



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه صنعتی اصفهان  
دانشکده مهندسی برق و کامپیوتر

## آتنا: چارچوبی برای دفاع در برابر حملات تخاصمی با استفاده از مدل های مولد مشروط بر کلاس

پایان نامه کارشناسی ارشد مهندسی کامپیوتر - هوش مصنوعی و رباتیکز

آرین تشکر

استاد راهنما

دکتر محمد حسین منشئی



دانشگاه صنعتی اصفهان

دانشکده مهندسی برق و کامپیوتر

پایان نامه کارشناسی ارشد رشته مهندسی کامپیوتر - هوش مصنوعی و رباتیکز آقای آراین  
تشکر

تحت عنوان

آتنا: چارچوبی برای دفاع در برابر حملات تخاصمی با استفاده از مدل های مولد مشروط بر  
کلاس

در تاریخ ۱۴۰۳/۰۱/۰۱ توسط کمیته تخصصی زیر مورد بررسی و تصویب نهایی قرار گرفت:

دکتر محمد حسین منشی

۱ - استاد راهنمای پایان نامه

ایکس وای زی

۳- استاد داور

ایکس وای زی

۴- استاد داور

دکتر بهزاد نظری

سرپرست تحصیلات تکمیلی دانشکده

### **تشکر و قدردانی**

قدردان راهنمایی های ارزنده استاد گرانقدر جناب آقای دکتر محمد حسین منشئی هستم که بدون شک پشتیبانی و راهنمایی هایشان روشنگر مسیر انجام پژوهش بود.

کلیه حقوق مالکیت مادی و معنوی مربوط به این پایان نامه متعلق به دانشگاه صنعتی اصفهان و پدیدآورندگان است. این حقوق توسط دانشگاه صنعتی اصفهان و بر اساس خط مشی مالکیت فکری این دانشگاه، ارزش گذاری و سهم بندی خواهد شد.  
هر گونه بهره برداری از محتوا، نتایج یا اقدام برای تجاری سازی دستاوردهای این پایان نامه تنها با مجوز کتبی دانشگاه صنعتی اصفهان امکان پذیر است.

تقدیم بہ

پدر و مادر عزیزم

مستحکم ترین پشتوانہ و حامیانم کہ وجودشان نشاء آرامش و انگیزہ است...

و خواہرم

مشوقان و ہمرانان ہمیشگی...

## فهرست مطالب

صفحه	عنوان
هشت	فهرست مطالب
ده	فهرست شکل ها
دوازده	فهرست جداول
سیزده	فهرست اختصارات
۱	چکیده
فصل اول : مقدمه	
۲	۱-۱ اهمیت مسئله
۳	۲-۱ ساختار گزارش
فصل دوم : پیشینه پژوهش و مبانی هوش مولد	
۴	۱-۲ مقدمه
۴	۲-۲ حملات تخاصمی
۶	۱-۲-۲ دسته بندی حملات تخاصمی
۶	۱-۲-۲-۱ رویه ی حمله
۱۰	۱-۲-۲-۲ FGSM حمله
۱۲	۱-۲-۲-۳ CW حمله
۱۴	۱-۲-۲-۴ PGD حمله
۱۶	۳-۲ روش های دفاع در برابر حملات تخاصمی
۱۶	۱-۳-۲ روش های پاکسازی حمله
۲۵	۲-۳-۲ روش های تشخیص حمله
۳۴	۴-۲ مختصری در مورد هوش مولد
۳۴	۱-۴-۲ شبکه های مولد تخاصمی
۳۸	۱-۴-۲-۱ cGAN
۴۰	۱-۴-۲-۲ ACGAN
۴۱	۱-۴-۲-۳ ReACGAN
۴۲	۲-۴-۲ مدل های انتشاری
۴۴	۱-۲-۴-۲ مدل های انتشاری هدایت شده (مشروط)



### فصل سوم : پیشنهاد روشی نوین برای مقابله با حملات تخاصمی

۴۷	۱-۳ مقدمه
۴۷	۲-۳ بیان مسئله
۴۸	۳-۳ روش پیشنهادی
۴۸	۱-۳-۳ استفاده از مولد های قوی تر
۴۹	۲-۳-۳ روش بهبود یافته برای تشخیص حمله
۵۳	۳-۳-۳ روش بهبود یافته برای پاکسازی حمله

### فصل چهارم : شبیه سازی و نتایج ارزیابی

۵۵	۱-۴ مقدمه
۵۵	۲-۴ روش شبیه سازی
۵۷	۳-۴ نتایج آزمون مدل ها
۵۷	۱-۳-۴ سنجش عملکرد مدل های مولد
۵۸	۲-۳-۴ سنجش عملکرد تشخیص حملات
۶۱	۳-۳-۴ سنجش عملکرد پاکسازی حملات
۶۱	۴-۴ بررسی و مقایسه نتایج

### فصل پنجم : نتیجه گیری و پیشنهادها

۶۲	۱-۵ نتیجه گیری
۶۲	۲-۵ پیشنهادها
۶۳	واژه نامه انگلیسی به فارسی
۶۹	واژه نامه فارسی به انگلیسی
۷۵	مراجع
۸۰	چکیده انگلیسی

## فهرست شکل‌ها

۱-۱	نمونه حمله به یک سیستم خودران	۳
۱-۲	حمله تخصصی به یک مدل تشخیص چهره	۵
۲-۲	حمله تخصصی به یک متن	۵
۳-۲	نمونه خط لوله یک سیستم ماشین خودران	۷
۴-۲	نمونه حمله FGSM	۱۱
۵-۲	نمودار حساسیت موفقیت حمله و اندازه نویز تخصصی بر حسب $c$ در حمله CW	۱۳
۶-۲	نمونه‌های تخصصی تولید شده توسط $L_2$ -CW	۱۴
۷-۲	چارچوب تقطیر دفاعی	۱۹
۸-۲	چارچوب DeepCloak	۲۰
۹-۲	چارچوب MagNet	۲۱
۱۰-۲	چارچوب Defense-GAN	۲۲
۱۱-۲	چارچوب APE-GAN	۲۳
۱۲-۲	چارچوب ER-classifier	۲۴
۱۳-۲	چارچوب DiffPure	۲۴
۱۴-۲	معماری تشخیص‌دهنده Metzen	۲۵
۱۵-۲	دورنمای نحوه آموزش f-AnoGAN	۲۷
۱۶-۲	f-AnoGAN در زمان تست	۲۷
۱۷-۲	عملکرد چارچوب ACGAN-ADA در زمان تست	۲۹
۱۸-۲	تشکیل نمونه‌های تخصصی در اطراف خمیدگی‌های مرز تصمیم یک دسته‌بند	۳۲
۱۹-۲	عملکرد SID در زمان تست	۳۳
۲۰-۲	دورنمای شبکه‌های مولد تخصصی	۳۴
۲۱-۲	معماری شبکه DCGAN	۳۶
۲۲-۲	نمونه‌های تولید شده توسط یک GAN معمولی	۳۷
۲۳-۲	معماری cGAN	۳۸
۲۴-۲	نمونه‌های تولید شده توسط cGAN	۳۹
۲۵-۲	معماری ACGAN	۴۰
۲۶-۲	فضای embedding تصاویر در یک ReACGAN	۴۱

۴۲	۲-۲۷ زنجیره مارکوف نظیر یک DDPM
۴۴	۲-۲۸ نمونه‌های تولید شده توسط DDPM
۴۵	۲-۲۹ نمونه‌های تولید شده توسط Classifier Guided Diffusion Model
۴۶	۲-۳۰ نمونه‌های تولید شده توسط Classifier-Free Guided Diffusion Model
۵۱	۳-۱ نحوه‌ی تشخیص حمله توسط ARCANE-GAN
۵۱	۳-۲ نحوه‌ی تشخیص حمله توسط ARCANE-Diff
۵۲	۳-۳ دورنمای ساختار تصمیم‌گیرنده‌ی نهایی مبتنی بر XGBoost
۵۴	۳-۴ نحوه‌ی پاکسازی حمله توسط ARCANE-GAN
۵۴	۳-۵ نحوه‌ی پاکسازی حمله توسط ARCANE-Diff
۶۰	۴-۱ اهمیت ویژگی‌های استفاده شده در XGBoost

## فهرست جداول

۹	۱-۲ مقایسه حملات جعبه سیاه و جعبه سفید
۱۵	۲-۲ مقایسه حملات FGSM، PGD و CW
۵۸	۱-۴ عملکرد مدل های مولد روی CIFAR۱۰ و Tiny-ImageNet
۵۹	۲-۴ عملکرد تشخیص ARCANE-Diff و ARCANE-GAN طبق معیار pAUC-0.2

## فهرست اختصارات

### **C**

cGAN ..... Conditional Generative Adversarial Network

CW ..... Carlini-Wagner

### **D**

D2D-CE ..... Data-to-Data Cross-Entropy

### **F**

FGSM ..... Fast Gradient Sign Method

FID ..... Fréchet Inception Distance

### **G**

GDA ..... Gaussian Discriminant Analysis

GDM ..... Guided Diffusion Model

GI-AT ..... Geometry-aware Instance-reweighted Adversarial Training

### **I**

IS ..... Inception Score

### **J**

JS ..... Jensen-Shannon

## **K**

KL ..... Kullback-Leibler

## **P**

PGD ..... Projected Gradient Descent

## **R**

ROC ..... Receiver Operating Characteristic

## **Y**

YOPO ..... You Only Propagate Once

## چکیده

این یک متن نمونه است... این یک متن نمونه است... این یک متن نمونه است... این یک متن نمونه است... این یک متن نمونه است... این یک متن نمونه است... این یک متن نمونه است... این یک متن نمونه است... این یک متن نمونه است... این یک متن نمونه است...

**کلمات کلیدی:** ۱- یادگیری ماشین، ۲- یادگیری عمیق، ۳- مدل های مولد<sup>۱</sup>، ۴- دسته بند ها<sup>۲</sup>، ۵- حملات تخاصمی<sup>۳</sup>

---

<sup>1</sup>Generative Models

<sup>2</sup>Classifiers

<sup>3</sup>Adversarial Attacks

# فصل اول

## مقدمه

در این فصل به طور مختصر ابتدا به بیان اهمیت دفاع در برابر حملات تخاصمی روی دسته‌بندها پرداخته خواهد شد و سپس در ادامه، ساختار گزارش پیش رو بسط داده خواهد شد.

### ۱-۱ اهمیت مسئله

امروزه یادگیری عمیق<sup>۱</sup> به عنوان ابزاری قدرت مند و بهینه برای حل گسترده‌ی وسیعی از مسائل پیچیده روزمره شناخته می‌شود که حل آن‌ها با روش‌های یادگیری ماشین سنتی بسیار دشوار و بعضاً غیر ممکن بود. در سال‌های اخیر، یادگیری عمیق چنان دستخوش پیشرفت‌های ژگرفی شده که اکنون قادر است در عدیده‌ای از اهداف یادگیری، از کارایی انسان نیز پیشی بگیرد. با توجه به گسترش روز افزون استفاده از هوش مصنوعی - و به خصوص یادگیری عمیق- در مصارف روزانه و صنایع، اهمیت اندیشیدن تسهیلاتی برای مقابله با حملات سایبری احتمالی به چنین سیستم‌هایی نیز به تبع دو چندان شده است. به عنوان مثال، در [۱] نشان داده شده است که می‌توان به یک سیستم ماشین خودران<sup>۲</sup> که توسط یک کنترل کننده‌ی هوش مصنوعی اداره می‌شود، در کمتر از ۲ ثانیه حمله و آن را از مسیر خارج کرد. در [۲] نمونه دیگری از یک حمله به سیستم ماشین خودران نشان

---

<sup>۱</sup>Deep Learning

<sup>۲</sup>Self-driving Car



داده شده است که در آن می‌توان این سیستم را در تشخیص علائم رانندگی دچار خطا کرد. این حمله در شکل ۱-۱ نشان داده شده است. همچنین در [۳] نمونه عملی یک حمله به سه نمونه الگوریتم تاجر خودکار<sup>۱</sup> حمله شده است که در طی آن سیستم تاجر خودکار مبتنی بر مدل های یادگیری ماشین دچار خطا در پیشبینی قیمت آینده یک سهم می شوند.

دسته خاصی از حملات سایبری<sup>۲</sup> اعمال پذیر روی دسته‌بند های مبتنی بر یادگیری ماشین، حملات تخصصی هستند که تمرکز اصلی این تحقیق می باشند. در فصل ۲ به تفصیل راجع به این حملات توضیح داده خواهد شد.

## ۲-۱ ساختار گزارش

در ادامه این گزارش، در فصل ۲ پیشینه پژوهش و مبانی لازم برای درک بهتر فصل‌های آتی مورد بررسی قرار خواهند گرفت. در فصل ۳ روش پیشنهادی مسئله مورد بررسی به طور مشخص مطرح شده و روش پیشنهادی برای حل آن بیان خواهد شد. سپس در فصل ۴ نتایج شبیه‌سازی و مقایسه‌های لازم با استفاده از سنجه<sup>۳</sup> های مناسب مورد بررسی قرار خواهند گرفت. در نهایت در فصل ۵ به جمع‌بندی نهایی و جهت های احتمالی تحقیقات آینده پرداخته خواهد شد.



شکل ۱-۱: نمونه حمله به یک سیستم خودران. ردیف بالا نشان دهنده تصاویر با تشخیص برجسته درست توسط مدل دسته‌بند می باشند. ردیف پایین همان تصاویر به همراه نویزی نامحسوس برای چشم غیر مسلح می باشند که باعث ایجاد خروجی اشتباه توسط مدل دسته‌بند می شود [۲].

<sup>1</sup>Auto Trader

<sup>2</sup>Cyber Attacks

<sup>3</sup>Metric

## فصل دوم

### پیشینه پژوهش و مبانی هوش مولد

#### ۱-۲ مقدمه

در این فصل به بیان مقدماتی در مورد حملات تخاصمی، روش های دفاع در برابر آن ها پرداخته خواهد شد. سپس مبانی هوش مولد و به خصوص مدل های مولد تخاصمی و مدل های انتشاری به تفصیل مورد بررسی قرار خواهند گرفت.

#### ۲-۲ حملات تخاصمی

حملات تخاصمی به تکنیک هایی در حوزه یادگیری ماشین گفته می شود که برای هدف خاص فریب دادن مدل های مختلف به کمک توسعه و اعمال ورودی های آسیب زننده به این مدل ها طراحی شده اند [۴-۶]. حملات تخاصمی عموماً طی فرایندی تحت عنوان **اختلال تخاصمی**<sup>۱</sup> ایجاد می شوند. این پروسه شامل افزودن مقادیر ناچیزی نویز<sup>۲</sup> به ورودی مدل دسته بند قربانی<sup>۳</sup> می شود که با وجود نامحسوس بودن به چشم غیر مسلح، باعث اختلال در خروجی دسته بند خواهند شد [۷]. تحقیقات انجام پذیرفته روی این دسته از حملات عموماً

---

<sup>۱</sup>Adversarial Perturbation

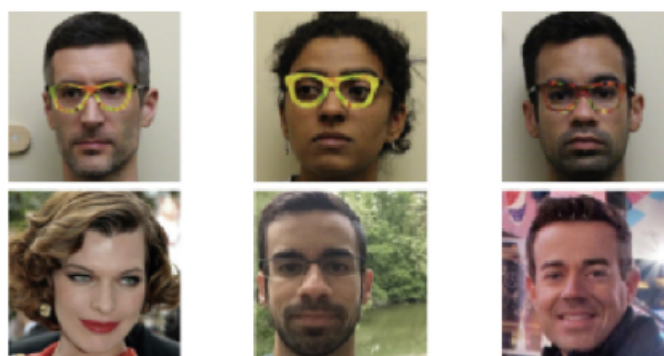
<sup>۲</sup>Noise

<sup>۳</sup>Victim

در حوزه تصویر می باشد که با تغییرات جزئی و نامحسوس روی مقادیر پیکسل<sup>۱</sup> های به خصوص، خروجی یک دسته‌بند تصویر تغییر خواهد کرد. با این وجود، از آنجایی که اکثر روش های طراحی حملات تخاصمی از ماهیت ورودی مدل به طور مستقیم برای طراحی حمله استفاده نمی کنند، این روش ها می توانند قابل تعمیم به تمامی دسته‌بندها، چه در حوزه تصویر و چه خارج از آن، باشند [۵، ۸]. در شکل ۱-۲ نمونه‌ای از یک حمله تخاصمی به یک مدل تشخیص چهره را می توان مشاهده کرد. با قرار دادن یک عینک حاوی نویز تخاصمی روی صورت افرادی که در سطر بالا قرار دارند، مدل تشخیص چهره، چهره‌ی این افراد را به اشتباه به عنوان افراد نظیر در سطر پایین تشخیص داده است. همچنین در ۲-۲ نمونه ای از یک حمله تخاصمی به یک مدل دسته‌بند متن برای وظیفه تحلیل احساسات<sup>۲</sup> را می توان مشاهده کرد.

به طور رسمی یک حمله تخاصمی را می توان طبق تعریف ارائه شده در [۹] بررسی کرد. فرض کنیم که مدل دسته‌بند  $f(\cdot)$  را در اختیار داشته باشیم که خروجی آن روی یک نمونه ورودی  $x$  یک توزیع آماری روی تمامی کلاس های ممکن باشد. کلاس تشخیص داده شده توسط این مدل در معادله ۱-۲ نمایش داده شده است.

$$y = \arg_c \max f(x) \quad (1-2)$$



شکل ۱-۲: نمونه ی حمله تخاصمی به یک مدل تشخیص چهره [۷]. افراد سطر بالا به اشتباه توسط مدل تشخیص چهره به فرد متناظر در سطر پایین تشخیص داده شده اند.

**Task:** sentiment analysis. **Classifier:** CNN. **Original label:** 99.8% negative. **Adversarial label:** 81.0% positive.

**Text:** I love these awful awf ul 80's summer camp movies. The best part about "Party Camp" is the fact that it literally literally has no No plot. The cliches clichs here are limitless: the nerds vs. the jocks, the secret camera in the girls locker room, the hikers happening upon a nudist colony, the contest at the conclusion, the secretly horny camp administrators, and the embarrassingly embarrassingy foolish foollish sexual innuendo littered throughout. This movie will make you laugh, but never intentionally. I repeat, never.

شکل ۲-۲: یک مدل دسته‌بند متن مبتنی بر CNN که با تغییرات جزئی در متن نظر ارسال شده، با وجود حفظ محتوای کلی نظر، دچار اشتباه شده است [۷].

<sup>1</sup>Pixel

<sup>2</sup>Sentiment Analysis

اکنون یک نمونه تخصصی<sup>۱</sup> به ورودی دستکاری شده  $\hat{x}$  گفته می شود که طبق معادلات ۲-۲ و ۳-۲ از افزودن مقدار ناچیزی نويز  $\delta$  به ورودی اولیه  $x$  به دست می آید به طوری که توزیع آماری خروجی مدل و یا کلاس تشخیص داده شده توسط مدل را طوری تغییر دهد که با حالت اولیه متفاوت باشد.

$$\hat{x} = x + \delta, \quad (2-2)$$

$$s.t. \|\delta\| < \epsilon, f(\hat{x}) \neq f(x) \vee \hat{y} = \arg \max_c f(\hat{x}) \neq y \quad (3-2)$$

## ۱-۲-۲ دسته بندی حملات تخصصی

امنیت یک مدل یادگیری ماشین با توجه به اهداف خصمانه مورد نظر و قابلیت های مهاجم<sup>۲</sup> ارزیابی می شود [۴، ۵]. پیش از پرداختن به دسته بندی حملات تخصصی، ابتدا کمی راجع به مدل های تهدید<sup>۳</sup> در حوزه یادگیری ماشین با توجه به قدرت و دسترسی مهاجم صحبت خواهد شد.

### ۱-۱-۲-۲ رویه ی حمله

رویه ی حمله<sup>۴</sup> عبارتست که به تمام روش های ممکن موجود برای یک مهاجم برای حمله به یک سیستم اطلاق می شود. یک سیستم تصمیم گیرنده مبتنی بر یادگیری ماشین را می توان عملاً به عنوان یک خط لوله<sup>۵</sup> برای پردازش داده های ورودی متصور شد. بدین ترتیب، دنباله ای از عملیات ساده روی داده های ورودی در زمان استفاده از یک مدل یادگیری ماشین را می توان به صورت زیر خلاصه کرد:

۱. جمع آوری داده های ورودی از سنسورها و یا انبارهای داده

۲. انتقال داده های جمع آوری شده در مرحله قبل به دامنه دیجیتال

۳. پردازش داده های دیجیتال برای تبدیل آنها به قالب قابل استفاده توسط مدل یادگیری ماشین و دریافت خروجی از مدل

۴. اتخاذ تصمیم بر اساس خروجی مدل

نمونه چنین دنباله ای را می توان در شکل ۲-۳ مشاهده کرد.

