN1+Detectors è soggetta ad un netto miglioramento rispetto a quella del classificatore NN1, raggiungendo il valore di circa 1.0 (per tutti i valori di max_iter), dovuto al fatto che i detectors scartano quasi tutti i campioni adversarial. Dunque, è possibile affermare che, per questo tipo di attacco, il sistema complessivo NN1+Detectors gode di ottime prestazioni.

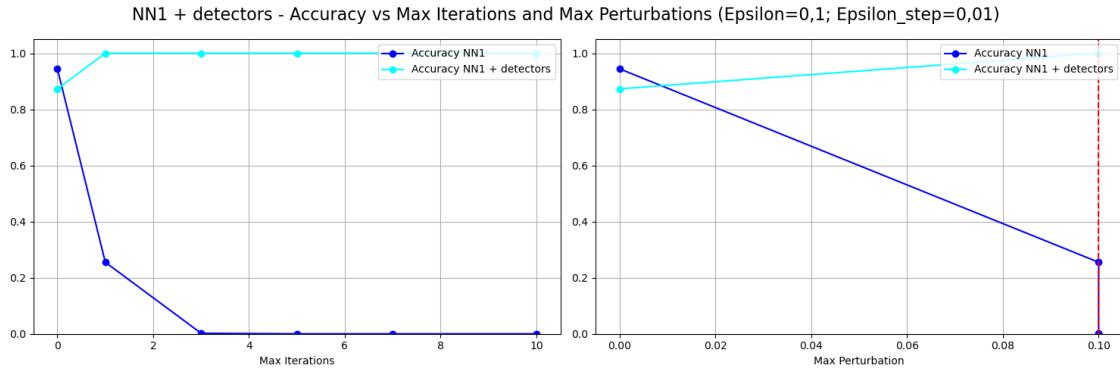


Figure 4.23: NN1+Detectors - PGD untargeted - Plot 3

4.9.2 PGD targeted

- **Plot 1:** il grafico in Figura 4.24 mostra l'efficacia del sistema di difesa rispetto all'attacco PGD targeted, al variare del parametro $epsilon$: come atteso, l'accuracy del classificatore NN1+Detectors è soggetta ad un netto miglioramento rispetto a quella del classificatore NN1, raggiungendo il valore di circa 1.0 (per tutti i valori di $epsilon$), dovuto al fatto che i detectors scartano quasi tutti i campioni adversarial. Inoltre, è possibile notare che la targeted accuracy del classificatore NN1+Detectors raggiunge il valore di circa 0.0 (per tutti i valori di $epsilon$), sottolineando l'inefficacia degli attacchi. Dunque, è possibile affermare che, per questo tipo di attacco, il sistema complessivo NN1+Detectors gode di ottime prestazioni.

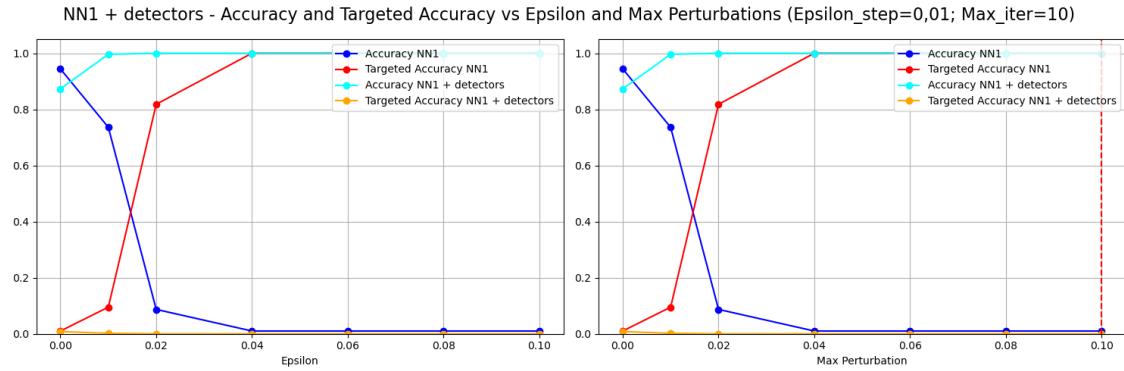


Figure 4.24: NN1+Detectors - PGD targeted - Plot 1

- Plot 2:** il grafico in Figura 4.25 mostra l'efficacia del sistema di difesa rispetto all'attacco PGD targeted, al variare del parametro *epsilon_step*: come atteso, l'accuracy del classificatore NN1+Detectors è soggetta ad un netto miglioramento rispetto a quella del classificatore NN1, raggiungendo il valore di circa 1.0 (per tutti i valori di *epsilon_step*), dovuto al fatto che i detectors scartano quasi tutti i campioni adversariali. Inoltre, è possibile notare che la targeted accuracy del classificatore NN1+Detectors raggiunge il valore di circa 0.0 (per tutti i valori di *epsilon_step*), sottolineando l'inefficacia degli attacchi. Dunque, è possibile affermare che, per questo tipo di attacco, il sistema complessivo NN1+Detectors gode di ottime prestazioni.

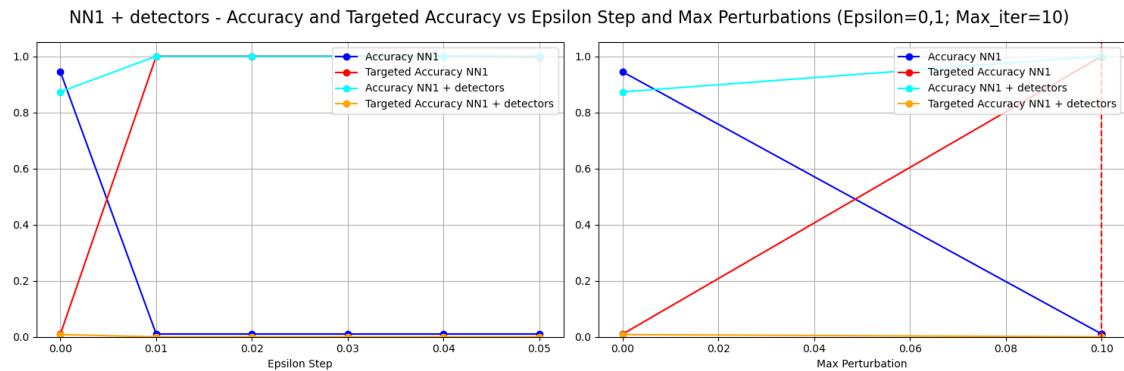


Figure 4.25: NN1+Detectors - PGD targeted - Plot 2

- Plot 3:** il grafico in Figura 4.26 mostra l'efficacia del sistema di difesa rispetto all'attacco PGD targeted, al variare del parametro *max_iter*: come atteso, l'accuracy del classificatore NN1+Detectors è soggetta ad un netto miglioramento rispetto a quella del classificatore NN1, raggiungendo il valore di circa 1.0 (per tutti i valori di *max_iter*), dovuto al fatto che i detectors scartano quasi tutti i campioni adversariali. Inoltre, è possibile notare che la targeted accuracy del classificatore NN1+Detectors raggiunge il valore di circa 0.0 (per tutti i valori di *max_iter*), sottolineando l'inefficacia degli attacchi. Dunque, è possibile affermare che, per questo tipo di attacco, il sistema complessivo NN1+Detectors gode di ottime prestazioni.