در این خط لوله، سیستم ابتدا با استفاده از دوربین های نصب شده در اقصی نقاط خودرو، تصاویری را به عنوان ورودی دریافت می کند. این تصاویر طی عملیات پیش پردازش مناسب به فرمت قابل استفاده برای مدل

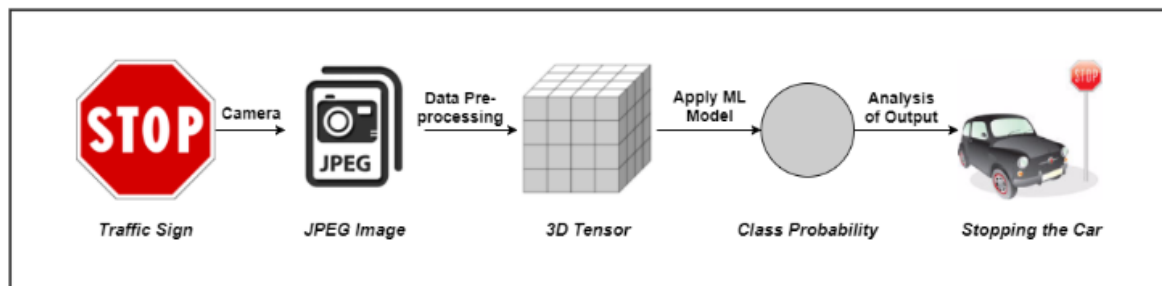
<sup>1</sup>Adversarial Sample

<sup>2</sup>Attacker

<sup>3</sup>Threat Models

<sup>4</sup>Attack Surface

<sup>5</sup>Pipeline



شکل ۲-۳: نمونه خط لوله یک سیستم ماشین خودران [۴]

یادگیری ماشین در می آیند (در این مثال خاص یک تنسور<sup>۱</sup> سه بعدی از مقادیر پیکسل های تصاویر دریافت). مدل پس از استخراج ویژگی از ورودی های دریافتی، خروجی مطلوب را تولید می کند (به عنوان مثال، احتمال مشاهده علامت ایست) و در نهایت یک تصمیم گیرنده بر اساس این خروجی، تصمیمی را اتخاذ می کند (در این مثال، توقف خودرو). در این مثال خاص، رویه ی حمله را می توان با توجه به خط لوله پردازش اطلاعات ورودی تعریف کرد. به طور دقیق تر، یک مهاجم می تواند با ایجاد اختلال در هر یک از مراحل جمع آوری و یا پردازش مدل قربانی را مسموم و در اثر آن، خروجی نهایی مدل را دستکاری کند.

سناریوهای اصلی حملات براساس رویه ی حمله مورد استفاده به شرح زیر می باشد [۴-۷، ۱۰]:

۱. **حمله گریزانه**<sup>۲</sup>: معمول ترین دسته ی حملات خصمانه. در این حالت مهاجم در صدد گریز از یک سیستم امنیتی به وسیله ی اعمال تغییرات در نمونه های ورودی در زمان تست، بر می آید. در این حالت هیچ پیش فرضی در رابطه با دسترسی مهاجم به داده های آموزشی مدل، وجود ندارد.

۲. **حمله مسموم کننده**<sup>۳</sup>: در این دسته از حملات که در زمان آموزش مدل قابل پیاده سازی هستند، مهاجم تلاش می کند که داده های آموزشی مدل را پیش از فرایند یادگیری، مسموم کند. به عبارت دقیق تر، مهاجم با افزودن نمونه های آموزشی جدیدی که با دقت و با هدف خاصی برای انحراف مدل آموزش دیده شده ی نهایی طراحی شده اند، به مجموعه داده های آموزشی، سعی می کند که کل فرایند آموزش مدل را مختل و یا خروجی آن را پس از اتمام آموزش دچار تغییرات نامطلوب نماید. واضح است که در این دسته از حملات فرض شده است که مجموعه داده آموزشی مدل قربانی در دسترس مهاجم قرار دارد.

۳. **حمله اکتشافی**<sup>۴</sup>: این حملات که با پیش فرض دسترسی جعبه سیاه<sup>۵</sup> به مدل پیاده سازی می شوند، برخلاف دسته پیشین، تاثیری روی مجموعه داده های آموزشی ندارند. در یک حمله اکتشافی، مهاجم سعی می کند

<sup>1</sup>Tensor

<sup>2</sup>Evasion Attack

<sup>3</sup>Poisoning Attack

<sup>4</sup>Exploratory Attack

<sup>5</sup>Black Box

با داشتن دسترسی محدود به مدل، تمام اطلاعات ممکن را راجع به الگوریتم یادگیری استفاده شده در سیستم مورد حمله و الگوهای احتمالی موجود در مجموعه داده‌های آموزشی را دریافت کند.

برای تعریف مدل تهدید نیازمند آن هستیم که نوع دسترسی مهاجم به قربانی را نیز در نظر بگیریم [۴، ۵] اگر به مثال ماشین خودران باز گردیم، برای یک مهاجم قوی می‌توان دسترسی به معماری مدل استفاده شده و مجموعه داده‌های آموزشی را متصور شد در حالی که یک مهاجم ضعیف تر فقط احتمالاً به داده‌های زمان تست دسترسی دارد. با وجود این که هر دو مهاجم از یک رویه برای حمله به مدل استفاده می‌کنند، مهاجم اول به دلیل در اختیار داشتن اطلاعات کامل تر، قوی تر تلقی می‌شود.

بدین ترتیب، حمله‌ها را می‌توان بر اساس گستره دسترسی مهاجم به دو دسته مهم حملات جعبه سیاه و جعبه سفید<sup>۱</sup> دسته‌بندی کرد [۲، ۱۱، ۱۲].

- **حملات جعبه سفید:** در این دسته از حملات، قوی‌ترین دسترسی ممکن برای مهاجم فرض می‌شود. مهاجم در این نوع حمله از اطلاعات کامل راجع به مدل مورد استفاده برای تصمیم‌گیری (مثلاً معماری دقیق مدل، تابع هزینه مورد استفاده برای آموزش، گرادیان خروجی مدل نسبت به هر متغیر مطلوب مهاجم، وزن‌های مدل و غیره) برخوردار است. همچنین مهاجم از نحوه آموزش مدل (مثلاً الگوریتم بهینه‌سازی مورد استفاده) مطلع بوده و به مجموعه داده‌های آموزشی مدل دسترسی کامل دارد. با این مفروضات، مهاجم سعی می‌کند از نقاط ضعف مدل در فضای ویژگی<sup>۲</sup> مطلع شود و از آن‌های برای تخریب عملکرد مدل سوء استفاده کند.

- **حملات جعبه سیاه:** در این حملات که در نقطه مقابل حملات جعبه سفید قرار می‌گیرند، هیچ پیش‌فرض خاصی برای مهاجم در نظر گرفته نمی‌شود و نمایانگر یک سناریوی حمله محتمل‌تر است که در آن مهاجم سعی می‌کند با در اختیار داشتن اطلاعات محدود راجع به عملکرد مدل و خروجی‌های دریافتی از ورودی‌هایی که خودش به مدل ارسال می‌کند، از نقاط ضعف مدل پرده برداری کند.

در جدول ۱-۲ مقایسه‌ای اجمالی بین این دو دسته از حملات آمده است.

مدل تهدید علاوه بر نوع دسترسی مهاجم، به هدف غایی او نیز وابسته است. اهداف یک مهاجم از حمله به یک سیستم تصمیم‌گیرنده مبتنی بر یادگیری ماشین را می‌توان به موارد زیر خلاصه کرد:

۱. **تقلیل اطمینان<sup>۳</sup>:** در این حالت، مهاجم سعی می‌کند سطح اطمینان<sup>۴</sup> خروجی دسته‌بند را برای دسته تشخیص داده شده  $y_{pred}$  کاهش دهد در حالی که خروجی کلاس دسته‌بند دچار تغییر نشود. به عبارت

<sup>1</sup> White Box

<sup>2</sup> Feature Space

<sup>3</sup> Confidence Reduction

<sup>4</sup> Confidence

دیگر:

$$y_{pred} = \hat{y} = y, f(x)_y > f(\hat{x})_y$$

۲. دسته‌بندی اشتباه غیر هدفمند<sup>۱</sup>: در این حالت، مهاجم در صدد تغییر خروجی مدل به هر کلاس  $\hat{y} \neq y_{pred}$  بر می آید.

۳. دسته‌بندی اشتباه هدفمند<sup>۲</sup>: در این حالت، مهاجم تلاش می‌کند خروجی مدل را به کلاس خاص مطلوب  $y_{desired} \neq y_{pred}$  تغییر دهد.

هدف و تمرکز اصلی این پژوهش روی حملات تخاصمی گریزانه‌ی جعبه سفید است که از گرادیان خروجی یک مدل نسبت به ورودی آن مطلع هستند. در این حملات یک ورودی ساختگی در زمان تست به صورت مصنوعی توسط مهاجم با علم به اطلاعات خصوصی سیستم مورد حمله و با هدف ایجاد اختلال در خروجی آن،

جدول ۲- ۱: مقایسه حملات جعبه سیاه و جعبه سفید

حملات جعبه سیاه	حملات جعبه سفید	
تعریف	به دسته ای از حملات گفته می شود که در آن ها ساختار درونی و طراحی سیستم مورد حمله از مهاجم مخفیست.	به دسته ای از حملات گفته می شود که در آن ها مهاجم اطلاعات کامل از معماری و ساختار درونی سیستم مورد حمله دارد.
سطح اطلاعات قابل دسترسی	نیازمند اطلاعات بنیادی راجع به سیستم قربانی و نحوه عملکرد آن نیست.	نیازمند اطلاعات محرمانه راجع به سیستم مانند وزن های مدل مورد استفاده، معماری آن، کتابخانه های استفاده شده برای پیاده سازی و غیره می باشد.
مزایا و معایب	پیاده سازی معمولاً ساده تر، اما ضعیف تر. فرض واقع گرایانه تر راجع به مهاجمین احتمالی. کارآیی بالا در کشف ایرادات رفتاری مدل.	پیاده سازی پیچیده تر، اما به مراتب قوی تر حصول امنیت نسبی در برابر این حملات معمولاً وضعیت آرمانیست.
استراتژی حمله	در این حملات معمولاً یک مدل محلی به آموزش داده می شود که بتواند رفتار مدل قربانی را تقلید کند. این کار با استفاده از تولید نمونه های ورودی ساختگی توسط مهاجم و استفاده از برجسب های خروجی مدل قربانی روی همین ورودی ها، صورت می گیرد.	این حملات از خصوصیات درونی مدل مورد استفاده سوء استفاده می کنند. به عنوان مثال حملات مبتنی بر گرادیان و یا استفاده از دانش قبلی راجع به ضعف های موجود در سیستم با فرض دانش کامل راجع به ساختار داخلی مدل استفاده شده در این دسته از حملات قرار می گیرند.

<sup>1</sup>Untargeted Misclassification

<sup>2</sup>Targeted Misclassification

طراحی و به مدل داده می‌شود. در سال‌های اخیر حملات گریزانه متعددی با موفقیت روی شبکه‌های یادگیری عمیق اعمال شده‌اند. خواننده برای مرور کاملی بر روش‌های روز به [۴-۸، ۱۰] ارجاع داده می‌شود. به طور کلی، فرایند تولید یک نمونه تخصصی را می‌توان به صورت معادله ۲-۴ فرمول‌بندی کرد [۱۳]:

$$\begin{aligned} &\text{minimize } \mathcal{D}(x, x + \delta) \\ &\text{s.t. } C(x + \delta) = t \\ &x + \delta \in [0, 1]^n \end{aligned} \quad (۴-۲)$$

که در آن  $\mathcal{D}$  یک معیار فاصله مشخص است،  $C$  یک مدل دسته‌بند و  $t$  برچسبی غیر از برچسب اصلی متناظر با ورودی  $x$  است. به عبارت ساده‌تر، معادله ۲-۴ بیان می‌کند که به ازای یک ورودی  $x$  ثابت، هدف یافتن  $\delta$  مناسب است طوری که علاوه بر کمینه شدن مقدار  $\mathcal{D}(x, x + \delta)$ ، خروجی دسته‌بند  $C$  روی  $x + \delta$  تغییر کند. از میان حملات جعبه سفید موجود دو حمله معروف برای آزمون مدل دفاعی ارائه شده در ادامه این گزارش، استفاده شده‌اند. حمله **Fast Gradient Sign Method (FGSM)** حاصل یکی از اولین پژوهش‌ها در زمینه حملات تخصصی بود و امروزه به عنوان یک حمله‌ی سریع، ساده، اما نسبتاً ضعیف در میان حملات تخصصی جعبه سفید شناخته می‌شود [۹]. این حمله معمولاً در حالت غیر هدفمند پیاده‌سازی می‌شود ولی پیاده‌سازی آن در حالت هدفمند نیز ممکن است. در مقابل این حمله یکی دیگر از معروف‌ترین حملات جعبه سفید، حمله **Carlini-Wagner (CW)** است که در سال ۲۰۱۷ طراحی شد [۱۴] و هنوز یکی از قوی‌ترین حملات جعبه سفید شناخته با زمان اجرای معقول است که در هر دو نوع هدفمند و غیر هدفمند قابل پیاده‌سازی است. حمله **Projected Gradient Descent (PGD)** [۱۵] مستقیماً در این پژوهش برای آزمایش مدل‌ها مورد استفاده قرار نگرفته است ولی از آنجایی که در ادامه مباحث در بخش ۲-۳ به تعریف آن احتیاج است، این حمله نیز که یکی از حملات قوی و شناخته‌شده جعبه سفید در ادبیات حملات تخصصیست، مورد بررسی قرار خواهد گرفت.

#### ۲-۱-۲-۲ حمله FGSM

فرض کنید مدل  $f$  با پارامترهای  $\theta$  را در اختیار داشته باشیم. همچنین فرض کنید که  $(X, y)$  زوج‌های مرتبی از ورودی‌ها و خروجی‌های متناظر به  $f$  باشند و نیز تابع هزینه  $J$  که مدل به وسیله آن و طی یک فرایند بهینه‌سازی، آموزش داده شده است. اکنون عبارت ۲-۵ را در نظر بگیرید:

$$\nabla_x J(f(x; \theta), y) \quad (۵-۲)$$



این عبارت مقدار گرادیان تابع هزینه آموزش مدل  $f$  را نسبت به ورودی مدل (و نه پارامترهای مدل،  $\theta$ ) نشان می دهد. بدین ترتیب اگر به هر نحو این گرادیان به ورودی اولیه افزوده شود، ورودی تولید شده احتمالاً منجر به زیاد شدن مقدار تابع هزینه نهایی و موفقیت حمله خواهد شد. در [۹] نویسندگان از نویز تخصصی ارائه شده در ۶-۲ استفاده می کنند و نمونه تخصصی نهایی از معادله ۷-۲ بدست می آید.

$$\delta = \epsilon \cdot \text{sgn}(\nabla_x J(f(x; \theta), y)) \quad (۶-۲)$$

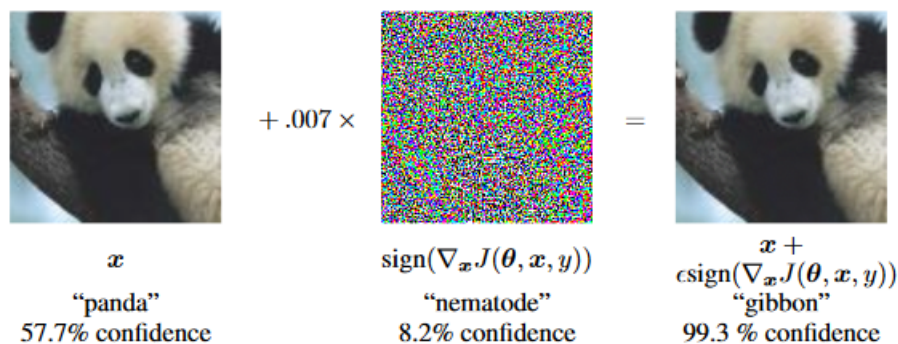
$$\hat{x} = x + \delta \quad (۷-۲)$$

که در آن مقدار  $\epsilon$  که کنترل کننده اندازه نویز تخصصی بوده و بسته به مجموعه داده مورد استفاده و ضریب اطمینان لازم برای موفقیت حمله قابل تنظیم است. نویسندگان در این پژوهش مقدار 0.007 را پیشنهاد کرده اند. به طور خلاصه در این حمله مقدار هر پیکسل از یک تصویر ورودی  $x$  به اندازه‌ی برابر بسته به جهت گرادیان تابع هزینه نسبت به  $x$  کم و یا زیاد می شود به طوری که تصویر حاصل نسبت به برجسب صحیح  $y$  هزینه بیشتر داشته باشد.

توجه شود که فرمول بندی ارائه شده در ۶-۲ و ۷-۲ خصوصاً برای دور کردن نتیجه دسته بند از برجسب حقیقی و پیاده سازی یک حمله غیرهدفمند است. اگر بخواهیم با استفاده از همین روش یک حمله هدفمند را با برجسب مطلوب  $y_{desired}$  اجرا کنیم، کافست که مقدار  $\delta$  به شکل معادله ۸-۲ تغییر داده شود:

$$\delta_{targeted} = -\epsilon \cdot \text{sgn}(\nabla_x J(f(x; \theta), y_{desired})) \quad (۸-۲)$$

شکل ۴-۲ یک نمونه از حمله FGSM آورده شده است. قربانی این حمله یک شبکه GoogLeNet است و ورودی یک تصویر ”پاندا”ست که مدل آموزش دیده شده می تواند با سطح اطمینان 57.7% برجسب این تصویر را به درستی تشخیص دهد. اکنون با افزودن 0.007 از نویز تخصصی تولید شده - که خود با سطح اطمینان 8.7%



شکل ۴-۲: نمونه حمله FGSM. در این حمله تصویر یک پاندا با افزودن مقدار ناچیزی نویز تخصصی به عنوان یک میمون دست دراز شناخته شده است [۹].

توسط مدل به عنوان “کرم لوله‌ای” دسته‌بندی شده. تصویر نهایی که همچنان به چشم غیرمسلح مانند تصویر اولیه است، با سطح اطمینان 99.3% توسط مدل به کلاس “میمون دست‌دراز” تعلق گرفته است. همانطور که مشخص است، این حمله در یک گام انجام می‌شود و یکبار محاسبه گرادیان تابع هزینه برای تولید حمله کفایت می‌کند و بنابراین حمله FGSM بسیار سریع قابل پیاده‌سازی است. با وجود این که امروزه حملات جعبه‌سفید به مراتب قوی‌تری برای سنجش مدل‌های دسته‌بند موجود است، FGSM همچنان به عنوان یک روش آسان، قابل فهم و سریع برای آزمایش‌های اولیه مورد استفاده قرار می‌گیرد.

### ۳-۱-۲-۲ حمله CW

حمله‌ی دیگر مورد استفاده در این پژوهش، حمله‌ی CW است که در [۱۴] ارائه شده است. برای توضیح نحوه‌ی ساختن یک نمونه تخاصمی با این روش به معادله ۲-۴ باز می‌گردیم. در این حمله، از آنجایی که شرط  $C(x + \delta) = t$  بسیار غیرخطی و غیرقابل بهینه‌سازیست، این شرط باید ابتدا به فرمی قابل بهینه‌سازی در آورده شود. برای این کار محققین این پژوهش تابع هدف  $f$  را طوری تعریف می‌کنند که شرط  $C(x + \delta) = t$  برقرار باشد اگر و تنها اگر  $f(x + \delta) \leq 0$ . اکنون به جای بیان مسئله به صورت

$$\begin{aligned} &\text{minimize } \mathcal{D}(x, x + \delta) \\ &\text{s.t. } f(x + \delta) \leq 0 \\ &x + \delta \in [0, 1]^n \end{aligned} \quad (۹-۲)$$

از فرمول‌بندی معادل ۲-۱۰ استفاده می‌شود:

$$\begin{aligned} &\text{minimize } \mathcal{D}(x, x + \delta) + c \cdot f(x + \delta) \\ &\text{s.t. } x + \delta \in [0, 1]^n \end{aligned} \quad (۱۰-۲)$$

که در آن  $c > 0$  مقداری ثابت است که قدرت حمله انجام شده را در ازای دادن کمینگی  $\mathcal{D}(x, x + \delta)$  تنظیم می‌کند. این دو فرمول‌بندی معادلند چرا که می‌توان نشان داد مقدار  $c > 0$  وجود دارد که راه‌حل بهینه در معادله ۲-۱۰ با راه‌حل بهینه ۲-۹ برابر است. در [۱۴] از نرم<sup>۱</sup>

$$L_p(\vec{v}) = \|\vec{v}\|_p = \left( \sum_{i=1}^n |v_i|^p \right)^{\frac{1}{p}}$$

به عنوان معیار فاصله  $\mathcal{D}$  استفاده می‌شود. و بدین ترتیب معادله کلی نهایی حمله CW به صورت زیر در

<sup>۱</sup>Norm

خواهد آمد:

$$\begin{aligned} & \text{minimize } \|\delta\|_p + c \cdot f(x + \delta) \\ & \text{s.t. } x + \delta \in [0, 1]^n \end{aligned} \quad (۱۱-۲)$$

اکنون برای حل این مسئله با استفاده از روش‌های بهینه‌سازی تکراری، نیاز به انتخاب  $p$ ،  $c$  و تابع هزینه مناسب  $f$  است.

این حمله با نرم‌های  $L_0$ ،  $L_2$  و  $L_\infty$  قابل پیاده‌سازی است. از آنجایی که احتمال موفقیت حمله و نیز اندازه نرم نمونه تخصصی بر حسب مقدار ثابت  $c$  استفاده شده، توابعی اکیداً نزولی هستند (طبق شکل ۲-۵)، می‌توان مقدار  $c$  بهینه را با استفاده از روش جستجوی دودویی<sup>۱</sup> بدست آورد.

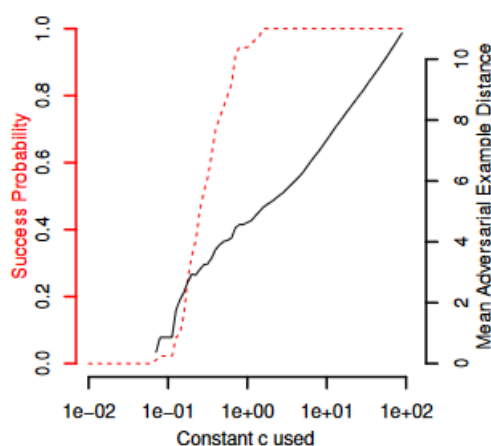
شروطی مانند  $x + \delta \in [0, 1]^n$  به نام محدودیت‌های جعبه‌ای<sup>۲</sup> معروف هستند. برای اجرا کردن این شرط، در حمله CW از یک تغییر متغیر استفاده می‌شود. به جای بهینه‌سازی مستقیم مقدار  $\delta$  در معادله اصلی، متغیر جدید  $w$  معرفی می‌شود و مقدار

$$\delta = \frac{1}{2}(\tanh(w) + 1) - x$$

بهینه می‌شود. از آنجایی که  $-1 \leq \tanh(w) \leq 1$  محدودیت جعبه‌ای ذکر شده به طور خودکار اعمال خواهد شد چرا که  $0 \leq x + \delta \leq 1$  خواهد بود.

قوی‌ترین نوع حمله CW،  $L_2$ -CW است که مسئله بهینه‌سازی آورده شده در ۲-۱۲ را حل می‌کند.

$$\text{minimize } \left\| \frac{1}{2}(\tanh(w) + 1) - x \right\|_2^2 + c \cdot f\left(\frac{1}{2}(\tanh(w) + 1)\right) \quad (۱۲-۲)$$



شکل ۲-۵: نمودار حساسیت موفقیت حمله و اندازه نویز تخصصی بر حسب مقدار ثابت  $c$  [۱۴]

<sup>۱</sup>Binary Search

<sup>۲</sup>Box Constraints

که تابع هزینه  $f$  در آن به صورت زیر تعریف می شود:

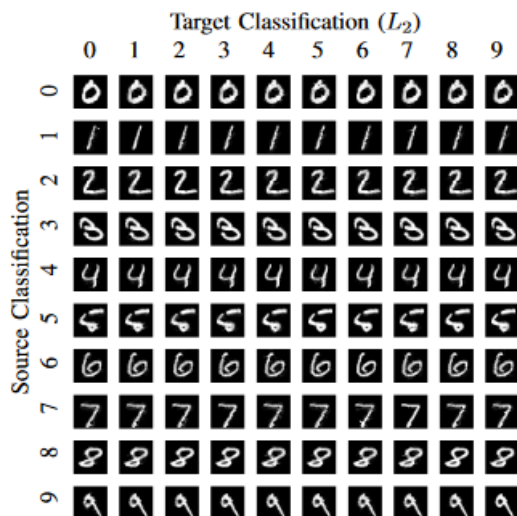
$$f(x) = \max(\max\{Z(x)_i : i \neq t\} - Z(x)_t, -\kappa)$$

و  $Z(\cdot)$  خروجی های logit مدل دسته‌بند قربانی  $F$  هستند (به طوری که  $F(x) = \text{softmax}(Z(x)) = y$ ) و پارامتر  $\kappa > 0$  میزان سطح اطمینان حمله را مشخص می کند به طوری که مقادیر بزرگتر  $\kappa$  الگوریتم بهینه‌ساز را مجبور به یافتن پاسخ‌هایی می کند که با احتمال قوی‌تری می‌توانند خروجی دسته‌بند را تغییر دهند. به طور معمول از مقدار 14 برای این پارامتر برای تولید حملات قوی استفاده می شود. مسئله بهینه‌سازی ۲-۱۲ توسط بهینه‌ساز Adam [۱۶] حل می‌شود و در نهایت نمونه‌های تخصصی به دست خواهند آمد. نمونه‌هایی از این حمله را می توان در شکل ۲-۶ مشاهده کرد. همانطور که می توان دید، بدون توجه به این که کلاس هدف دارای چه مقداریست تمامی نمونه‌های تخصصی متناظر با تصویر سالم به چشم غیرمسلح کاملاً شبیه نمونه اصلی هستند.

#### ۴-۱-۲-۲ حمله PGD

در نهایت در این بخش به توضیح حمله PGD [۱۵] خواهیم پرداخت. در این حمله دیدگاه جدیدی برای تولید حملات در نظر گرفته شده است. بر خلاف تعاریف پیشین، در این پژوهش محققین عملیات حمله و دفاع را به عنوان یک بازی تخصصی بین مهاجم و مدل قربانی بررسی می کنند که تحت عنوان مسئله نقطه‌زینی زیر به عنوان تابع هزینه‌ی مدل قربانی بیان می شود:

$$\min_{\theta} \max_{\delta \in \mathcal{S}} \mathbb{E}_{(x,y) \sim \mathcal{D}} L(\theta, x + \delta, y) \quad (۲-۱۳)$$



شکل ۲-۶: نمونه‌های تخصصی تولید شده توسط  $L_2$ -CW روی مجموعه داده MNIST [۱۴].

که در آن وظیفه مهاجم حل مسئله بیشینه‌سازی درونی و وظیفه قربانی حل مسئله کمینه‌سازی برونیت. اکنون اگر فرمول ۲-۶ تشکیل حمله تخصصی در حمله FGSM بازگردیم، می‌توان گفت که این حمله در واقع می‌تواند به عنوان یک گام از حل مسئله بیشینه‌سازی مهاجم در فرمول‌بندی ۲-۱۳ در محدوده‌ی  $L_\infty$  حول یک نمونه‌ی سالم عمل کند. اکنون اگر بخواهیم این مسئله را با استفاده از نزول گرادیان<sup>۱</sup> حل کنیم، کافیت همین گام معرفی شده را چندین بار تکرار کنیم و حاصل را روی  $\epsilon$ -کری حول نمونه‌ی سالم بیافکنیم. این روش بهینه‌سازی با محدودیت که مبنای حمله‌ی PGD است، نزول گرادیان افکنده<sup>۲</sup> نام دارد. به طور دقیق‌تر اگر بخواهیم همچنان مانند حمله FGSM از نرم  $L_\infty$  بهره بگیریم، تشکیل یک حمله تخصصی در PGD از تکرار گام زیر به‌دست خواهد آمد:

$$x^{t+1} = \Pi_{x+\mathcal{S}}(x^t + \alpha \cdot \text{sgn}(\nabla_x L(\theta, x, y))) \quad (۱۴-۲)$$

که در آن  $\Pi_{x+\mathcal{S}}$  نشان دهنده عملیات افکنش روی  $\epsilon$ -کری  $x + \mathcal{S}$  بوده و این بار  $\alpha$  نمایانگر نرخ یادگیری<sup>۳</sup> نزول گرادیان افکنده است. نکته قابل توجه در این رابطه آن است که عملیات افکنش حول  $\epsilon$ -کری نمونه سالم اولیه  $x$  انجام می‌شود (و نه روی  $x^t$ ). بدین ترتیب پس از چندین گام به حمله‌ای خواهیم رسید که طبق تعریف، شرایط یک حمله تخصصی را - در صورت امکان- برآورده خواهد کرد.

در جدول ۲-۲ مقایسه‌ای اجمالی میان حملات بررسی شده، آمده است.

جدول ۲-۲: مقایسه حملات FGSM، PGD و CW

CW	PGD	FGSM	
چند گام	چند گام	یک گام	تعداد گام‌های لازم برای پیاده‌سازی حمله
زیاد	متوسط	نسبتاً کم ولی همچنان موثر	قوت حمله
کند تر، زیرا علاوه بر چندین گام نزول گرادیان، نیاز به انجام جستجوی دودویی برای بهینه‌سازی مقدار یکی از پارامترها است.	کند، چون به چند گام از نزول گرادیان افکنده احتیاج است	بسیار سریع، چراکه حمله در یک گام صورت می‌پذیرد	زمان اجرای حمله
پیچیده تر، علاوه بر استفاده از تابع هزینه مخصوص، به پیاده‌سازی جستجوی دودویی نیز احتیاج است	نسبتاً پیچیده، نیاز به اعمال چندین گام نزول گرادیان افکنده است که کمی دشوارتر از نزول گرادیان معمولیست	ساده، فقط نیاز به یکبار محاسبه گرادیان تابع هزینه مدل نسبت به ورودی اعمال شده به مدل است	پیچیدگی پیاده‌سازی حمله
جعبه سفید	جعبه سفید	جعبه سفید	نوع دسترسی مهاجم

<sup>۱</sup>Gradient Descent

<sup>۲</sup>Projected Gradient Descent

<sup>۳</sup>Learning Rate

## ۳-۲ روش های دفاع در برابر حملات تخاصمی

دفاع در برابر حملات تخاصمی از دو جنبه کلی مورد بررسی قرار می گیرد [۱۰]:

۱. پاکسازی<sup>۱</sup>: در این حالت، هدف پاکسازی ورودی های دسته بند و یا مقاوم سازی دسته بند از لحاظ ساختاری در راستای اصلاح خروجی آن در هنگام بروز حمله است به طوری که به خروجی در زمان های دیگر آسیبی وارد نشود.

۲. تشخیص<sup>۲</sup>: در این حالت - که تمرکز اصلی این پژوهش است- هدف صرفاً تشخیص حمله ی تخاصمی پیش از ورود آن به مدل و اتخاذ تصمیم بر اساس خروجی اشتباه احتمالی دسته بند است. این امر عموماً با یک دسته بند مجزا برای تشخیص حملات صورت می گیرد.

ابتدا مختصری راجع به روش های پاکسازی و سپس در مورد روش های تشخیص بحث خواهد شد.

### ۱-۳-۲ روش های پاکسازی حمله

در ادبیات مرتبط با دفاع در برابر حملات تخاصمی، عبارت پاکسازی به دسته خاصی از روش های ممکن برای دفاع اطلاق می شود. در این پژوهش، برای سهولت دسته بندی، کمی از تعریف رسمی پاکسازی تخاصمی<sup>۳</sup> دور شده ایم و آن را مطابق تعریف ارائه شده در بخش ۲-۳ در نظر می گیریم. بنابراین پاکسازی تخاصمی خود در این دسته قرار خواهد گرفت. با این مقدمه، روش های پاکسازی را می توان در ابعاد زیر دسته بندی نمود [۵، ۶، ۱۰]:

۱. آموزش تخاصمی<sup>۴</sup>: یکی از ابتدایی ترین روش های دفاع که می تواند قابل اعمال روی هر شبکه عصبی مورد حمله باشد. در [۱۳] پیشنهاد شده است که انجام فرایند آموزش روی ترکیبی از نمونه های سالم و نمونه های تخاصمی می تواند باعث منظم سازی<sup>۵</sup> شبکه عصبی آموزش دیده و پرورش توانایی برای مقابله در برابر حملات تخاصمی باشد. در [۹] این نظریه با اعمال حمله FGSM روی مجموعه داده های آموزشی و افزودن این نمونه ها به مجموعه داده و سپس آموزش نهایی مدل روی این مجموعه داده جدید، مورد آزمون قرار گرفته و تاثیر آن به طور تجربی ثابت شده است.

همانطور که پیش تر توضیح داده شد، در [۱۵] روشی نوین برای آموزش تخاصمی ارائه شده است که در

<sup>1</sup>Purification

<sup>2</sup>Detection

<sup>3</sup>Adversarial Purification

<sup>4</sup>Adversarial Training

<sup>5</sup>Regularization

آن تابع ضرر<sup>۱</sup> به فرم یک مسئله بهینه‌سازی کمینه-بیشینه<sup>۲</sup> بیان شده است (رابطه ۲-۱۳) در این رابطه، مسئله بیشینه‌سازی درونی سعی می‌کند که شبیه فرمول‌بندی ارائه شده در بخش ۲-۱-۲-۲ برای حمله FGSM، قوی‌ترین مقدار  $\delta$ ی ممکن برای بزرگ کردن مقدار تابع هزینه پیدا شود. این در حالیکه در مسئله کمینه‌سازی برونی هدف کمینه کردن مقدار این تابع ضرر تخصصی با تنظیم کردن پارامترهای مدل ( $\theta$ ) است. بدین ترتیب، اگر مقدار تلورانس<sup>۳</sup> حمله را  $\epsilon$  در نظر بگیریم، مدل با نمونه‌های تخصصی در یک  $\epsilon$ -کره حول هر نمونه‌ی سالم آموزش داده خواهد شد.

مهم‌ترین ایراد آموزش تخصصی این است که تولید نمونه‌های تخصصی قوی در زمان آموزش - خصوصاً روی مجموعه داده‌های بزرگ مانند ImageNet - می‌تواند بسیار زمان بر باشد و بدین ترتیب اکثر روش‌های آموزش تخصصی از حملات تک-مرحله‌ای مانند FGSM برای افزایش مجموعه‌داده‌های آموزشی استفاده می‌کنند. برای حل این مسئله روش Free Adversarial Training [۱۷] ارائه شد که از اطلاعات گرادیان مدل در زمان آموزش برای تولید حملات تخصصی (PGD) استفاده می‌کند و بنابراین نیازمند تولید مجدد حمله نیست. همچنین در این روش، برای کاهش زمان همگرایی حمله PGD نیز تخصصی بدست آمده برای یک دسته از ورودی‌های سالم، به عنوان نقطه آغازین حمله PGD روی دسته ورودی‌های بعد مورد استفاده قرار می‌گیرد. این دفاع روی حمله چند - مرحله‌ای PGD [۱۵] مورد آزمایش قرار گرفته و نتایج امیدوار کننده‌ای داشته است. همچنین در You Only Propagate Once (YOPO) [۱۸] این مسئله مورد بررسی قرار گرفته که نیاز به یک لایه دفاعی در برابر حملات تخصصی را می‌توان تقریباً فقط به اولین لایه‌ی یک شبکه عمیق خلاصه کرد. بنابراین آموزش تخصصی می‌تواند بسیار ارزان‌تر صورت بگیرد. این روش که مستقیماً با Free Adversarial Training مورد مقایسه قرار گرفته است، نشان می‌دهد که می‌تواند در زمانی کمتر دارای عملکرد مشابه باشد. از دیگر ایده‌های جالب توجه مطرح شده در این زمینه می‌توان به Geometry-aware Instance-reweighted Adversarial Training (GI-AT) [۱۹] و Fast Adversarial Training [۲۰] اشاره کرد. در GI-AT از این نکته بهره گرفته می‌شود که نمونه داده‌هایی که نزدیک مرزهای تصمیم‌گیری یک مدل قرار دارند، بیشتر می‌توانند در موفقیت یک حمله تخصصی تاثیرگذار باشند. بدین ترتیب این روش با بهره‌گیری از راهکارهای استاندارد آموزش تخصصی، هر یک از نمونه داده‌های آموزشی را بر مبنای این که تولید حمله‌ی تخصصی موفق از روی آن‌ها چقدر دشوار است، وزن‌دهی کرده و در فرایند پس‌انتشار<sup>۴</sup> دخیل می‌کند. نهایتاً در Fast Adversarial Training این

<sup>1</sup>Loss

<sup>2</sup>Min-max

<sup>3</sup>Tolerance

<sup>4</sup>Backpropagation

نکته به صورت تجربی نشان داده می‌شود که آموزش تخصصی با استفاده از حمله تک-مرحله‌ای FGSM ولی با آغاز تصادفی<sup>۱</sup> (بر خلاف [۱۷]) می‌تواند به اندازه آموزش تخصصی با استفاده از حملات قوی‌تر چند-مرحله‌ای (مانند PGD) اثر بخش باشد. در این پژوهش محققین موفق شده‌اند که در کسری از زمان گزارش شده در [۱۷] به نتیجه مشابه دست بیابند.

۲. **تغییرات در مدل قربانی و فرایند آموزش:** پژوهش [۲۱] یکی از اولین کارهای انجام شده در این زمینه که با فاصله بسیار اندکی از کشف حملات تخصصی و آسیب‌پذیری شبکه‌های عصبی به این حملات، صورت گرفت. با وجود این که حملات تخصصی ایجاد شده پس از این مقاله، می‌توانند به راحتی راهکارهای ارائه شده در این کار را دور بزنند، ایده‌های مطرح شده همچنان شایان ذکر به نظر می‌رسند. ایده‌ی اصلی این پژوهش ارائه روش‌های آموزش جدید به گونه‌ایست که نمونه‌های تخصصی تولید شده توسط حمله L-BFGS [۱۳] دارای اعوجاج<sup>۲</sup> بیشتری نسبت به نمونه‌های متناظر سالم دارند و دیگر قابل صرف نظر نیستند. در این پژوهش سه روش پیش پردازش برای مقابله با حمله L-BFGS مورد بررسی قرار گرفته‌اند: (آ) **ترریق نویز:** افزودن نویز گاوسی اندک به ورودی‌ها می‌تواند به تشخیص درست تعداد بیشتری از نمونه‌های تخصصی در ازای مقدار اندکی هدر رفت دقت دسته‌بندی بیانجامد.

(ب) **خودرمزگذار<sup>۳</sup>:** در این روش یک خودرمزگذار با هدف بازسازی نمونه‌های سالم از روی نمونه‌های تخصصی، آموزش داده شده است که می‌تواند از حمله جلوگیری کند.

(ج) **خودرمزگذار نویزگیر:** در این روش، شبیه روش قبلی، یک خودرمزگذار نویزگیر معمولی بدون دانش پیشین از توزیع آماری نویز تخصصی و صرفاً با هدف از بین بردن نویز آموزش داده شده است. سپس در زمان آموزش، هر پیکسل از نمونه‌ی آموزشی با یک نویز گاوسی با میانگین ۰ و انحراف معیار متغیر  $\sigma$  ترکیب می‌شود. نشان داده شده است که با قرار دادن  $\sigma = 0.1$  این خودرمزگذار می‌تواند به خوبی خودرمزگذار قبلی عمل کند.

ایراد اصلی وارد به موارد ۲ب و ۲ج این است که با سری کردن خودرمزگذار و مدل قربانی و حمله به مدل سری شده، همچنان می‌توان حمله موفق داشت. این در حالیست که روش ۲آ نمی‌توانست در برابر حملات جعبه‌سفید قوی‌تر که بعدها ارائه شدند (مانند PGD و CW) مقاومت کند.