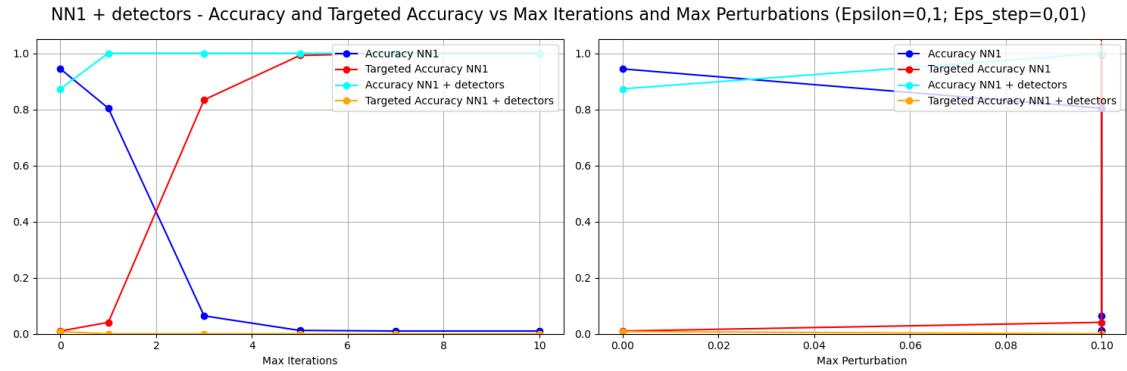


Figure 4.26: NN1+Detectors - PGD targeted - Plot 3

4.10 Prestazioni della rete sulle immagini adversarial DF

Per valutare le prestazioni della rete NN1+detectors sulle immagini adversarial generate dall'attacco DF, sono state tracciate diverse security evaluation curve. Nello specifico, è stata calcolata una curva per ciascuno dei principali parametri dell'attacco, calcolando l'*accuracy* al variare di esso. Questo approccio consente di valutare l'impatto specifico di ciascun parametro dell'attacco sulle prestazioni della rete e di avere un'analisi dettagliata sulla robustezza della rete in funzione della forza dell'attacco. In particolare, verranno tracciati i seguenti plot al variare dei seguenti parametri:

Plot	<i>epsilon</i>	<i>nb_grads</i>	<i>max_iter</i>
1	$[1^{-5}, 1^{-4}, 1^{-3}, 1^{-2}, 1^{-1}, 1]$	10	10
2	1^{-2}	$[5, 10, 20, 50]$	10
3	1^{-2}	10	$[1, 3, 5, 7, 10]$

4.10.1 DF untargeted

- **Plot 1:** il grafico in Figura 4.27 mostra l'efficacia del sistema di difesa rispetto all'attacco DF untargeted, al variare del parametro *epsilon*: come atteso, l'accuracy del classificatore NN1+Detectors è soggetta ad un netto miglioramento rispetto a quella del classificatore NN1, dovuto al fatto che i detectors scartano quasi tutti i campioni adversarial. In particolare, le performance del sistema crescono all'aumentare di *epsilon*. Ciò è dovuto al fatto che le perturbazioni applicate alle immagini diventano sempre più evidenti e quindi il detector riesce a classificare con più sicurezza i campioni come adversarial. Dunque, è possibile affermare che, per questo tipo di attacco, il sistema complessivo NN1+Detectors gode di buone prestazioni.

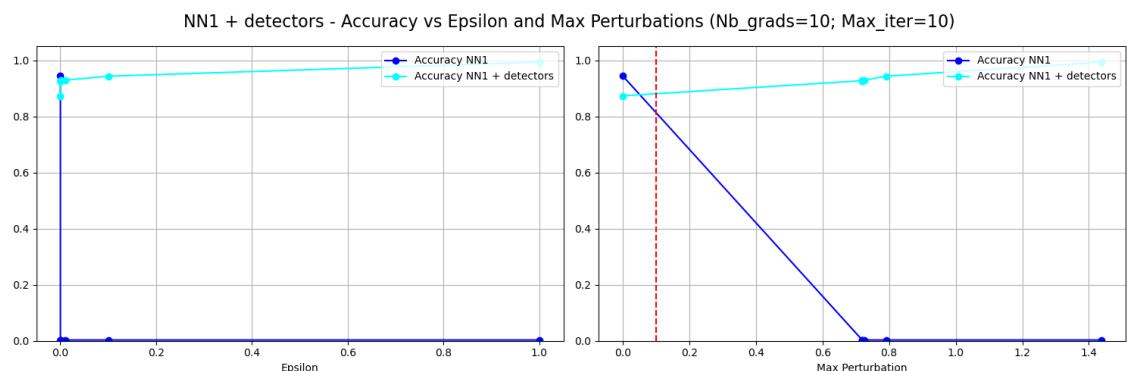


Figure 4.27: NN1+Detectors - DF untargeted - Plot 1

- **Plot 2:** il grafico in Figura 4.28 mostra l'efficacia del sistema di difesa rispetto all'attacco DF untargeted, al variare del parametro nb_grads : come atteso, l'accuracy del classificatore NN1+Detectors è soggetta ad un netto miglioramento rispetto a quella del classificatore NN1, dovuto al fatto che i detectors scartano quasi tutti i campioni adversarial. In particolare, le performance del sistema decrescono all'aumentare di nb_grads . Ciò è dovuto al fatto che le perturbazioni applicate alle immagini diventano sempre meno evidenti e quindi il detector riesce a classificare con meno sicurezza i campioni come adversarial. Dunque, è possibile affermare che, per questo tipo di attacco, il sistema complessivo NN1+Detectors gode di buone prestazioni.