ایده‌ی مهم دیگری که کمی بعدتر ارائه شد، ایده‌ی تقطیر دفاعی<sup>۴</sup> [۲۲] است. همانطور که در شکل ۲-۷

<sup>1</sup>Random Initialization

<sup>2</sup> $\text{Distortion}(x, \hat{x}) = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2$  where  $x, \hat{x} \in \mathbb{R}^n$

<sup>3</sup>Autoencoder

<sup>4</sup>Defensive Distillation



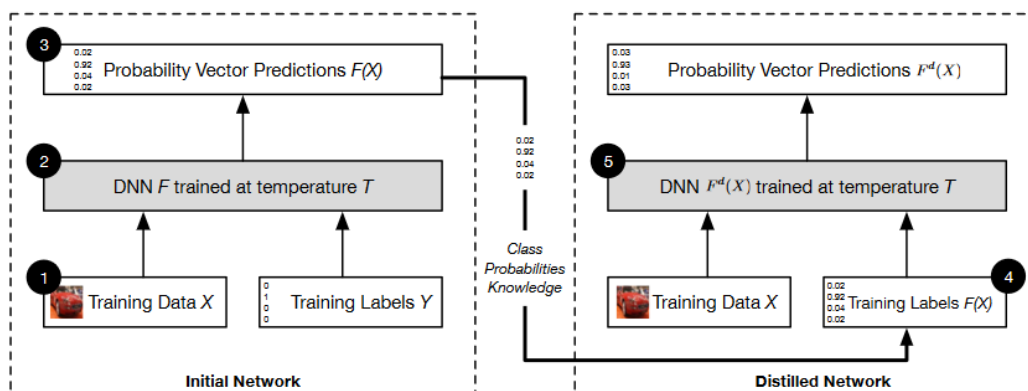
نشان داده شده است، ایده‌ی اصلی در این روش استفاده از پیش‌بینی‌های یک مدل از پیش آموزش داده شده روی مجموعه داده‌های سالم، به عنوان برجسب‌های جدید مدل تقطیر شده است. بدین ترتیب، مدل تقطیر شده برخلاف مدل اصلی دارای برجسب‌هایی با مقادیر پیوسته هستند که این امر آن‌ها را نسبت به حملات تخاصمی مقاوم‌تر می‌سازد. این روش در برابر حمله CW ناتوان است ولی با توجه به موفقیت آن در برابر گستره نسبتاً وسیعی از حملات دیگر، در سال‌های آینده‌های قوی‌تری بر مبنای همین پژوهش ارائه شدند که مطالعه آن‌ها به خواننده واگذار می‌شود [۲۳-۲۵].

در [۲۶] روشی تحت عنوان “منظم‌سازی گرادیان” ارائه شده است. ایده‌ی کلی در این روش اعمال یک جمله منظم‌سازی به تابع هزینه آموزش مدل است به طوری که هدف آموزش را می‌توان به صورت زیر بازنویسی کرد:

$$\arg \min_{\theta} H(y, \hat{y}) + \lambda \|\nabla_x H(y, \hat{y})\|_2^2$$

که در آن  $y$  و  $\hat{y}$  به ترتیب برجسب حقیقی و پیش‌بینی مدل و  $\lambda$  پارامتر تنظیم کننده میزان جریمه منظم‌سازی می‌باشد. هدف این روش (که با افزودن جمله منظم‌ساز شامل  $\nabla_x H(y, \hat{y})$  محقق می‌شود) این است که از موضوع اطمینان حاصل شود که در صورت ایجاد تغییرات اندک در یک نمونه ورودی، دیورژانس KL بین پیش‌بینی مدل و برجسب واقعی تغییر چندانی نخواهد کرد و بنابراین نمونه‌های تخاصمی که در یک  $\epsilon$ -کره محدود می‌شوند، نخواهند توانست خروجی مدل قربانی را تغییر دهند.

در روش DeepCloak [۲۷] که در شکل ۲-۸ نمایش داده شده، ایده مطرح شده آن است که با قرار دادن یک لایه ماسک‌کننده دقیقاً قبل از لایه خطی که تولید کننده logit های مدل، ویژگی‌های بی‌اهمیت در خروجی نهایی مدل را از صفر کرده و آن را نسبت به نویز تخاصمی مقاوم‌تر ساخت. آموزش این لایه با دادن ورودی‌های سالم و تخاصمی به مدل و encode کردن اختلاف بین ویژگی‌های آن‌ها در لایه پیشین،



شکل ۲-۷: چارچوب ارائه شده در روش تقطیر دفاعی [۲۲]

صورت می گیرد. قوت این روش در برابر حمله FGSM به صورت تجربی نشان داده شده است. در ایده‌ای مشابه پژوهش قبلی و ترکیب آن با [۲۱]، در Random Self-Ensemble [۲۸] یک لایه نویز برای افزودن مقدار ناچیزی نویز ایزوتروپیک گاوسی به ورودی‌های مدل و نیز خروجی‌های لایه‌های پنهان درون مدل قرار داده می شود:

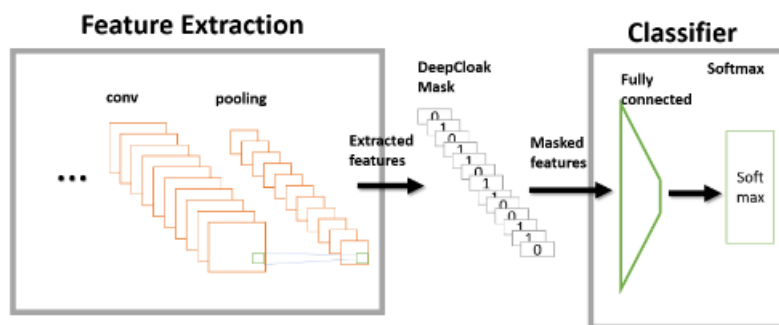
$$\text{NoiseLayer}(x) \rightarrow x + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I).$$

مقادیر بزرگتر  $\sigma$  در این لایه مقاومت در برابر حملات تخصصی را در ازای افت دقت دسته‌بندی، افزایش می‌دهند. برای کاهش تاثیر منفی  $\sigma$  از این لایه هم در زمان آموزش (با استفاده از تکنیک reparameterization) و هم در زمان تست، استفاده می‌شود. سپس، با تغییر مقدار  $\sigma$  - و به تبع آن، مقدار نویز افزوده شده به خروجی لایه‌ها،  $\epsilon$  - می‌توان بدون overhead اضافی، عملاً به ensemble ای از مدل‌ها دست پیدا کرد که بر اساس آن‌ها می‌توان تصمیم‌گیری‌های دقیق‌تری نسبت به نمونه‌های تخصصی انجام داد.

۳. استفاده از شبکه‌های جانبی: در [۲۹] دو علت اصلی برای اشتباه دسته‌بندها در مواجهه با یک نمونه تخصصی برشمرده شده است:

(آ) نمونه تخصصی از مرزهای خمینه<sup>۱</sup> وظیفه مورد نظر دور است. به عنوان مثال اگر وظیفه تشخیص اعداد دست نویس و مجموعه داده MNIST را در نظر بگیریم، یک نمونه تخصصی ممکن است تصویری باشد که اصلاً شامل یک عدد دست‌نویس نیست ولی دسته‌بند از آنجایی که مجبور به تولید خروجی است، دچار اشتباه خواهد شد.

(ب) نمونه تخصصی به مرزهای خمینه مورد نظر خیلی نزدیک است. در این حالت که اکثر



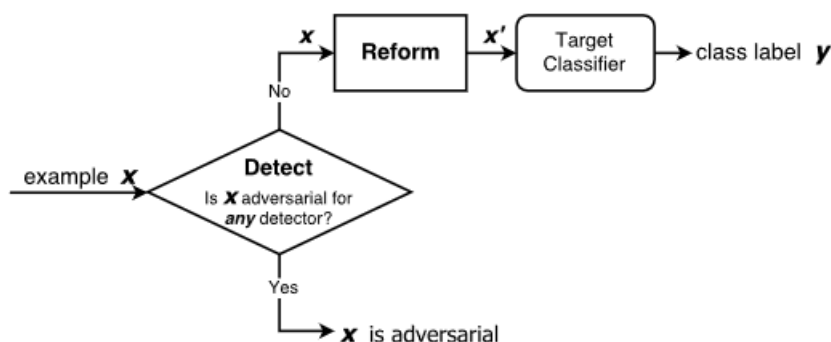
شکل ۲-۸: نحوه عملکرد DeepCloak [۲۷]

<sup>۱</sup>Manifold

حملات تخاصمی جدید از آن استفاده می کنند، اگر دارای یک دسته‌بند باشیم، دسته‌بندی که در فضای اطراف یک نمونه تخاصمی قدرت تعمیم‌پذیری کمی دارد، در مواجهه با این نمونه خروجی اشتباه تولید خواهد کرد.

با توجه به دلایل ارائه شده، در این پژوهش چارچوبی به نام MagNet ارائه می شود. برای دفاع در برابر علت اول مطرح شده، MagNet از چندین شبکه تشخیص‌دهنده استفاده می‌کند برای آن که فاصله یک نمونه‌ی زمان تست را با نمونه‌های آموزشی بسنجد. به عبارت دقیق‌تر، یک تشخیص‌دهنده تابع  $f: \mathcal{X} \rightarrow (0, 1)$  را یاد می‌گیرد که در آن  $\mathcal{X}$  مجموعه تمام نمونه‌های زمان تست است و خروجی این تشخیص‌دهنده معیاری از فاصله نمونه زمان تست با خمینه نمونه‌های سالم زمان آموزش است. و سپس برای برطرف کردن مشکل دوم، این چارچوب از یک بهسازی<sup>۱</sup> برای تصحیح نمونه‌های دور از خمینه نمونه‌های سالم استفاده می‌شود. این شبکه بهسازی توسط یک خود-رمزگذار پیاده‌سازی می‌شود. در صورت تشخیص نمونه تخاصمی توسط حداقل یکی از تشخیص‌دهنده‌ها، این نمونه به شبکه بهسازی داده می‌شود و خروجی شبکه بهسازی در نهایت به دسته‌بند می‌رسد (شکل ۲-۹).

یکی از مهم‌ترین ایده‌های مطرح شده در زمینه استفاده از شبکه‌های جانبی چارچوب Defense-GAN [۳۰] است. در این پژوهش برای پاکسازی یک نمونه تخاصمی از یک شبکه مولد تخاصمی<sup>۲</sup> بهره برده می‌شود. در این چارچوب که در شکل ۲-۱۰ نمایش داده شده است، ابتدا یک WGAN روی مجموعه‌ای از داده‌های سالم آموزش داده می‌شود<sup>۳</sup>. سپس در زمان تست، پیش از ورود یک نمونه به دسته‌بند، با استفاده از شبکه مولد GAN آموزش دیده شده، این نمونه به خمینه‌ی توزیع احتمالی یادگیری شده توسط



شکل ۲-۹: نحوه‌ی عملکرد MagNet [۲۹]

<sup>۱</sup>Reformer

<sup>۲</sup>Generative Adversarial Network (GAN)

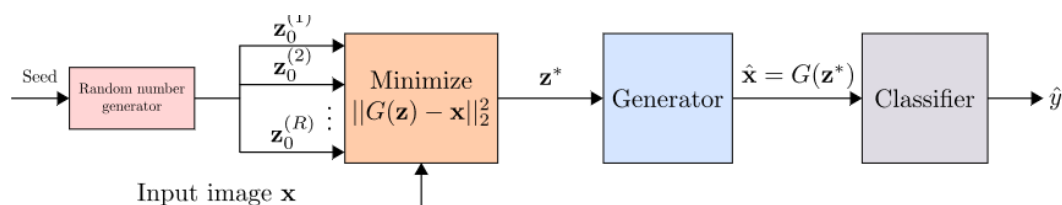
<sup>۳</sup>برای توضیحات مقدماتی راجع به GAN ها، می‌توانید به بخش ۲-۴-۱ مراجعه کنید

مولد، افکنده می‌شود. به عبارت دقیق‌تر، ابتدا  $R$  نمونه بردار نهفته<sup>۱</sup> به صورت تصادفی تولید می‌شوند. سپس با استفاده از نزول گرادیان، تمام این  $R$  بردار پنهان در کمینه کردن تابع هزینه

$$\|G(z) - x\|_2^2$$

شرکت داده می‌شوند. بدین ترتیب، بردار برتر  $z^*$  که  $G(z^*)$  نزدیک‌ترین نمونه‌ی ساختگی ممکن توسط مولد به نمونه‌ی احتمالاً تخصی‌ورودی یافته خواهد شد و در نهایت  $\hat{x} = G(z^*)$  به جای نمونه‌ی اصلی ( $x$ ) به دسته‌بند داده خواهد شد. هدف از این افکنش آن است که با توجه به این که GAN در زمان آموزش روی نمونه‌های سالم آموزش داده شده است، یافتن  $z^*$  به ترتیب توضیح داده شده، به از بین بردن هرگونه نویز تخصی‌کمک خواهد کرد. یکی از نقاط قوت Defense-GAN ماهیت غیرخطی آن به دلیل وجود یک حلقه بهینه‌سازی نزول گرادیان در پروسه‌ی پیاده‌سازی مکانیزم دفاع می‌باشد. این امر، Defense-GAN را نسبت به حملات جعبه‌سفید مقاوم می‌سازد. همچنین، این چارچوب هیچ پیش‌فرضی راجع به نوع حمله تخصی‌مورد استفاده ندارد و از آنجایی که صرفاً با تخمین توزیع احتمالی نمونه‌های سالم کار می‌کند، می‌تواند برای دفاع در برابر هر حمله‌ای مورد استفاده قرار بگیرد.

مشابه این پژوهش، در APE-GAN<sup>۲</sup> [۳۱] مجدداً از یک شبکه مولد تخصی‌برای پاکسازی نویز تخصی‌استفاده شده است. این چارچوب که در شکل ۲-۱۱ نمایش داده شده است، از یک خودرمزگذار به عنوان مولد و یک شبکه ممیز تشکیل شده است. هدف نهایی این چارچوب آن است که مولد  $G$  طوری آموزش ببیند که بتوان نویزهای ناچیز تخصی‌را از روی ورودی تخصی‌احتمالی، حذف کند، بدون آن که خروجی مدل در زمان دریافت ورودی‌های سالم دچار تغییر محسوسی بشود. برای تحقق این هدف از یک ساختار تخصی‌و برقراری یک بازی خصمانه بین این  $G$  و یک مدل ممیز  $D$  استفاده می‌شود. وظیفه  $D$  در این وضعیت تشخیص دادن نمونه‌های پاکسازی شده توسط  $G$  از نمونه‌های سالم متناظر است. در نهایت این مولد قادر به تولید نمونه‌های پاکسازی شده‌ای خواهد بود که توسط ممیز غیر قابل تشخیص



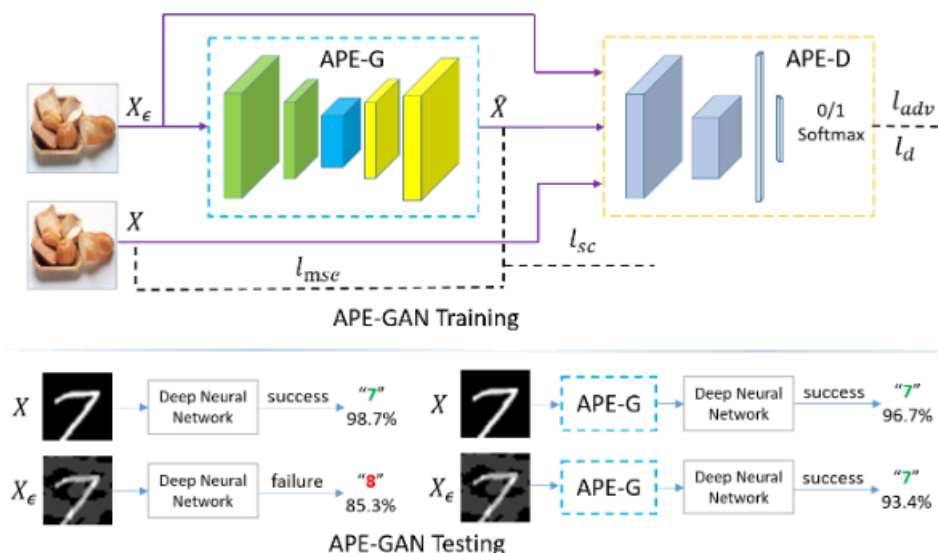
شکل ۲-۱۰: دورنمای عملکرد چارچوب Defense-GAN [۳۰]

<sup>۱</sup>Latent

<sup>۲</sup>Adversarial Perturbation Elimination Generative Adversarial Network

هستند و در نتیجه تغییرات ایجاد شده در آن‌ها نسبت به ورودی اصلی، ناچیز است. برای حصول اطمینان از ناچیز بودن تغییرات اعمال شده توسط  $G$  چندین جمله به تابع هزینه  $G$  افزوده می‌شود که پایداری فرایند آموزش مولد را بهبود ببخشد. در نهایت در زمان تست، تمامی ورودی‌ها از مولد  $G$  عبور کرده و سپس به مدل قربانی تحویل داده می‌شوند.

در [۳۲] استفاده از یک دسته‌بند با Embedding منظم شده (Embedding Regularized Classifier) پیشنهاد شده است. ایده‌ی اصلی این پژوهش آن است که نمونه‌های تخصصی و نمونه‌های سالم از توزیع‌های احتمالی متفاوتی تولید می‌شوند، بدین ترتیب، اگر بتوانیم به نحوی یک قسمتی از دسته بند را که موظف به استخراج ویژگی از ورودی پیش از انجام دسته‌بندی است، به سمت تولید ویژگی‌هایی از توزیع سالم ترغیب کنیم، دسته‌بند عملکرد بهتری در برابر نمونه‌های تخصصی خواهد داشت. همان‌طور که در شکل ۲-۱۲ نشان داده شده است، هر دسته‌بند را می‌توان به صورت دو زیرشبکه متصور شد: یک Encoder که وظیفه آن استخراج ویژگی از ورودی دسته‌بند است و یک هِد دسته‌بند برای تولید خروجی کلاس نهایی. اکنون در این چارچوب توزیع بردارهای نهفته تولید شده توسط شبکه Encoder با یک توزیع پیشین<sup>۱</sup> مقایسه می‌شوند و توسط یک شبکه جانبی ممیز<sup>۲</sup> این توزیع پیشین مطلوب و توزیع تولید شده توسط Encoder مورد مقایسه قرار می‌گیرند. بدین ترتیب شبکه Encoder به سمت ایجاد بردارهای نهفته از توزیع مطلوب، منظم می‌شود و احتمال موفقیت حملات تخصصی کاهش خواهد یافت.

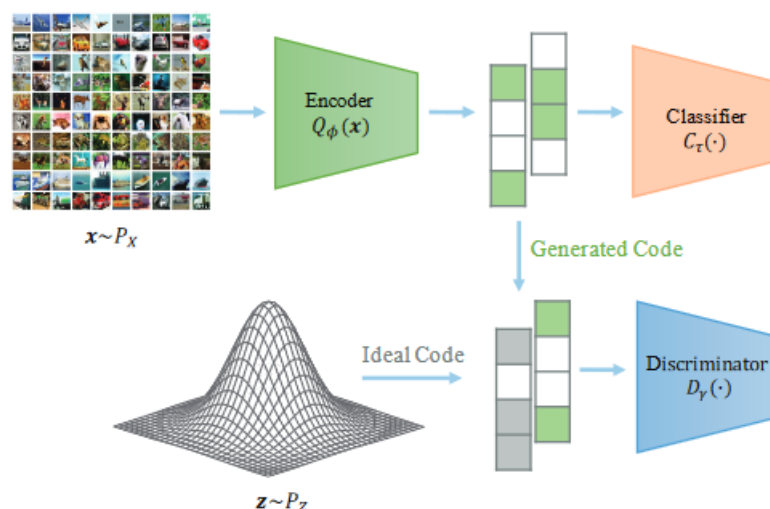


شکل ۲-۱۱: دورنمای APE-GAN در زمان آموزش و تست [۳۱]

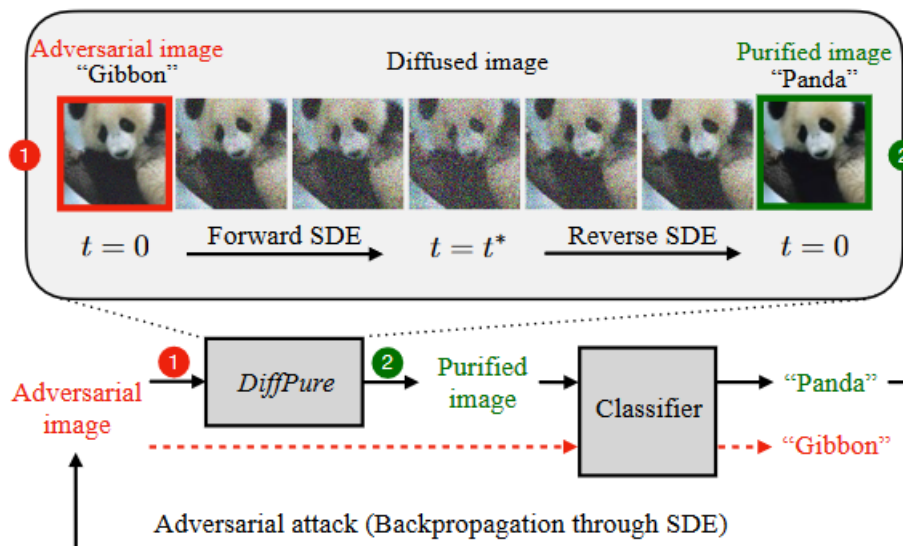
<sup>1</sup>Prior

<sup>2</sup>Discriminator

یکی از جدیدترین ایده های مطرح شده در زمینه پاکسازی نمونه های تخصصی DiffPure [۳۳] است. در این چارچوب از یک مدل انتشاری<sup>۱</sup> برای پاکسازی نویز تخصصی تزریق شده به یک تصویر استفاده می شود<sup>۲</sup>. نحوه عملکرد این چارچوب در شکل ۲-۱۳ نمایش داده شده است. مدل های انتشاری از دو فرایند انتشار پیش رو<sup>۳</sup> و انتشار معکوس<sup>۴</sup> تشکیل می شوند. ایده ی اصلی DiffPure آن است که با اعمال انتشار پیش رو روی ورودی تخصصی احتمالی و به تبع آن افزودن مقداری نویز ایزوتروپیک گاوسی به آن،



شکل ۲-۱۲: دورنمایی از ER-classifier [۳۲]



شکل ۲-۱۳: نحوه عملکرد DiffPure [۳۳]

<sup>1</sup>Diffusion Model

<sup>۲</sup> برای اطلاعات بیشتر راجع به مدل های انتشاری می توانید به بخش ۲-۴-۲ مراجعه کنید

<sup>3</sup>Forward Diffusion

<sup>4</sup>Reverse Diffusion

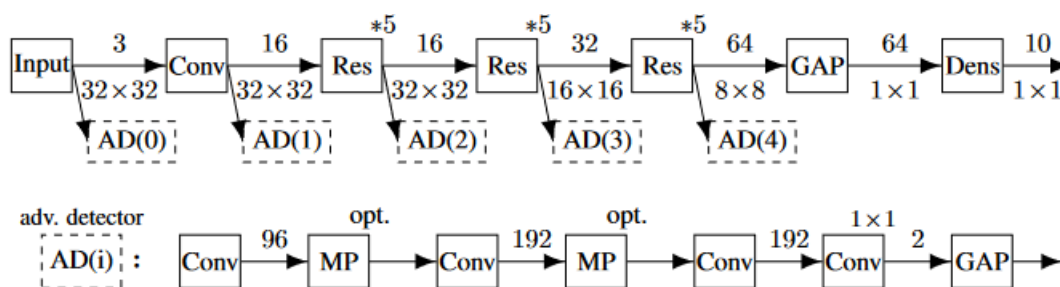
تأثیر نویز تخصصی تزریق شده به ورودی از بین خواهد رفت. سپس برای آن که خروجی دسته‌بند از نویزی بیش از حد اعمال شده به ورودی دچار تغییر نشود، به همان تعداد گام پیش‌روی، فرایند انتشار معکوس انجام خواهد شد تا در نهایت دوباره به ورودی اصلی اما این بار بدون نویز تخصصی دست یافته شود. این ایده به صورت تجربی و همچنین با شهود ریاضی در این مقاله مورد بررسی و به اثبات رسیده است.

## ۲-۳-۲ روش‌های تشخیص حمله

روش‌های تشخیص حمله را می‌توان عموماً از دو جهت بررسی کرد:

۱. تشخیص‌دهنده‌های مبتنی بر ورودی یا خروجی‌های مدل قربانی: ساده‌ترین روش تشخیص نمونه‌های تخصصی، پیاده‌سازی یک دسته‌بند دو-کلاسه برای تمیز دادن نمونه‌های سالم از نمونه‌های تخصصیست. این ایده در [۳۴] مورد بررسی قرار گرفته است. یکی از نقاط قوت اصلی این روش آن است که هیچ پیش‌فرضی راجع به مدلی که از آن دفاع می‌کند ندارد. ضعف اصلی ذکر شده برای این روش در همین مقاله، این است که چنین دسته‌بندی قدرت تعمیم‌پذیری کمی دارد و خصوصاً نسبت به نوع حمله مورد استفاده شده بسیار حساس است. در این پژوهش که از حملات FGSM و JSMA [۳۵] برای ارزیابی مدل استفاده شده است، نتایج تجربی نشان می‌دهند که تشخیص‌دهنده‌هایی که روی حملات FGSM شده‌اند، نمی‌توانند در برابر حملات JSMA به خوبی دفاع کنند و برعکس.

به طور مشابه در [۳۶] از یک زیر-شبکه تشخیص‌دهنده درون دسته‌بند استفاده می‌شود. بدین ترتیب که پس از یکی از لایه‌های پنهان<sup>۱</sup>، شبکه اصلی به دو شاخه تقسیم می‌شود و ویژگی‌های مستخرج از لایه‌ی پنهان قبلی به عنوان ورودی زیر شبکه تشخیص‌دهنده و نیز ادامه دسته‌بند، داده خواهند شد. این معماری در شکل ۱۴-۲ نشان داده شده است. برای آموزش این شبکه، ابتدا دسته‌بند اصلی روی نمونه‌های سالم آموزش داده خواهد شد. سپس، زیرشبکه‌های تشخیص‌دهنده در لایه‌های میانی به دسته‌بند اضافه



شکل ۱۴-۲: معماری تشخیص‌دهنده Metzen [۳۶]

<sup>۱</sup>Hidden Layers

می‌شوند و نمونه‌های تخصصی به تعداد برابر نمونه‌های آموزشی، تشکیل خواهند شد. در نهایت وزن‌های دسته‌بند اصلی freeze شده و فقط زیرشبکه‌های تشخیص‌دهنده روی نمونه‌های سالم و تخصصی به عنوان یک دسته‌بند دو-کلاسه، آموزش داده خواهند شد. در [۳۷] ایده‌ی مشابه ولی با ارتباط بین زیرشبکه‌های دسته‌بند با الهام گرفتن از ایده‌ی Boosting برای دسته‌بندهای آبشاری<sup>۱</sup> [۳۸]، مطرح شده است.

در [۳۹] چارچوبی تحت عنوان I-defender مطرح شده است که در آن توزیع احتمالی خروجی‌های لایه‌های کاملاً متصل<sup>۲</sup> یک مدل دسته‌بند در حین دسته‌بندی نمونه‌های سالم و تخصصی مورد بررسی قرار گرفته است. در این پژوهش نشان داده شده است که توزیع احتمالی لایه‌های پنهان کاملاً متصل نه تنها برای کلاس‌های مختلف مجموعه داده آموزش، تفاوت دارد، بلکه برای یک کلاس در حالت سالم و تخصصی نیز دارای تفاوت‌های چشمگیری هستند. با الهام گرفتن از این نتیجه، محققین این پژوهش ایده‌ی تخمین زدن این توزیع‌های احتمالی را با استفاده از GMM<sup>۳</sup> مطرح می‌کنند. بدین ترتیب برای هر کلاس از کلاس‌های مجموعه داده مورد استفاده،

$$p(\mathcal{H}(x)|\theta, c) = \sum_{k=1}^K w_k \mathcal{N}(\mathcal{H}(x)|\mu_{ck}, \Sigma_{ck})$$

احتمال بروز توزیع احتمالی لایه پنهان کاملاً متصل برای ورودی  $x$  به شرط پارامترهای مدل  $\theta$  و کلاس  $c$  به صورت ترکیب وزن‌داری از توزیع‌های گاوسی تخمین زده می‌شود و سپس با یافتن یک حد آستانه  $TH_c$  برای هر کلاس، می‌توان از

$$Reject(x, c) = p(\mathcal{H}(x)|\theta, c) < TH_c$$

برای رد یا قبول کردن یک ورودی در زمان تست استفاده کرد.

در ML-LOO [۴۰] مفهومی تحت عنوان Feature Attribution برای تشخیص حملات تخصصی مطرح می‌شود. این مفهوم که برای هر ورودی  $x \in \mathbb{R}^d$  با  $\Phi(x) : \mathbb{R}^d \rightarrow \mathbb{R}^d$  نشان نمایش داده شده است، معیاری از میزان تاثیر هر ویژگی از ورودی (برای تصاویر، هر پیکسل از تصویر) در خروجی دسته‌بندی نهاییست. برای اندازه‌گیری  $\Phi$  از روش Leave-One-Out(LOO) استفاده می‌شود، بدین ترتیب که به ازای هر ویژگی از ورودی اختلاف احتمال خروجی محتمل‌ترین کلاس در حالت عادی و در زمانی که آن ویژگی با یک مقدار مرجع (مثلاً ۰) جایگزین شده است، محاسبه می‌شود. به عبارت دقیق‌تر:

$$\Phi(x)_i := f(x)_c - f(x_{(i)})_c, \quad \text{s.t. } c = \arg \max_{j \in C} f(x)_j.$$

<sup>1</sup>Cascade

<sup>2</sup>Fully Connected

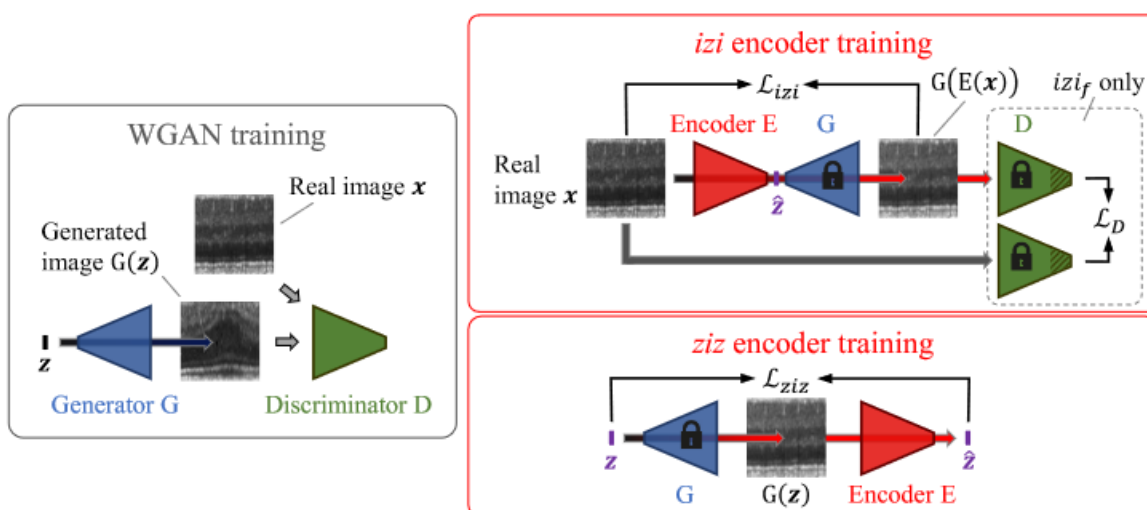
<sup>3</sup>Gaussian Mixture Model



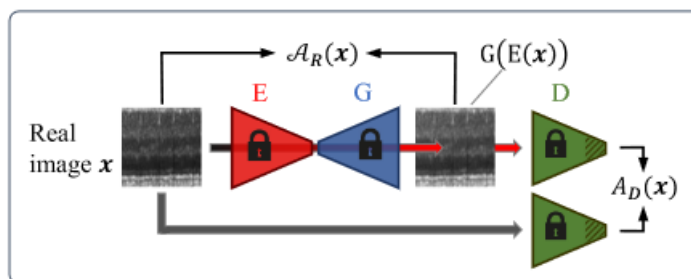
محققین این پژوهش مشاهده کردند که در نمونه‌های تخصصی و سالم تفاوت قابل توجهی بین مقدار  $\Phi$  آن‌ها وجود دارد. از این تفاوت به عنوان معیاری برای تشخیص حملات استفاده می‌شود.

f-AnoGAN [۴۱] چارچوب جالب توجه دیگری با استفاده شبکه‌های مولد تخصصیست که برای منظور خاص تشخیص آنومالی‌ها در تصاویر پزشکی به کار رفته است ولی ایده‌ی اصلی آن می‌تواند قابل تعمیم به هر محیطی باشد. در این روش که در شکل ۲-۱۵ آمده است، ابتدا یک WGAN<sup>۱</sup> روی داده‌های سالم آموزش داده می‌شود. سپس وزن‌های شبکه‌های مولد و ممیز Freeze می‌شوند و یک رمزگذار به کمک این دو در دو مرحله آموزش می‌بیند (مراحل  $izi_f$  و  $ziz$  در شکل). در نهایت برای تشخیص آنومالی از ترکیب دو سنج‌ی  $A_R(x)$  و  $A_D(x)$  استفاده می‌شود (شکل ۲-۱۶):

$$A(x) = A_R(x) + \kappa \cdot A_D(x)$$



شکل ۲-۱۵: دورنمای نحوه آموزش f-AnoGAN [۴۱]



شکل ۲-۱۶: نحوه عملکرد f-AnoGAN در زمان تست [۴۱]

<sup>۱</sup>Wasserstein Generative Adversarial Network

که در آن

$$A_R(x) = \frac{1}{n} \cdot \|x - G(E(x))\|^2$$

معیاری از خطای بازسازی<sup>۱</sup> است چرا که انتظار می رود پس از آموزش رمزگذار  $E$  روی داده‌های سالم، در حالت ایده‌آل، شبکه‌های  $G$  و  $E$  توابع عکس نظیر یک دیگر باشند و بنابراین

$$G(E(x)) \approx x.$$

ولی اگر در نمونه  $x$  آنومالی وجود داشته باشد، از آنجایی که  $E$  فقط می تواند به فضای پنهان نمونه‌های سالم رمز کند، نرُم اختلاف دو مقدار  $x$  و  $G(E(x))$  می تواند سنجه خوبی برای تشخیص آنومالی باشد. از طرف دیگر

$$A_D(x) = \frac{1}{n_d} \cdot \|D_{interm}(x) - D_{interm}(G(E(x)))\|^2$$

که در آن  $D_{interm}(\cdot)$  ویژگی های یکی از لایه‌های میانی شبکه ممیز است، با استدلالی مشابه  $A_R(x)$  و با الهام گرفتن از روش تطبیق ویژگی<sup>۲</sup> [۴۲] به عنوان سنجی دیگر برای تشخیص آنومالی استفاده می شود. در نهایت ترکیب این دو سنجی امتیاز  $A(x)$  را بدست خواهد داد که از آن و با کمک یک حد آستانه تنظیم شده می توان برای تشخیص آنومالی استفاده کرد.

در نهایت نگاهی به روش ارائه شده در [۴۳] خواهیم انداخت<sup>۳</sup>. در این روش - که نزدیک ترین کار انجام شده به کار ماست- بار دیگر از شبکه‌های مولد تخصصی، ولی این بار مشروط بر کلاس<sup>۴</sup> کمک گرفته شده است. در ACGAN-ADA ابتدا یک شبکه ACGAN [۴۴] روی نمونه‌های سالم آموزش داده می شود. سپس از تمام بخش های این شبکه برای ایجاد سنجیهایی برای تشخیص حمله استفاده می شود. به طور مشخص، سنجیهای مورد استفاده موارد زیر هستند:

$$S_R = D(x)$$

$$S_C = p_D(\hat{c}|x)$$

$$S_g = \min_z \|x - G(z|\hat{c})\|^2$$

که در آن  $\hat{c}$  خروجی موقت دسته‌بند مورد دفاع نسبت به ورودی اعمال شده است.  $S_R$  خروجی زیرشبکه ممیز ACGAN و معیاری از واقعی بودن نمونه ورودی با توجه به مشاهدات صورت گرفته در زمان آموزش

<sup>1</sup>Reconstruction Error

<sup>2</sup>Feature Matching

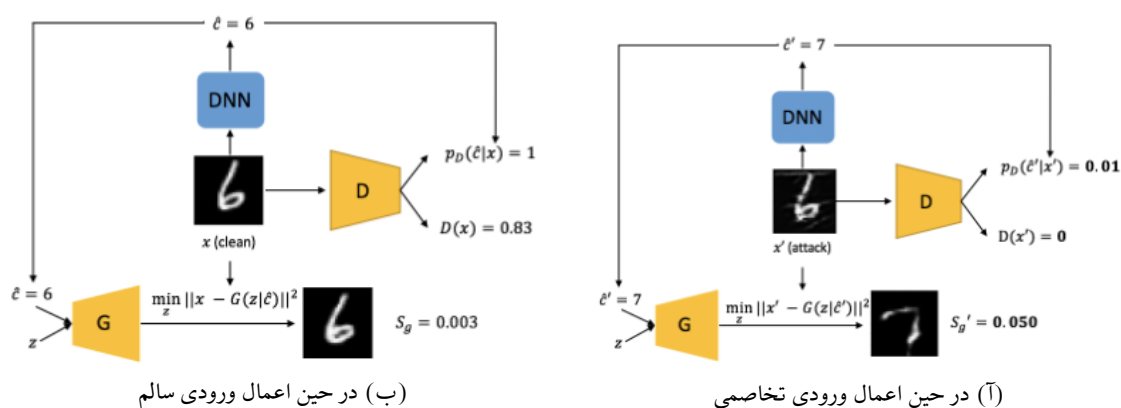
<sup>۳</sup> برای ارجاع دادن در این روش و با توجه به این که محققین اسم خاصی در مقاله به آن نسبت نداده‌اند، از این جا به بعد این روش را ACGAN-ADA خطاب خواهیم کرد

<sup>4</sup>Class Conditional

از داده‌های سالم است. این سنجه حملاتی را که نویز تخصصی آن‌ها دارای نرُم بزرگی باشد، جریمه می‌کند.  $S_C$  احتمال پسین<sup>۱</sup> کلاس تشخیص داده شده توسط مدل مورد دفاع، به شرط ورودی داده شده ولی از دسته‌بند اضافی<sup>۲</sup> شبکه‌ی ACGAN است. مشخص است در صورتی که مقدار  $S_C$  کم باشد احتمال بروز حمله وجود دارد چرا که بین دسته‌بند اضافی و مدل مورد دفاع روی کلاس نهایی توافق کمی وجود دارد. در نهایت  $S_g$  کمینه‌ی نرم افکنش ورودی به شرط کلاس تشخیص داده شده توسط مدل مورد دفاع است. با استدلالی همانند روش ارائه شده در Defense-GAN، کوچک بودن این مقدار به منزله احتمال بیشتری برای سالم بودن ورودیست. نحوه عملکرد این روش در شکل ۲-۱۷ نشان داده شده است. در حین اعمال ورودی سالم (شکل ۲-۱۷ ب) مقدار  $D(x)$  و  $p_D(\hat{\ell}|x)$  بالا بوده در حالی که مقدار  $S_g$  ناچیز است. در مقابل، در حین اعمال ورودی تخصصی (شکل ۲-۱۷ آ) دقیقاً برعکس این حالت اتفاق خواهد افتاد. در ادامه این پژوهش ترکیب‌های مختلفی از این سنجه‌ها را مورد استفاده قرار داده است و بهترین حالت ممکن گزارش شده، حالت D-AD است که فقط از سنجه‌های مربوط به زیرشبکه ممیز ( $S_C$  و  $S_R$ ) استفاده می‌کند.

۲. تشخیص‌دهنده‌های مبتنی بر ویژگی‌های خاص نمونه‌های تخصصی: منظر دیگری که می‌توان برای تشخیص نمونه‌های تخصصی اتخاذ کرد آن است که به جای استفاده از ورودی‌ها و یا خروجی‌های مدل مورد دفاع، به طور مشخص از خواص نمونه‌های تخصصی برای تشخیص آن‌ها استفاده کنیم.