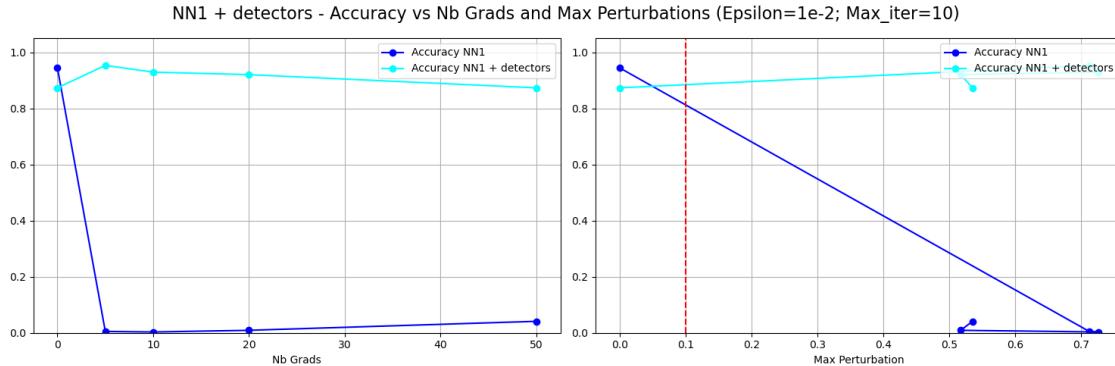


Figure 4.28: NN1+Detectors - DF untargeted - Plot 2

- **Plot 3:** il grafico in Figura 4.29 mostra l'efficacia del sistema di difesa rispetto all'attacco DF untargeted, al variare del parametro max_iter : come atteso, l'accuracy del classificatore NN1+Detectors è soggetta ad un netto miglioramento rispetto a quella del classificatore NN1, raggiungendo il valore di circa 0.95 (per tutti i valori di max_iter), dovuto al fatto che i detectors scartano quasi tutti i campioni adversarial. Dunque, è possibile affermare che, per questo tipo di attacco, il sistema complessivo NN1+Detectors gode di buone prestazioni.

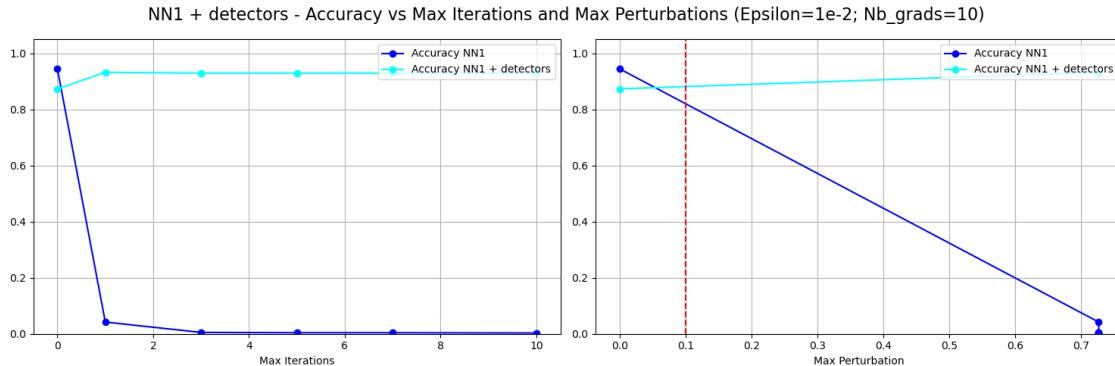


Figure 4.29: NN1+Detectors - DF untargeted - Plot 3

4.11 Prestazioni della rete sulle immagini adversarial CW

Per valutare le prestazioni della rete NN1+detectors sulle immagini adversarial generate dall'attacco CW, sono state tracciate diverse security evaluation curve. Nello specifico, è stata calcolata una curva per ciascuno dei principali parametri dell'attacco, calcolando sia l'*accuracy* che la *targeted_accuracy* al variare di esso. Questo approccio consente di valutare l'impatto specifico di ciascun parametro dell'attacco sulle prestazioni della rete e di avere un'analisi dettagliata sulla robustezza della rete in funzione della forza dell'attacco. In particolare, verranno tracciati i seguenti plot al variare dei seguenti parametri:

Plot	<i>confidence</i>	<i>learning_rate</i>	<i>max_iter</i>
1	[0.01, 0.1, 1]	0.01	3
2	0.1	[0.01, 0.05, 0.1]	3
3	0.1	0.01	[1, 3, 5]

Nota: per l'attacco CW è stato deciso di analizzare esclusivamente l'efficacia dell'attacco *untargeted*, in quanto dall'analisi effettuata sulla rete NN1 è emerso che la versione *targeted* dell'attacco non è in grado di compromettere efficacemente il modello, per cui si è valutato superfluo valutare la robustezza del sistema di difesa su un attacco già di per sé inefficace.

4.11.1 CW untargeted

- **Plot 1:** il grafico in Figura 4.30 mostra l'efficacia del sistema di difesa rispetto all'attacco CW untargeted, al variare del parametro *confidence*: come atteso, l'accuracy del classificatore NN1+Detectors è soggetta ad un netto miglioramento rispetto a quella del classificatore NN1, raggiungendo il valore di circa 1.0 (per tutti i valori di *confidence*), dovuto al fatto che i detectors scartano quasi tutti i campioni adversariali. Dunque, è possibile affermare che, per questo tipo di attacco, il sistema complessivo NN1+Detectors gode di ottime prestazioni.

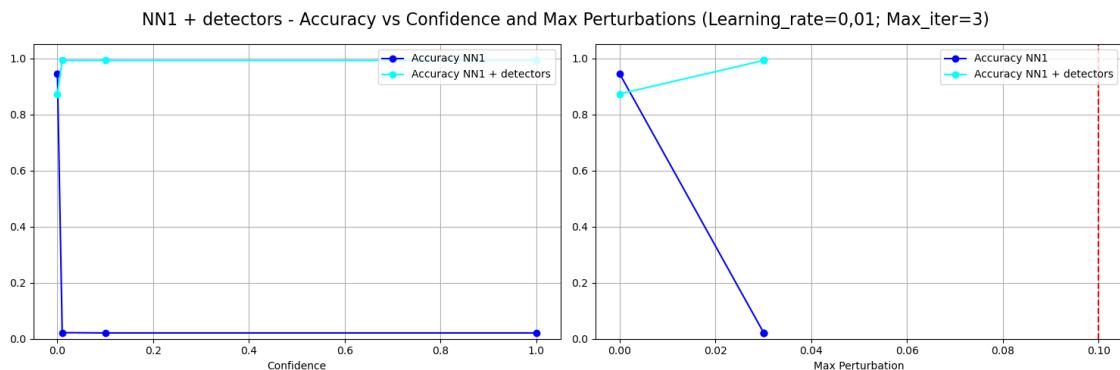


Figure 4.30: NN1+Detectors - CW untargeted - Plot 1

- **Plot 2:** il grafico in Figura 4.31 mostra l'efficacia del sistema di difesa rispetto all'attacco CW untargeted, al variare del parametro *learning_rate*: come atteso, l'accuracy del classificatore NN1+Detectors è soggetta ad un netto miglioramento rispetto a quella del classificatore NN1, raggiungendo il valore di circa 1.0 (per tutti i valori di *learning_rate*), dovuto al fatto che i detectors scartano quasi tutti i campioni adversariali. Dunque, è possibile affermare che, per questo tipo di attacco, il sistema complessivo NN1+Detectors gode di ottime prestazioni.