برای تشخیص نمونه‌های تخصصی که دور از خمینه نمونه‌های واقعی قرار دارند روشی تحت عنوان KD-Detection [۴۵] پیشنهاد شد. در این روش از تخمین چگالی هسته‌ای<sup>۳</sup> (KDE) برای تخمین چگالی



شکل ۲-۱۷: عملکرد چارچوب ACGAN-ADA در زمان تست

<sup>1</sup>Posterior

<sup>2</sup>Auxiliary Classifier

<sup>3</sup>Kernel Density Estimation

نمونه‌های سالم از روی ویژگی‌های استخراج شده توسط آخرین لایه پنهان برای کمک به تشخیص نمونه‌های تخاصمی استفاده می‌شود. به طور کلی، تخمین چگالی به عنوان معیاری از فاصله یک نمونه از یک خمینه هدف مورد استفاده قرار می‌گیرد. فرض کنید

$$x_1, x_2, \dots, x_n$$

نمونه‌هایی از نمونه‌های آموزشی هستند که از توزیع احتمالی مجهول و احتمالاً بسیار پیچیده  $p_X(x)$  نمونه برداری شده‌اند. به ازای هر ورودی  $x$ ، می‌توان چگالی تخمین زده شده حول  $x$  را به صورت زیر محاسبه کرد:

$$\hat{p}_X(x) = \frac{1}{n} \sum_{i=1}^n K_\sigma(x, x_i),$$

که در آن  $K_\sigma(\cdot, \cdot)$  یک تابع هسته است. در روش KD-Detection برای هر کلاس یک مدل چگالی هسته‌ای تخمین زده می‌شود به طوری که اگر  $x$  دارای کلاس پیش‌بینی شده  $y$  است، تنها نمونه‌های آموزشی دارای برچسب  $y$  برای تخمین چگالی هسته‌ای آن مورد استفاده قرار می‌گیرند. پس از آن که تمامی مدل‌های KDE روی نمونه‌های سالم آموزش دیده شدند، نمونه‌های زمان تست ابتدا به صورت موقت به دسته‌بند مورد دفاع داده می‌شوند و پس از مشخص شدن برچسب پیش‌بینی شده توسط این مدل (چه درست، چه غلط بر اثر حمله تخاصمی)، این ورودی به KDE متناظر فرستاده خواهد شد و در نهایت از یک مدل Logistic Regression برای تشخیص نمونه‌های تخاصمی از روی امتیاز چگالی آن نمونه خاص توسط KDE متناظر، استفاده می‌شود.

ایده‌ی دیگر مطرح شده در این زمینه، ADA [۴۶] است. بنای اصلی این پژوهش، این فرضیه است که نمونه‌های تخاصمی موفق در فضای مرز تصمیم ایجاد شده توسط دسته‌بند، به اندازه کافی به کلاس هدف شبیه هستند ولی از آنجایی که نرم‌نویز تخاصمی اعمال شده به نمونه باید محدود شده باشد، انتظار است که این نمونه‌های تخاصمی از کلاس نمونه‌ی سالم متناظرشان نیز چندان دور نخواهند بود. در همین راستا، در این پژوهش با استفاده از مدل‌های تخمین چگالی، فضای تولید شده توسط لایه‌های پنهان دسته‌بند مورد دفاع را به صورت ریاضی مدل می‌کنند. سپس از دیورژانس<sup>۱</sup> Kullback-Leibler (KL) [۴۷] بین معیارهای تخمین زده شده توسط خود دسته‌بند و مدل‌های تخمین چگالی و یک حد آستانه مناسب، برای تشخیص حملات استفاده می‌شود.

در [۴۸] روش دیگری تحت عنوان Mahalanobis Detector (MD) مطرح شده است. فرض کنید یک

<sup>۱</sup>Divergence

دسته‌بند عصبی softmax آموزش دیده شده در اختیار داشته باشیم که احتمال پسین

$$P(y = c|x) = \frac{\exp(w_c^T f(x) + b_c)}{\sum_{c'} \exp(w_{c'}^T f(x) + b_{c'})}$$

را تولید می‌کند و  $w_c$  و  $b_c$  وزن‌ها و بایاس‌های لایه آخر دسته‌بند، و  $f(\cdot)$  نمایانگر خروجی لایه‌ی ماقبل آخر دسته هستند. اکنون بدون هیچ تغییری در دسته‌بند آموزش دیده شده، اگر فرض کنیم که توزیع احتمالی نمونه‌ها مشروط بر هر کلاس از یک گاوسی چند متغیره پیروی می‌کند، می‌توان یک دسته‌بند مولد تعریف کرد. به عبارت دقیق‌تر، می‌توان  $C$  توزیع گاوسی با کواریانس مشترک  $\Sigma$  را در نظر گرفت:

$$P(f(x)|y = c) = \mathcal{N}(f(x)|\mu_c, \Sigma)$$

که  $\mu_c$  میانگین مختص به کلاس  $c \in \{1, \dots, C\}$  است. علت منطقی بودن این فرض آن است که می‌توان نشان داد چنین دسته‌بند مولدی تحت Gaussian Discriminant Analysis (GDA) با یک دسته‌بند softmax معادل است. اکنون برای تشکیل معیاری از سطح اطمینان نسبت به یک نمونه از فاصله ماهالانوبیس<sup>۱</sup> بین نمونه  $x$  و نزدیک‌ترین گاوسی مشروط بر کلاس موجود از بین  $C$  گاوسی ممکن، استفاده می‌شود:

$$M(x) = \max_c -(f(x) - \mu_c)^T \Sigma^{-1} (f(x) - \mu_c).$$

در نهایت با قرار دادن یک حد آستانه روی  $M(x)$  می‌تواند تشخیص داد که نمونه مورد آزمون  $x$  سالم و یا دارای آنومالی می‌باشد.

در [۴۹] روش آماری دیگری برای تشخیص نمونه‌های تخاصمی ارائه شده است. اگر logit های پیش از اعمال softmax در یک دسته‌بند با بردار  $f(x)$  نشان داده شوند،  $f_y(x)$  درایه‌ی  $y$ -ام این بردار خواهد بود که به عبارتی دیگر، log-odds برای کلاس  $y$  نیز پنداشته می‌شود. همچنین فرض کنیم که مقادیر جفتی log-odds نیز برای دو کلاس  $y$  و  $z$  به صورت زیر تعریف شوند:

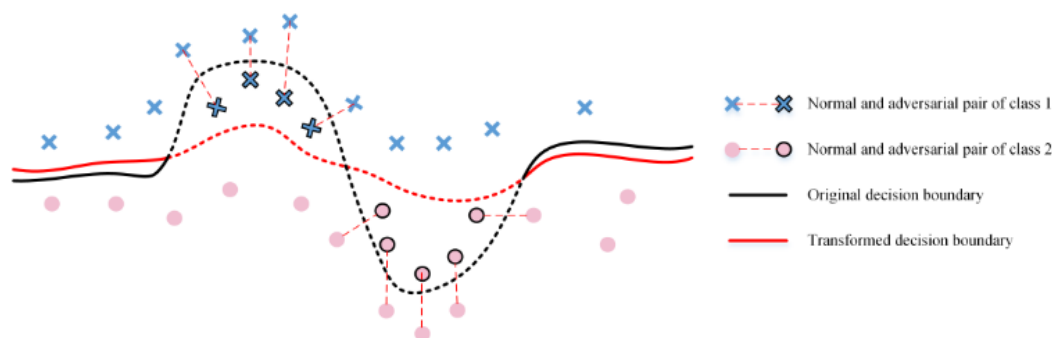
$$f_{y,z}(x) = f_z(x) - f_y(x) \quad (۲-۱۵)$$

اکنون، ایده‌ی اصلی مطرح شده در این پژوهش آن است که این مقدار log-odds جفتی چگونه با افزودن مقدار کمی نویز  $\eta$  به ورودی  $x$  تغییر می‌کنند هنگامی که  $y = y_{true}$  اگر برچسب‌های واقعی در اختیار باشند و هنگامی که  $y = F(x) = \arg \max_y f_y(x)$  در زمان تست. یافته‌های این پژوهش نشان می‌دهند که این مقدار log-odds می‌تواند بسته به این که نمونه‌ی ورودی  $x$  سالم و یا تخاصمی بوده است، به شدت

<sup>۱</sup> Mahalanobis Distance

متفاوت باشد. همچنین، اگر فرض شود که نویز تخصصی افزوده شده به نمونه‌های سالم نسبت به نویز افزوده  $\eta$  مقاوم نیست، بنابراین اگر نمونه تخصصی  $\delta$  در اختیار باشد که  $F(\hat{x}) = \hat{y} \neq F(x) = y$ ، آنگاه انتظار می‌رود که  $f_{\hat{y},y}(\hat{x} + \eta) > f_{\hat{y},y}(\hat{x})$  چرا که نویز کوچک  $\eta$  اضافه شده مقدار کمی از تاثیر نویز تخصصی  $\delta$  را از بین برده و مقدار log-odds جفتی بین کلاس به اشتباه تشخیص داده شده توسط دسته‌بند و کلاس اصل را طبق معادله ۲-۱۵ افزایش می‌دهد. بنابراین از امید ریاضی این مقدار log-odds جفتی (روی  $\eta$  های مختلف) می‌توان به عنوانی معیاری برای تشخیص نمونه‌های تخصصی با استفاده از یک حد آستانه مطلوب بهره برد.

در چارچوبی تحت عنوان SID<sup>۱</sup> [۵۰] این ایده مطرح شده است که نمونه‌های تخصصی از هل دادن نمونه‌های سالم به سمت دیگر مرز تصمیم در جاهایی که مرز تصمیم یک دسته‌بند دارای تلاطم زیاد است، تشکیل می‌شوند (شکل ۲-۱۸). برای حل این مشکل ایده‌ی مطرح شده آن است که مرز تصمیم دسته‌بند را طوری تغییر داد که دیگر نتوان با تغییرات اندک نمونه‌های سالم را به سوی دیگر مرز تصمیم هل داد و موجب تشکیل نمونه‌های تخصصی شد بدون آن که مرز تصمیم در نقاط کم تلاطم دچار تغییرات چشم‌گیری بشود. به طور دقیق‌تر، اگر  $B_{i,j}$  مرز تصمیم بین دو کلاس  $i$  و  $j$  در دسته‌بند اصلی ( $\mathcal{F}$ ) باشد، و  $\tilde{B}_{i,j}$  مرز تصمیم دسته‌بند جدید ( $\mathcal{G}$ ) باشد، مطلوب آن است که  $B_{i,j}$  و  $\tilde{B}_{i,j}$  در همه‌ی نقاط، به جز در اطراف نقاط پر تلاطم  $B_{i,j}$  شبیه یک دیگر باشند که این مطلوب را می‌توان به تولید بردارهای احتمال مشابه در دو دسته‌بند در تمام نقاط غیر از نقاط مذکور، کاهش داد. این مسئله را می‌توان به صورت یک مسئله بهینه‌سازی کمینه-بیشینه بیان کرد که هزینه متحمل شده در بدترین حالت اختلاف مرز تصمیم



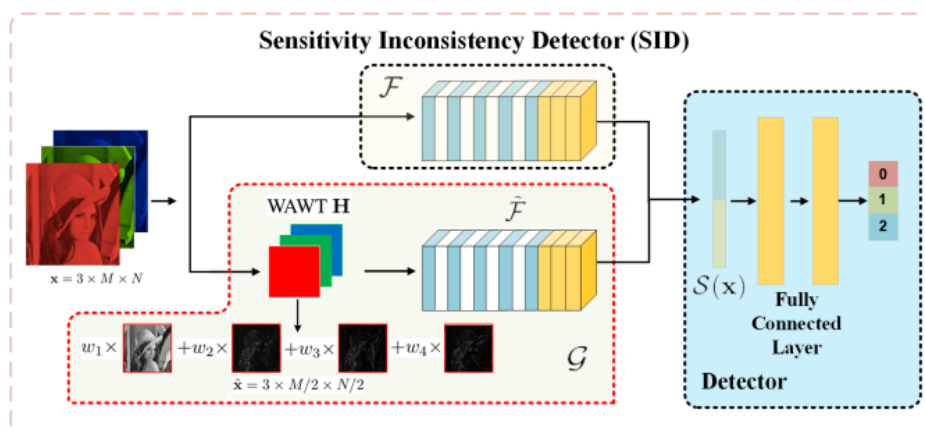
شکل ۲-۱۸: تشکیل نمونه‌های تخصصی در اطراف خمیدگی‌های مرز تصمیم یک دسته‌بند [۵۰]

<sup>۱</sup> Sensitivity Inconsistency Detector

ایجاد شده توسط  $\mathcal{F}$  و  $\mathcal{G}$  را حداقل می‌کند:

$$\min_{\mathcal{G}} \max_{x \in X} \|\mathcal{F}(x) - \mathcal{G}(x)\|_2^2, \text{ s.t. } \|\mathcal{F}(\hat{x}) - \mathcal{G}(\hat{x})\|_2^2 \geq \xi, \quad \forall \hat{x} \in \hat{X} \quad (۱۶-۲)$$

که در آن  $X$  مجموعه نمونه‌های سالم،  $\hat{X}$  مجموعه نمونه‌های تخصصی و  $\xi$  یک متغیر لنگی<sup>۱</sup> است که به  $\mathcal{G}$  اجازه می‌دهد در اطراف نقاط پر تلاطم مرز تصمیم  $\mathcal{F}$  به بتواند به اندازه کافی از آن فاصله بگیرد. برای آن که دسته‌بند جدید بتواند مرز تصمیم مطلوب ارائه شده در معادله ۱۶-۲ را برآورده کند، در این پژوهش از تبدیل Wavelet با میانگین وزن دار<sup>۲</sup> استفاده می‌شود. در نهایت، مطابق شکل ۱۹-۲ خروجی‌های دو دسته‌بند به یک شبکه تشخیص‌دهنده‌ی نهایی تحویل داده می‌شوند و انتظار می‌رود که بر اساس اختلاف مرز تصمیم‌های ایجاد شده توسط دو دسته‌بند، شبکه تشخیص‌دهنده بتواند سالم و یا تخصصی بودن یک نمونه‌ی ورودی را تشخیص دهد.



شکل ۱۹-۲: عملکرد SID در زمان تست [۵۰]

<sup>۱</sup>Slack Variable

<sup>۲</sup>Weighted Average Wavelet Transform (WAWT)

## ۴-۲ مختصری در مورد هوش مولد

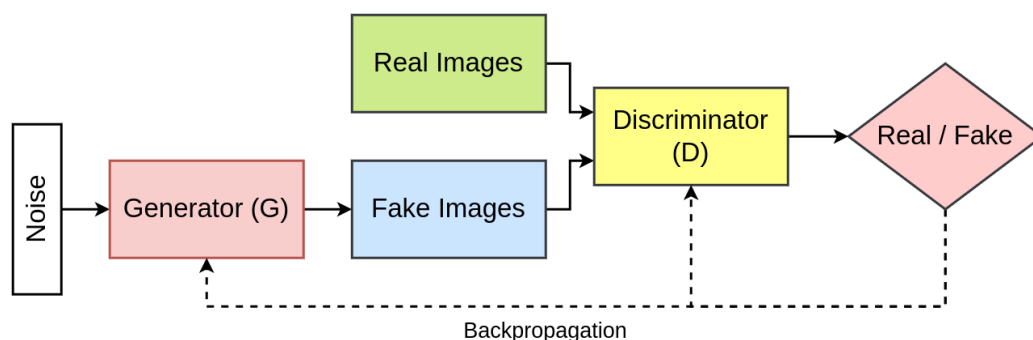
هوش مولد<sup>۱</sup> عبارتست که به یکی از زیر شاخه های هوش مصنوعی اطلاق می شود که تمرکز اصلی آن تولید رسانه های مختلف از جمله متن، تصویر، فیلم و مدل های سه بعدی و غیره است. در هوش مولد از محتوای موجود کنونی و برای آموزش مدل هایی استفاده می شود که توانایی خلق محتوای جدید را داشته باشند بدون آن که لزوماً نمونه های آموزشی را دقیقاً در خروجی تکرار کنند. در حقیقت مدل های مولد تخمینی پارامتری از توزیع احتمالی پیچیده محتوایی که قرار است تولید کنند را یاد می گیرند و در نهایت روشی برای نمونه گیری از این تخمین پارامتری به ما ارائه خواهند کرد.

در ادامه این بخش به بررسی دو مورد از اصلی ترین مدل های مولد در حوضه ی تصویر خواهیم پرداخت که همچنان به طور گسترده مورد پژوهش و تحقیق فعال هستند: شبکه های مولد تخصصی و مدل های انتشاری.

### ۱-۴-۲ شبکه های مولد تخصصی

شبکه های مولد تخصصی (GAN) اولین بار در پژوهش معروف [۵۱] توسط آقای Goodfellow و همکارانش، در یکی از هوشمندانه ترین استفاده های نظریه ی بازی ها در هوش مصنوعی، معرفی شدند. بنای اصلی این شبکه ها به عنوان چارچوبی جدید برای یادگیری بدون نظارت<sup>۲</sup> و به عنوان گام بعدی در مدل های مولد بعد از خود رمزگذارها، نهاده شده.

همانطور که در شکل ۲-۲۰ نشان داده شده است، ایده ی کلی GAN ها برقراری یک بازی مجموع-صفر<sup>۳</sup> بین دو شبکه به نام های مولد<sup>۴</sup> و ممیز است. هدف شبکه ی مولد در حالت ایده آل تولید تصاویری از توزیع تصاویر حقیقی موجود است. برآورده کردن این هدف به کمک بازی ایجاد شده میان مولد و ممیز صورت می گیرد.



شکل ۲-۲۰: دورنمای شبکه های مولد تخصصی

<sup>1</sup>Generative AI

<sup>2</sup>Unsupervised Learning

<sup>3</sup>Zero-sum

<sup>4</sup>Generator



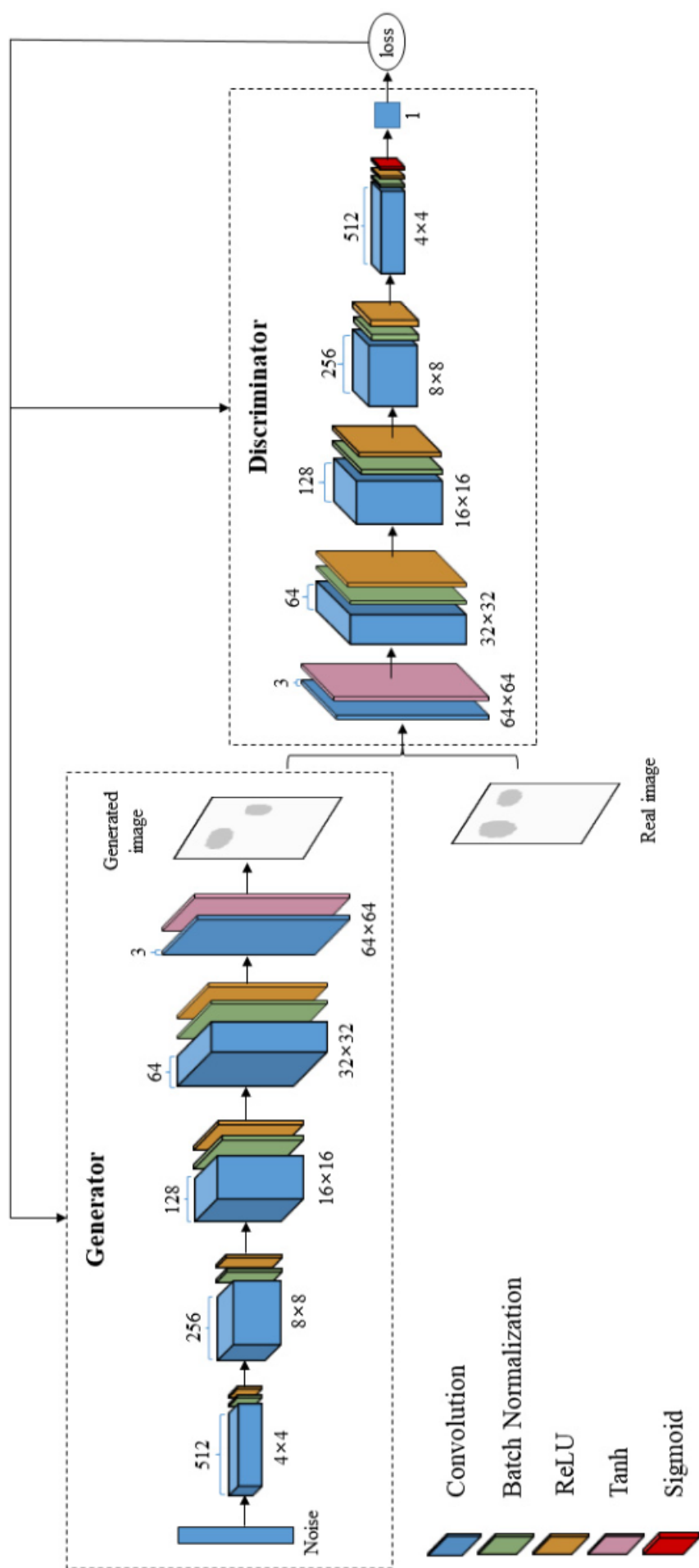
مولد تلاش می‌کند با تولید نمونه‌هایی که به نمونه‌های واقعی شبیه هستند، ممیز را فریب دهد. از طرف دیگر، ممیز در تلاش است که با تقویت خودش در برابر فریب خوردن از مولد مصون بماند و همچنان بتواند نمونه‌های ساختگی و واقعی را از هم تفکیک کند. به طور دقیق‌تر، مولد  $G$  و ممیز  $D$  بازی کمینه-بیشینه زیر را با تابع مقدار  $V(D, G)$ ، انجام خواهند داد:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (۱۷-۲)$$

که در آن  $p_{data}$  توزیع احتمالی آرمانی نمونه‌های واقعی و  $p_z$  توزیع احتمالی بردار پنهانیست که مدل مولد بر پایه‌ی آن نمونه‌های جدید تولید می‌کند. اگر این بازی به صورت پایدار بین دو بازیکن مولد و ممیز بازی شود، در نهایت می‌توان نشان داد که با استفاده از فرمول بندی مناسب برای بازی، نقطه تعادل بهینه بازی در جایی اتفاق خواهد افتاد که توزیع آماری نمونه‌های تولید شده توسط مولد دقیقاً با توزیع نمونه‌های واقعی برابر باشد (قضیه ۱ در [۵۱]).

در ادامه به طور خاص نگاهی مختصر به شبکه‌های مولد تخصصی عمیق کانولوشنی (DCGAN) خواهیم داشت که اولین نمونه شبکه‌های GAN با هدف تولید تصاویر بودند. همانطور که در شکل ۲-۲۱ مشاهده می‌شود، مولد یک DCGAN از سری کردن چندین لایه Upsampling (که در اولین نمونه‌های DCGAN صرفاً با استفاده از کانولوشن ترانهاد<sup>۱</sup> پیاده سازی می‌شدند) به همراه تابع فعال ساز ReLU (به غیر از در آخرین لایه که از tanh استفاده می‌کند) به دست می‌آید. با شروع از یک بردار پنهان (یا نویز) که از یک توزیع تصادفی نمونه برداری می‌شود، در نهایت به یک تصویر سه-کاناله می‌رسیم. در مقابل شبکه ممیز ساختاری دقیقاً عکس مولد را اتخاذ می‌کند. با شروع از یک تصویر سه-کاناله و اعمال پی در پی لایه‌های کانولوشن به همراه فعال ساز ReLU و در نهایت یک sigmoid در لایه آخر، به خروجی ممیز خواهیم رسید که عملاً معیاری از احتمال واقعی بودن تصویر دریافت شده توسط ممیز است. شبکه‌های ممیز با گرفتن دسته‌ای از نمونه‌های واقعی و ساختگی مولد که دارای برجسب‌های به ترتیب ۱ و ۰ هستند، با استفاده از یک تابع هزینه Binary Crossentropy آموزش خواهد دید در حالی که شبکه‌ی مولد ثابت در نظر گرفته می‌شود. سپس، با ثابت در نظر گرفتن ممیز، شبکه مولد سعی می‌کند همان تابع هزینه را ماکزیمم کند تا ممیز را به اشتباه بیاندازد. بدین ترتیب دو شبکه به صورت متوالی تا رسیدن به همگرایی آموزش خواهند دید. در شکل ۲-۲۲ نمونه‌های تولید شده از این شبکه‌ی GAN را می‌توان مشاهده کرد. با وجود پیشرفت‌های ژگرفی که در راستای بهبود GAN‌ها صورت گرفته است، این شبکه‌ها همچنان از سه ایراد اصلی رنج می‌برند [۵۲، ۵۳]:

<sup>۱</sup> Transposed Convolution

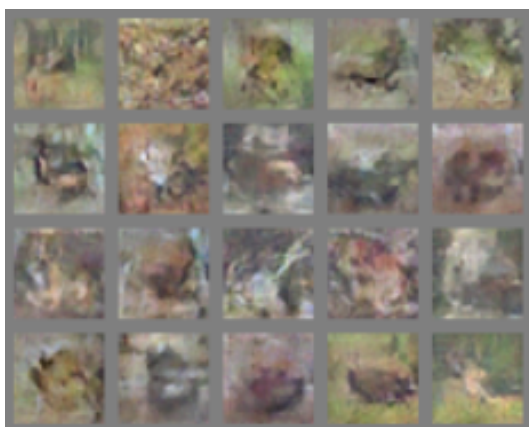


شکل ۲-۲۱: معماری شبکه DCGAN

۱. **فروپاشی مد<sup>۱</sup>**: اگر در حین فرآیند آموزش به طور اتفاقی مولد بتواند نمونه‌ی ساختگی بسیار خوبی تولید کند، بازخوردی که از ممیز بابت این نمونه‌ی به خصوص دریافت می‌کند بسیار مثبت خواهد بود. به عبارت دیگر از آنجایی که این یک نمونه‌ی خاص احتمالاً می‌تواند ممیز را فریب دهد، مولد تشویق به تولید نمونه‌های مشابه این نمونه می‌شود. اگر هیچ مکانیزمی برای جلوگیری از این اتفاق وجود نداشته باشد، در بدترین حالت مولد یاد می‌گیرد که کل فضای حالت بردار پنهانی که به عنوان ورودی دریافت می‌کند را به یک نمونه یا چندین نمونه‌ی بسیار مشابه نگاشت کند. به چنین حالتی که مولد فقط یکی از مدهای توزیع احتمالی ورودی را یاد می‌گیرد، فروپاشی مد گفته می‌شود.

۲. **عدم همگرایی<sup>۲</sup>**: رسیدن به نقطه‌ی تعادل تعادل نش<sup>۲</sup> در GANها به دلیل ساختار بازی کمینه-بیشینه برقرار شده بین دو شبکه و پیچیدگی حل یک مسئله بهینه‌سازی نقطه زینی، امری آرمانیست. از آنجایی که دو بازیکن این بازی مجبورند به نوبت بازی کنند، باید تعادل دقیقی بین مولد و ممیز برقرار باشد تا بتوان به نقطه تعادل آرمانی نزدیک شد. در شرایط واقعی، کنترل کردن تمامی این پارامترها امری بسیار زمان بر و گاهی غیر ممکن است و بنابراین ممکن است یک GAN هیچوقت به نقطه تعادل نرسد. در بدترین حالت این امکان وجود دارد که یکی از دو شبکه به واگرایی میل کند که به تبع کیفیت خروجی‌های نهایی مولد را به شدت کاهش خواهد داد.

۳. **سستی فرایند آموزش**: حالت مقابل فروپاشی مد را تصور کنید. اگر در چندین گام اول آموزش مولد خروجی‌های مولد بسیار دور از فضای ورودی‌های واقعی باشد، از آنجایی که وظیفه‌ی ممیز نسبت به



(ب) نمونه‌های تولید شده روی مجموعه داده CIFAR



(آ) نمونه‌های تولید شده روی مجموعه داده MNIST

شکل ۲-۲۲: نمونه‌های تولید شده توسط یک GAN معمولی [۵۱]

<sup>1</sup>Mode Collapse

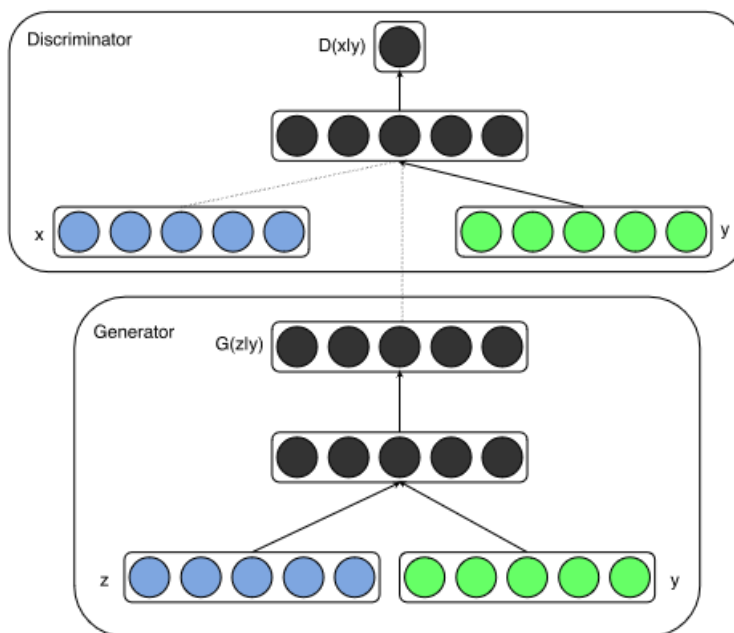
<sup>2</sup>Nash Equilibrium

مولد به مراتب ساده‌تر است، ممیز به سرعت از مولد در بازی پیشی می‌گیرد و می‌تواند به راحتی نمونه‌های ساختگی و واقعی را از هم تفکیک کند. بنابراین از آنجایی که خروجی ممیز به ازای هر ورودی دلخواه عددی بسیار نزدیک به صفر (برای ورودی‌های ساختگی) و یا بسیار نزدیک به ۱ (برای ورودی‌های واقعی) است، مولد دچار پدیده‌ی گرادیان محو شونده<sup>۱</sup> می‌شود و نمی‌تواند باز خورد مفیدی از ممیز برای تقویت خود دریافت کند.

در سالیان پس از گسترش استفاده از GANها راه حل‌های زیادی برای حل پاره‌ای از مسائل ذکر شده، ارائه شده است. یکی از اولین بهبودهایی که خصوصاً برای حل مشکل فروپاشی مد مطرح شد اما نشان داده شده است که می‌تواند روی بهبود همگرایی و پایداری فرایند آموزش نیز تاثیرگذار باشد [۵۴، ۵۵]، مشروط کردن خروجی مولد به برچسب مورد انتظار است. این ایده منجر به معرفی ساختارهای جدیدی به نام‌های cGAN و کمی بعدتر ACGAN شد که در ادامه کمی بیشتر این دو ساختار را مورد بررسی قرار خواهیم داد.

#### ۲-۴-۱-۱ cGAN

مشروط کردن شبکه‌های مولد و ممیز یک GAN ایده‌ای بود که در خود مقاله اصلی معرفی GANها به عنوان راستایی برای پژوهش‌های آینده معرفی شده بود و ظرف مدت چند ماه، اولین تحقیق در این زمینه به ثمر رسید و در [۵۶] محققین معماری Conditional Generative Adversarial Network (cGAN) را معرفی کردند. ایده‌ی



شکل ۲-۲۳: معماری cGAN [۵۶]

<sup>۱</sup> Vanishing Gradient

تزریق یک شرط به ورودی‌های مولد و ممیز در این پژوهش به صورت بسیار ابتدایی و بدیهی مطرح شده است و به عنوان عامل مشروط کننده از برچسب کلاس نمونه‌ها استفاده می‌شود. مطابق شکل ۲-۲۳ برای تزریق این شرط در هر دو شبکه‌ی مولد و ممیز یک لایه‌ی embedding قابل یادگیری معرفی می‌شود که برچسب را به فضای چند بعدی دلخواه ببرد. پس از دریافت embedding متناظر با برچسب ورودی، این بردار به بردار ورودی چسبانده<sup>۱</sup> می‌شود و لایه‌های بعدی مانند یک GAN معمولی عمل خواهند کرد. در این وضعیت مولد  $G$  و ممیز  $D$  توزیع‌های شرطی  $G(z|y)$  و  $D(x|y)$  را تخمین می‌زنند و بازی ۲-۱۷ به شکل ساختار یافته تر زیر در خواهد آمد:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x|y)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z|y)))]$$

که پایداری بیشتری نسبت به بازی یک GAN معمولی دارد و منجر به تولید خروجی‌های با کیفیت تری می‌شود. علاوه بر آن، همانطور که در هر سطر از شکل ۲-۲۴ مشاهده می‌شود، مولد با استفاده از این روش قادر به یادگرفتن اطلاعاتی راجع به برچسب مورد انتظار است و می‌تواند خروجی‌هایی متناظر با برچسب ورودی تولید کند.



شکل ۲-۲۴: تاثیر تغییر برچسب ورودی مولد در نمونه‌های تولید شده توسط یک cGAN. هر سطر از شکل متناظر با یک برچسب ورودی در مولد است [۵۶].

<sup>۱</sup> Concatenate

در [۴۴] برای تولید مشروط بر کلاس تصاویر با استفاده از GAN معماری‌ای تحت عنوان ACGAN ارائه شده است. ایده‌ی اصلی این پژوهش استفاده از یک دسته‌بند اضافی برای مجبور کردن مدل ممیز به تشخیص کلاس نمونه‌های ورودی است. مزیت این روش آن است که نمونه‌های ساختگی مولد به سمت مدهای قابل دسته‌بندی هل داده می‌شوند و انتظار می‌رود که علاوه بر بالا رفتن کیفیت نمونه‌های تولید شده، فرایند آموزش شبکه نیز پایدارتر بشود. در حقیقت، همانطور که در شکل ۲-۲۵ نیز می‌توان مشاهده کرد، در این معماری فقط مولد به طور مستقیم برچسب مورد انتظار را به عنوان ورودی دریافت می‌کند و ممیز برخلاف حالت عادی دو توزیع احتمال در خروجی تولید خواهد کرد:

۱.  $P(S|x)$ ، احتمال واقعی بودن نمونه‌ی ورودی.

۲.  $P(C|x)$ ، توزیع احتمالی روی تمامی کلاس‌های ممکن برای نمونه‌ی ورودی اعمال شده.

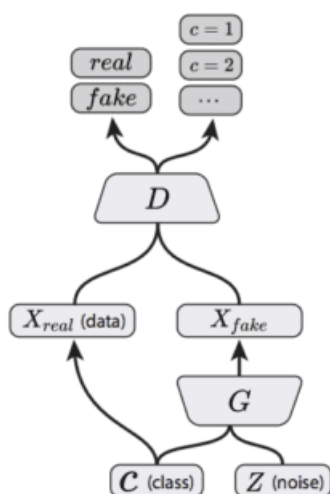
به تبع این موضوع، تابع هزینه‌ی آموزش یک ACGAN از دو بخش تشکیل می‌شود:

$$L_S = -(\mathbb{E}[P(S = real|X_{real})] + \mathbb{E}[P(S = fake|X_{fake})])$$

و

$$L_C = -(\mathbb{E}[P(C = c|X_{real})] + \mathbb{E}[P(C = c|X_{fake})])$$

که  $L_S$  ممیز را مجبور به تشخیص درست نمونه‌های ساختگی و نمونه‌های واقعی خواهد کرد و  $L_C$  بدون توجه به واقعی یا ساختگی بودن نمونه، ممیز را ترغیب به تشخیص کلاس درست ورودی دریافت شده خواهد کرد.



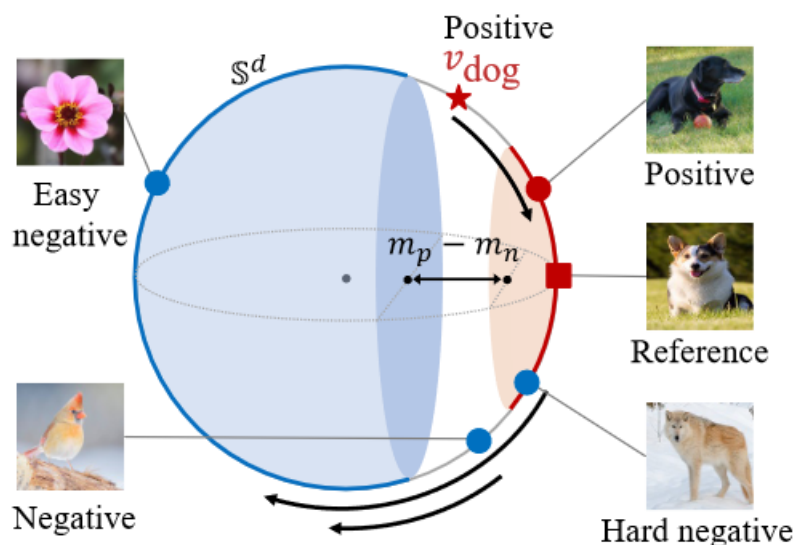
شکل ۲-۲۵: معماری ACGAN

بدین ترتیب ممیز در راستای کمینه کردن  $L_S + L_C$  و مولد در راستای کمینه کردن  $L_C - L_S$  آموزش خواهند دید. با وجود بهبودهای این روش نسبت به cGAN، این معماری همچنان خالی از ایراد نبود. یکی از اصلی‌ترین ایراد های وارد به این معماری، عدم توانایی مولد در یادگیری هنگامیست که تعداد کلاس‌های مجموعه داده‌ها زیاد می‌شود.

## ۳-۱-۴-۲ ReACGAN

در [۵۷] معماری جدیدی برای تولید مشروط تصاویر به نام ReACGAN در راستای رفع دو ایراد اساسی ACGAN ها ارائه شد. محققین در این پژوهش، دو ایراد اصلی معماری ACGAN را چنین بر می‌شمارند:

۱. عدم پایداری آموزش زمانی که تعداد کلاس‌های مجموعه‌ی داده‌ها زیاد باشد.
  ۲. نمونه‌های تولید شده توسط ACGAN دارای تنوع نسبتاً کم و به راحتی قابل تشخیص و دسته‌بندی هستند.
- علت بروز مشکل اول، انفجار گرادیان در دسته‌بند جانبی ACGAN در گام‌های اولیه‌ی فرایند آموزش مطرح شده است. در این پژوهش به صورت تجربی نشان داده می‌شود که افکندن ورودی‌های دسته‌بند جانبی ACGAN به یک ابرکری شعاع واحد می‌تواند از این مشکل جلوگیری کند. در ادامه برای برطرف کردن ایراد دوم و مجبور کردن مولد به تولید نمونه‌های متنوع، جمله‌ی جدیدی تحت عنوان Data-to-Data Cross-Entropy (D2D-CE) به عنوان تابع هزینه‌ی آموزش جدید مدل معرفی می‌شود. هدف این تابع هزینه، استفاده از ویژگی‌های مستتر در خود نمونه‌های آموزش با در نظر گرفتن تفاوت‌ها و شباهت‌های آن‌ها نسبت به فاصله‌ی آن‌ها در فضای embedding است. همانطور که در شکل ۲-۲۶ مشاهده می‌شود فضای embedding یک ReACGAN برخلاف روش‌های



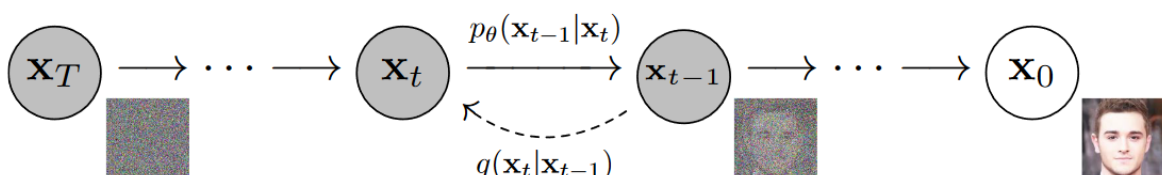
شکل ۲-۲۶: فضای embedding تصاویر در یک ReACGAN

قبلی (مانند ACGAN) روی یک ابرکری  $\mathbb{S}^d$  قرار می گیرد. از فواید مهم تابع هزینه D2D-CE که در همین شکل مشخص هستند می توان به استخراج نمونه های منفی دشوار و سرکوب کردن نمونه های مثبت و منفی آسان اشاره کرد. اگر تصویری به عنوان مرجع<sup>۱</sup> در اختیار باشد، مثلاً تصویری با کلاس سگ در شکل ۲-۲۶، یک نمونه ی منفی دشوار، نمونه ای خواهد بود که از لحاظ ویژگی های تصویری بسیار مشابه کلاس مرجع باشد (مثلاً تصویر یک گرگ). از طرف دیگر، یک نمونه ی منفی آسان در نقطه ی مقابل یک منفی دشوار قرار می گیرد (مثلاً تصویر یک گل). نهایتاً یک نمونه ی مثبت آسان، نمونه ایست که مانند یک منفی دشوار، ویژگی های مشترک زیادی با تصویر مرجع داشته باشد ولی برچسب آن نیز با برچسب مرجع برابر باشد. برای ایجاد تنوع در نمونه های تولید شده D2D-CE تلاش می کند که embedding نمونه های مثبت آسان را بیش از حد به هم نزدیک نکند تا کمی واریانس در خروجی های مولد باقی بماند. همچنین از طرف دیگر برای این که نمونه های تولید شده در حین داشتن تنوع، همچنان متناظر با برچسب های صحیحشان باشند.