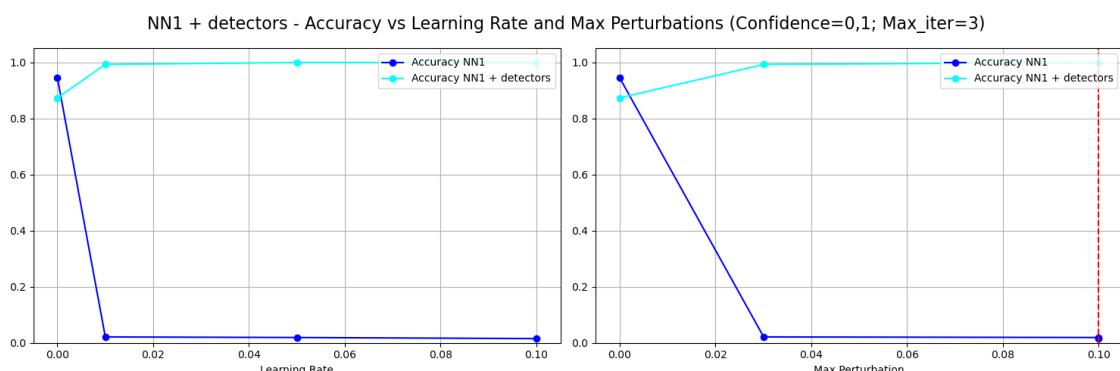


Figure 4.31: NN1+Detectors - CW untargeted - Plot 2

- **Plot 3:** il grafico in Figura 4.32 mostra l'efficacia del sistema di difesa rispetto all'attacco CW untargeted, al variare del parametro *max_iter*: come atteso, l'accuracy del classificatore NN1+Detectors è soggetta ad un netto miglioramento rispetto a quella del classificatore NN1, raggiungendo il valore di circa 1.0 (per tutti i valori di *max_iter*), dovuto al fatto che i detectors scartano quasi tutti i campioni adversarial. Dunque, è possibile affermare che, per questo tipo di attacco, il sistema complessivo NN1+Detectors gode di ottime prestazioni.

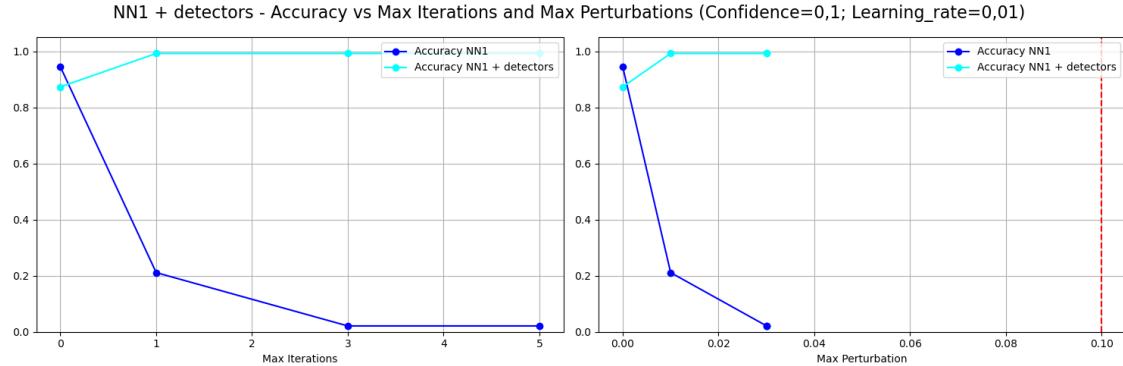


Figure 4.32: NN1+Detectors - CW untargeted - Plot 3

4.12 Riepilogo sull'analisi del sistema di difesa

In questa sezione viene presentato un breve riepilogo sull'efficacia degli attacchi sul sistema complessivo, composto dai detectors e dal classificatore NN1. Per ciascun tipo di attacco viene riportata l'**accuracy minima** (ed eventualmente la **targeted accuracy massima**) ottenuta, insieme alla relativa **perturbazione massima** osservata (nel caso in cui il valore massimo dell'accuracy sia stato ottenuto più volte con perturbazioni massime diverse, verrà considerata la perturbazione massima più piccola). Inoltre, per ogni attacco, viene riportata l'**efficacia del sistema di difesa**, che può essere: nulla(*), bassa(**), media(***), alta(****).

Attacco	Accuracy _{min} (\approx)	Targeted Accuracy _{max} (\approx)	Linf (\approx)	Efficacia
FGSM (untargeted)	1.0	/	0.1	****
FGSM (targeted)	1.0	0.0	0.1	****
BIM (untargeted)	1.0	/	0.1	****
BIM (targeted)	1.0	0.0	0.1	****
PGD (untargeted)	1.0	/	0.1	****
PGD (targeted)	1.0	0.0	0.1	****
DF (untargeted)	0.85	/	0.55	***
CW (untargeted)	1.0	/	0.1	****

Dall'analisi dei risultati riportati in tabella, si osserva che il sistema di difesa implementato risulta altamente efficace contro gli attacchi FGSM, BIM, PGD e CW, indipendentemente dall'intensità dell'attacco. Invece, per quanto riguarda l'attacco DF, l'efficacia del sistema risulta leggermente inferiore in corrispondenza di valori di *epsilon* bassi, mentre all'aumentare di *epsilon* il sistema riesce a migliorare le sue prestazioni, raggiungendo un'accuratezza pari a 1.0 per *epsilon*= 1.0. Dunque, nel caso di DF, l'efficacia del sistema di difesa tende a migliorare al crescere dell'intensità dell'attacco.

CHAPTER 5

CONCLUSIONE

L'**analisi della rete NN1** ha evidenziato una forte vulnerabilità del modello agli attacchi adversarial, con una drastica riduzione dell'accuracy anche per basse perturbazioni. Gli attacchi iterativi, come BIM e PGD, si sono rivelati i più efficaci, portando rapidamente l'accuracy vicino allo zero, sia in modalità untargeted che targeted. Gli attacchi one-shot, come FGSM, hanno mostrato un'efficacia minore, soprattutto negli scenari targeted. Invece, gli attacchi più sofisticati, come DF e CW, si sono dimostrati molto potenti in modalità untargeted, ma nel caso di DF non è stato possibile rispettare i vincoli imposti sulla norma Linf (l'attacco è stato comunque considerato nelle fasi successive del progetto perché alcuni campioni adversarial rientrano nei vincoli). Infine, l'efficacia dell'attacco CW targeted si è rivelata molto limitata, per cui questo tipo di attacco non è stato analizzato nelle fasi successive del progetto.

Per quanto riguarda la **trasferibilità**, gli attacchi generati sulla rete NN1 mantengono in parte la loro efficacia anche su un modello differente (NN2). In particolare, gli attacchi untargeted hanno evidenziato una buona trasferibilità, soprattutto per quanto riguarda gli attacchi iterativi. Invece, gli attacchi targeted hanno mostrato una trasferibilità molto bassa, indicando una forte dipendenza dall'architettura e dai pesi del modello originale.

Infine, il **sistema di difesa** implementato, basato su un insieme di detector specializzati, ha introdotto un miglioramento sostanziale della robustezza del modello. È risultato estremamente efficace nel rilevare e bloccare quasi tutti gli esempi avversari generati con attacchi FGSM, BIM, PGD e CW. Anche per l'attacco DF il sistema di difesa ha evidenziato delle buone prestazioni, seppur leggermente inferiori per perturbazione di bassa intensità. Complessivamente, la sistema di difesa implementato riduce drasticamente gli errori sui campioni adversarial, seppur compromettendo leggermente le prestazioni sui dati clean, a causa della rilevazione di falsi positivi.

Attacco	Analisi della rete NN1		Trasferibilità	Sistema di difesa
	Accuracy _{min}	Targeted Accuracy _{max}		
FGSM (untargeted)	0.0	/	***	****
FGSM (targeted)	0.1	0.45	*	****
BIM (untargeted)	0.0	/	****	****
BIM (targeted)	0.0	1.0	*	****
PGD (untargeted)	0.0	/	****	****
PGD (targeted)	0.0	1.0	*	****
DF (untargeted)	0.0	/	**	***
CW (untargeted)	0.0	/	***	****
CW (targeted)	0.4	0.01	/	/

Table 5.1: Tabella riepilogativa delle analisi effettuate. La trasferibilità e l'efficacia del sistema di difesa possono essere: nulla(*), bassa(**), media(***) o alta(****).

LIST OF FIGURES

1.1	Esempio di <i>Resize</i> e <i>CenterCrop</i> applicato alle immagini.	6
1.2	Esempio di attacco FGSM al variare della forza dell'attacco.	8
1.3	Esempio di attacco BIM al variare della forza dell'attacco.	8
1.4	Esempio di attacco PGD al variare della forza dell'attacco.	9
1.5	Esempio di attacco DF al variare della forza dell'attacco.	10
1.6	Esempio di attacco CW al variare della forza dell'attacco.	10
2.1	NN1 - FGSM untargeted - Plot 1	13
2.2	NN1 - FGSM targeted - Plot 1	13
2.3	NN1 - BIM untargeted - Plot 1	14
2.4	NN1 - BIM untargeted - Plot 2	14
2.5	NN1 - BIM untargeted - Plot 3	15
2.6	NN1 - BIM targeted - Plot 1	15
2.7	NN1 - BIM targeted - Plot 2	15
2.8	NN1 - BIM targeted - Plot 3	16
2.9	NN1 - PGD untargeted - Plot 1	16
2.10	NN1 - PGD untargeted - Plot 2	17
2.11	NN1 - PGD untargeted - Plot 3	17
2.12	NN1 - PGD targeted - Plot 1	17
2.13	NN1 - PGD targeted - Plot 2	18
2.14	NN1 - PGD targeted - Plot 3	18
2.15	NN1 - DF untargeted - Plot 1	19
2.16	NN1 - DF untargeted - Plot 2	19
2.17	NN1 - DF untargeted - Plot 3	19
2.18	Esempio di istogramma delle perturbazioni massime dell'attacco DF.	20
2.19	NN1 - CW untargeted - Plot 1	21
2.20	NN1 - CW untargeted - Plot 2	21
2.21	NN1 - CW untargeted - Plot 3	21
2.22	NN1 - CW targeted - Plot 1	22
2.23	NN1 - CW targeted - Plot 2	22
2.24	NN1 - CW targeted - Plot 3	22
3.1	NN2 - FGSM untargeted - Plot 1	26
3.2	NN2 - FGSM targeted - Plot 1	26
3.3	NN2 - BIM untargeted - Plot 1	27
3.4	NN2 - BIM untargeted - Plot 2	27
3.5	NN2 - BIM untargeted - Plot 3	28
3.6	NN2 - BIM targeted - Plot 1	28
3.7	NN2 - BIM targeted - Plot 2	29
3.8	NN2 - BIM targeted - Plot 3	29
3.9	NN2 - PGD untargeted - Plot 1	30

LIST OF FIGURES

3.10 NN2 - PGD untargeted - Plot 2	30
3.11 NN2 - PGD untargeted - Plot 3	31
3.12 NN2 - PGD targeted - Plot 1	31
3.13 NN2 - PGD targeted - Plot 2	32
3.14 NN2 - PGD targeted - Plot 3	32
3.15 NN2 - DF untargeted - Plot 1	33
3.16 NN2 - DF untargeted - Plot 2	33
3.17 NN2 - DF untargeted - Plot 3	34
3.18 NN2 - CW untargeted - Plot 1	34
3.19 NN2 - CW untargeted - Plot 2	35
3.20 NN2 - CW untargeted - Plot 3	35
4.1 Architettura del sistema di difesa implementato.	37
4.2 Architettura di ogni singolo detector	38
4.3 Matrice di confusione FGSM Detector	41
4.4 Curva ROC FGSM Detector	41
4.5 Matrice di confusione BIM Detector	42
4.6 Curva ROC BIM Detector	42
4.7 Matrice di confusione PGD Detector	43
4.8 Curva ROC PGD Detector	43
4.9 Matrice di confusione DF Detector	44
4.10 Curva ROC DF Detector	44
4.11 Matrice di confusione CW Detector	45
4.12 Curva ROC CW Detector	45
4.13 NN1+Detectors - FGSM untargeted - Plot 1	47
4.14 NN1+Detectors - FGSM targeted - Plot 1	48
4.15 NN1+Detector - BIM untargeted - Plot 1	48
4.16 NN1+Detectors - BIM untargeted - Plot 2	49
4.17 NN1+Detectors - BIM untargeted - Plot 3	49
4.18 NN1+Detectors - BIM targeted - Plot 1	50
4.19 NN1+Detectors - BIM targeted - Plot 2	50
4.20 NN1+Detectors - BIM targeted - Plot 3	51
4.21 NN1+Detectors - PGD untargeted - Plot 1	51
4.22 NN1+Detectors - PGD untargeted - Plot 2	52
4.23 NN1+Detectors - PGD untargeted - Plot 3	52
4.24 NN1+Detectors - PGD targeted - Plot 1	53
4.25 NN1+Detectors - PGD targeted - Plot 2	53
4.26 NN1+Detectors - PGD targeted - Plot 3	54
4.27 NN1+Detectors - DF untargeted - Plot 1	54
4.28 NN1+Detectors - DF untargeted - Plot 2	55
4.29 NN1+Detectors - DF untargeted - Plot 3	55
4.30 NN1+Detectors - CW untargeted - Plot 1	56
4.31 NN1+Detectors - CW untargeted - Plot 2	56
4.32 NN1+Detectors - CW untargeted - Plot 3	57