## ۲-۴-۲ مدل های انتشاری

دسته ی دیگری از مدل های مولد که اخیراً معرفی شده اند و به سرعت در حال گسترش هستند، مدل های انتشاری هستند. ایده ی مدل های انتشاری از مبحثی در فیزیک تحت عنوان ترمودینامیک غیرتعادلی<sup>۲</sup> الگو برداری شده است. با وجود این که اولین پژوهش در رابطه با مدل های انتشاری در سال ۲۰۱۵ و تقریباً به صورت همزمان با GAN منتشر شد [۵۸]، آوازه این مدل ها اولین بار در ۲۰۲۰ پس از انتشار مقاله معروف [۵۹] تحت عنوان مدل های انتشار احتمالی حذف کننده نویز<sup>۳</sup> بر سر زبان ها آمد. مدل های انتشاری یک زنجیره مارکوف<sup>۴</sup> از گام های انتشار را تعریف می کنند که به تدریج مقداری نویز تصادفی به یک نمونه می افزایند و سپس یاد می گیرند که عکس این فرایند انتشار را انجام دهند تا بتوانند با شروع از نویز به یک نمونه ی ساختگی که به توزیع نمونه های واقعی نزدیک است، برسند (شکل ۲-۲۷).

همان طور که پیش تر توضیح داده شد، مدل های انتشاری از دو فرایند انتشار پیش رو و انتشار معکوس تشکیل



شکل ۲-۲۷: زنجیره مارکوف نظیر یک DDPM [۵۹]

<sup>1</sup>Reference

<sup>2</sup>Non-equilibrium Thermodynamics

<sup>3</sup>Denoising Diffusion Probabilistic Model (DDPM)

<sup>4</sup>Markov Chain



می‌شوند. در ادامه به طور مختصر به توزیع این دو فرایند پرداخته خواهد شد [۶۰، ۶۱]:

۱. **انتشار پیش‌رو:** فرض کنید یک نمونه داده از توزیع نمونه‌های واقعی  $q(x)$   $x_0 \sim$ ، در اختیار باشد. در فرایند انتشار پیش‌رو مقادیر کمی نویز گاوسی به این نمونه در  $T$  گام زمانی افزوده می‌شود که به ترتیب نمونه‌های نویزی شده‌ی  $x_1, \dots, x_T$  را تولید خواهد کرد. مقدار واریانس این نویزهای افزوده شده در هر گام زمانی توسط برنامه‌زمانی  $\{\beta_t \in (0, 1)\}_{t=1}^T$  کنترل می‌شود. اگر توزیع احتمالی فرایند انتشار پیش‌رو را با  $q$  نشان دهیم، طبق تعریف و با کمی محاسبات خواهیم داشت:

$$q(x_t | x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t} x_{t-1}, \beta_t \mathbf{I}) \quad q(x_{1:T} | x_0) = \prod_{t=1}^T q(x_t | x_{t-1}) \quad (۱۸-۲)$$

بدین ترتیب، همانطور که در شکل ۲-۲۷ نیز نشان داده شده است به مرور زمان و با بزرگ شدن  $t$  نمونه‌ی  $x_0$  تخریب می‌شوند و اگر  $T \rightarrow \infty$ ، می‌توان نشان داد که  $x_T$  نمونه‌ای از توزیع یک گاوسی ایزوتروپیک است.

۲. **انتشار معکوس:** حال اگر بتوان به نحوی معکوس پروسه‌ی انتشار پیش‌رو  $q$  را یاد گرفت و به جای نمونه‌گیری از  $q(x_t | x_{t-1})$  از  $q(x_{t-1} | x_t)$  نمونه برداری کرد، می‌توان با شروع از نمونه‌ای از یک گاوسی ایزوتروپیک و اعمال  $T$  گام زمانی معکوس، به نمونه‌ای از توزیع واقعی  $q(x)$  رسید. می‌توان نشان داد که اگر  $\beta_t$  به اندازه کافی کوچک باشد، انتشار معکوس  $q(x_{t-1} | x_t)$  نیز تقریباً یک گاوسی است. ولی از آنجایی که محاسبه‌ی  $q(x_{t-1} | x_t)$  نیازمند استفاده از کل مجموعه‌ی داده است (به دلیل وجود جمله‌ی امید ریاضی)، مجبوریم از یک مدل ریاضی  $p_\theta$  برای تخمین  $q$  استفاده کنیم. می‌توان نشان داد که  $p_\theta$  بدین صورت تعریف می‌شود:

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^T p_\theta(x_{t-1} | x_t) \quad p_\theta(x_{t-1} | x_t) = \mathcal{N}(x_{t-1} | \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \quad (۱۹-۲)$$

در [۵۹] نشان داده می‌شود که می‌توان به جای تخمین زدن پارامترهای گاوسی تعریف شده در معادله ۱۹-۲ هدف آموزش مدل  $p_\theta$  را طوری تغییر داد که فقط نیاز به تخمین نویز اضافه شده  $\epsilon_t$  با در اختیار داشتن  $x_t$  در گام زمانی  $t$  باشیم که  $x_{t-1} = x_t + \epsilon_t$

بدین ترتیب از آنجایی که ابعاد نویز  $\epsilon_t$  با ابعاد ورودی  $x_t$  برابر به یک معماری U-Net [۶۲] مانند نیاز خواهد بود که وظیفه‌ی آن تخمین زدن نویز اضافه شده به نمونه‌ی ورودی در هر گام زمانی  $1 \leq t \leq T$  است. از آنجایی که هدف آموزشی مدل‌های انتشاری مطابق تخمین بیشینه درست‌نمایی<sup>۱</sup> انجام می‌شود که به مراتب

<sup>۱</sup>Maximum Likelihood Estimation

ساختار یافته تر از بازی کمینه-بیشینه بین مولد و ممیز در یک GAN است، آموزش مدل‌های انتشاری پایدار تر بوده و مشکلاتی مانند فروپاشی مُد معمولاً در این مدل‌ها مشاهده نمی‌شود [۶۱]. از طرف دیگر، همانطور که پیش‌تر توضیح داده شد مقدار  $\beta_t$  در هر گام زمانی باید به اندازه کافی کوچک باشد تا بتوان انتشار معکوس را توسط یک گاوسی تخمین زد و بنابراین به مقدار  $T$  (تعداد کل گام‌های زمانی فرایند انتشار) باید بزرگ باشد (مثلاً ۱۰۰۰ گام) و در نتیجه برای تولید هر نمونه به تعداد قابل توجهی forward pass نیاز است که می‌تواند هزینه بر باشد. با این وجود، نمونه‌گیری از DDPM‌ها در تعداد گام‌های زمانی کمتر همچنان یک موضوع تحقیق فعال است و پژوهش‌هایی (مانند DDIM [۶۳]) در این زمینه صورت گرفته است که می‌تواند تعداد گام‌های مورد نیاز برای نمونه برداری را به تعداد بسیار کمی کاهش دهد [۶۰].

در شکل ۲-۲۸ نمونه‌های تولید شده توسط یک DDPM را روی مجموعه داده Celeb-HQ می‌توانید مشاهده می‌کنید. در ادامه کمی راجع به مدل‌های انتشاری مشروط صحبت خواهیم کرد.

#### ۱-۲-۴-۲ مدل‌های انتشاری هدایت شده (مشروط)

مانند GAN‌ها تولید مشروط محتوا بر مبنای شروطی مانند برچسب کلاس و یا متن توصیف‌کننده‌ی محتوای مورد نظر، امری مطلوب است. در رابطه با مدل‌های انتشاری برای تحقق این هدف دو راستای مهم تحقیق صورت گرفته است:

۱. مدل انتشاری هدایت شده با دسته‌بند<sup>۱</sup>: ایده‌ی ارائه شده در [۶۴] این بود که ابتدا یک دسته‌بند  $f_\phi(y|x_t)$  روی تصاویر نویزی  $x_t$  آموزش داده شده و سپس از لگاریتم گرادیان خروجی این دسته‌بند نسبت به تصویر



شکل ۲-۲۸: نمونه‌های تولید شده توسط DDPM روی مجموعه داده Celeb-HQ [۵۹]

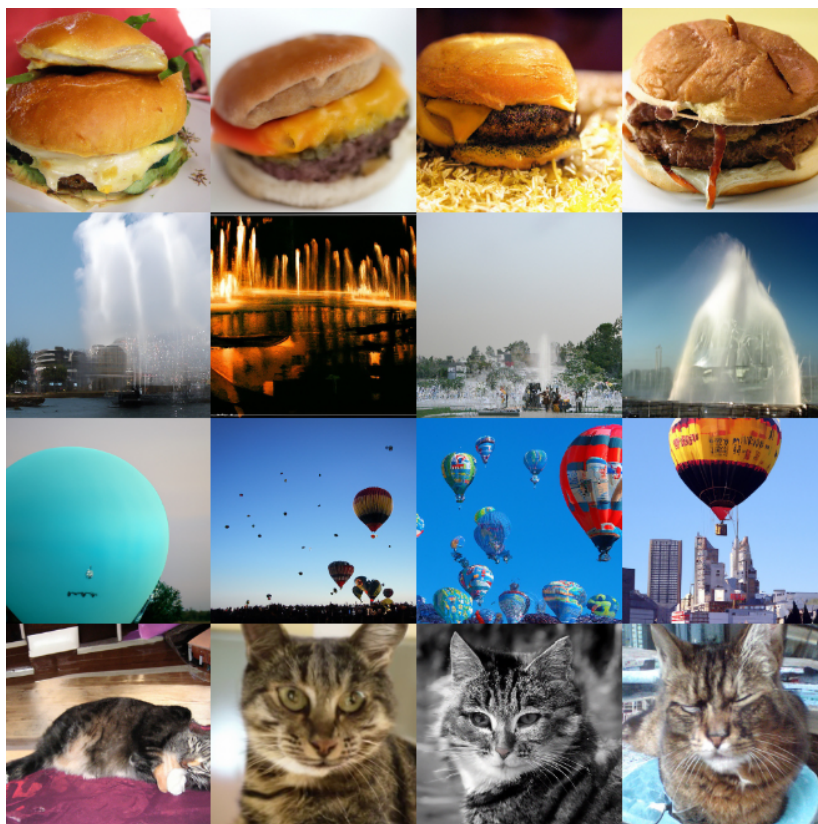
<sup>۱</sup>Classifier Guided Diffusion Model

ورودی در گام زمانی  $t$  ( $\nabla_{x_t} \log f_\phi(y|x_t)$ ) برای هدایت فرایند انتشار معکوس در راستای شرط  $y$  (مثلاً برچسب مورد انتظار)، استفاده می شود. در واقع مقدار نویز پیش‌بینی شده توسط مدل انتشاری در هر گام زمانی با ضربی از گرادیان ذکر شده ترکیب می‌شود به طوری که حاصل نویز نهایی اثری از هدایت دسته‌بند را نیز در خود بگنجاند. به عبارت دقیق تر خواهیم داشت:

$$\epsilon_\theta^*(x_t, t) = \epsilon_\theta(x_t, t) + \alpha \cdot \text{guidance}$$

که  $\text{guidance}$  از گرادیان ذکر شده مشتق می‌شود و  $\alpha$  تعیین کننده نسبت ترکیب تصمیم دسته‌بند و خروجی اصلی مدل انتشاری است که اهمیت هر کدام را در ازای کاهش اهمیت دیگری، تعیین می کند. در شکل ۲-۲۹ نمونه‌هایی از تصاویر تولید شده توسط چنین مدلی را مشاهده می‌کنید.

۲. مدل انتشاری با هدایت بدون دسته‌بند<sup>۱</sup>: هدایت یک مدل انتشاری بدون استفاده از یک دسته‌بند جانبی نیز ممکن است. اگر بتوان به نحوی با استفاده از خود مدل انتشاری و بدون هیچ شبکه اضافی، هر دو توزیع  $p_\theta(x)$  و  $p_\theta(x|y)$  را یادگرفت، آنگاه می‌توان از ترکیب نمونه‌های این دو توزیع، نمونه‌های جدیدی



شکل ۲-۲۹: نمونه‌های تولید شده توسط Classifier Guided Diffusion روی مجموعه داده ImageNet. برچسب‌های استفاده شده به عنوان شرط برای مدل انتشاری در سطر ها به ترتیب از بالا به پایین: چیزبرگر، آنبما، بالن، گربه tabby [۶۴]

<sup>۱</sup>Diffusion Model with Classifier-Free Guidance

تولید کرد که هم به واسطه‌ی  $p_\theta(x)$  دارای واریانس و تنوع کافی هستند و هم به واسطه‌ی  $p_\theta(x|y)$  هدایت شده توسط شرط  $y$  خواهند بود. این ایده در [۶۵] مطرح شده است. برای پیاده‌سازی آن، یک مدل انتشاری توزیع شرطی  $p_\theta(x|y)$  را روی زوج‌های مرتب  $(x, y)$  یاد می‌گیرد، اما به طور تصادفی در برخی از نمونه‌های آموزشی برچسب  $y$ ، drop می‌شود تا مدل توزیع غیرشرطی  $p_\theta(x)$  را نیز به طور ضمنی یاد بگیرد. در نهایت برای تولید یک نمونه مانند حالت هدایت شده با دسته‌بند، ترکیبی از توزیع‌های شرطی و غیر شرطی به عنوان خروجی نهایی ارائه می‌شود. توجه کنید که دلیل استفاده نکردن از توزیع شرطی به تنهایی آن است که خروجی‌های این توزیع معمولاً بسیار شبیه هم هستند چرا که ویژگی‌های کلی متناظر با شرط داده شده در زمان آموزش را یاد گرفته اند که در زمان تست متغیر نیست. بنابراین برای ایجاد تنوع، در هر دو روش هدایت شرطی، به نحوی از ترکیب توزیع بدون شرط و جمله‌ی هدایت کننده استفاده می‌شود. در [۶۶] به صورت تجربی نشان داده شده است که مدل‌های انتشاری هدایت شده بدون دسته‌بند مجزا قوی‌تر از مدل‌های انتشاری هدایت شده با دسته‌بند عمل می‌کنند. در شکل ۲-۳۰ تعدادی از نمونه‌های تولید شده توسط این مدل مشاهده می‌شود.



شکل ۲-۳۰: نمونه‌های تولید شده توسط Classifier-Free Guided Diffusion Model روی مجموعه داده‌ی ImageNet برای سه کلاس گربه tabby، پروانه و چیزبرگر به ترتیب از بالا به پایین [۶۵].

## فصل سوم

### پیشنهاد روشی نوین برای مقابله با حملات تخاصمی

#### ۱-۳ مقدمه

در این فصل ابتدا بیان دقیق مسئله هدف مطرح می‌شود. سپس روش پیشنهادی ما برای حل این مسئله بر مبنای کارهای انجام شده‌ی پیشین به طور دقیق بیان خواهد شد.

#### ۲-۳ بیان مسئله

دسته‌بند تصویر

$$f_{\theta} : R^{H \times W} \rightarrow R^{|C|}$$

موجود است که به عنوان تابع  $f$  با پارامترهای  $\theta$  روی تصاویری با ابعاد  $H \times W$  عمل کرده و این تصاویر را به کلاس متناظرشان نگاشت می‌کند. اکنون حمله‌ی جعبه سفید

$$A : (f, R^{H \times W}) \rightarrow R^{H \times W}$$

را در نظر بگیرید که با دریافت یک دسته‌بند و یک تصویر به عنوان ورودی، یک نمونه‌ی تخاصمی با ابعاد برابر تصویر ورودی تولید می‌کند که خروجی دسته‌بند را دستخوش تغییر می‌کند در حالی که تغییرات نمونه‌ی

تخاصمی نسبت به نمونه‌ی اصلی ناچیز است. به عبارت دقیق‌تر برای تصویر سالم  $x$  داریم:

$$\arg \max_c f_\theta(x) \neq \arg \max_c f_\theta(A(f_\theta, x)), \quad \|x - A(f_\theta, x)\| \leq \epsilon$$

حال هدف ما پیاده‌سازی یک مکانیزم دفاعی با دو هدف تشخیص و پاکسازی - مطابق تعاریف ارائه شده در بخش ۲-۳ - چنین حملاتی است.

### ۳-۳ روش پیشنهادی

روش پیشنهادی ما بر مبنای روش ACGAN-ADA که در [۴۳] ارائه شده است و پیش‌تر در بخش ۲-۳-۲ راجع به آن بحث شد، بنیان‌گذاری می‌شود. در این پژوهش، روش پیشین ACGAN-ADA از چند جهت بهبود داده شده است که در نهایت منجر به چارچوب پیشنهادی ما یعنی آرکین<sup>۱</sup> خواهد شد. در ادامه به بررسی این تغییرات می‌پردازیم.

#### ۱-۳-۳ استفاده از مولد های قوی‌تر

یکی از ایرادات مطرح شده توسط خود محققین ACGAN-ADA، کمبودهای ذاتی معماری ACGAN در مدل‌سازی آماری داده‌های دارای مدهای زیاد است. ایراداتی که در ابتدای بخش ۲-۴-۱-۳ نیز راجع به آن‌ها توضیح داده شد. بنابراین یکی بهبودهای بدیهی روی ACGAN-ADA برای بهتر کردن عملکرد این چارچوب، استفاده از مولدهای بهتر با قابلیت تولید نمونه‌های متنوع حتی در صورت تعداد زیاد کلاس هاست. در این راستا پیشنهاد ما استفاده از یک ReACGAN (بخش ۲-۴-۱-۳) و یک مدل انتشاری شرطی (بخش ۲-۴-۲-۱) به جای مولد چارچوب ACGAN-ADA می‌باشد. تاکید اصلی ما در این پژوهش بر استفاده از مدل‌های شرطی از نظریه‌ی ثابت شده در [۶۷] نشأت می‌گیرد که ادعا می‌کند مدل‌های مشروط از آنجایی که در ازای هر شرط باید احتمالاً توزیع احتمالی ساده‌تر را یاد بگیرند، بنابراین از فرایند آموزش پایدار تری نسبت به مدل‌های بدون شرط برخوردار خواهند بود و به نقطه‌ی بهینه‌ی نزدیک تری به بهینه‌ی سراسری همگرا در فضای پارامترها، همگرا خواهند شد. این امر به مدل کمک می‌کند که علاوه بر یادگرفتن مفاهیم مرتبط با تنوع نمونه های تولید شده، نمونه‌های نزدیک تری به توزیع واقعی نمونه‌ها تولید کند.

<sup>۱</sup>ARCANE



### ۳-۳-۲ روش بهبود یافته برای تشخیص حمله

در روش اصلی مبنای این پژوهش، ACGAN-ADA، برای تشخیص حملات تخاصمی فقط از سه معیار استفاده می‌شود:

$$S_R = D(x)$$

$$S_C = p_D(\hat{c}|x)$$

$$S_g = \min_z \|x - G(z|\hat{c})\|^2$$

که هر کدام به ترتیب در برابر موارد زیر موثر هستند:

۱.  $S_R$ : بزرگ بودن این مقدار می‌تواند به منزله‌ی بزرگ بودن میزان نویز تخاصمی تزریق شده به نمونه‌ی ورودی و مواجهه با یک نمونه‌ی تخاصمی ضعیف از لحاظ بصری تلقی شود. بنابراین انتظار می‌رود که برای نمونه‌های سالم مقدار  $S_R$  کوچک باشد.

۲.  $S_C$ : متناظر با احتمالیست که دسته‌بند جانبی ACGAN به کلاس تشخیص داده شده توسط دسته‌بند مورد حمله، نسبت داده است. اگر این احتمال نسبت به احتمال نظیر دسته‌بند قربانی نباشد، احتمالاً با یک نمونه‌ی تخاصمی مواجه هستیم.

۳.  $S_g$ : اگر مولد بتواند به شرط کلاس تشخیص داده شده توسط دسته‌بند قربانی، نمونه‌ای تولید کند که نرم اختلاف آن با نمونه‌ی ورودی کوچک باشد بنابراین احتمالاً نمونه‌ی ورودی در توزیع آماری شرطی یادگیری شده توسط مولد قرار دارد و سالم است.

در ACGAN-ADA برای تعیین حد آستانه برای هر کدام از معیارهای ارائه شده از روش جستجوی شبکه‌ای<sup>۱</sup> روی آستانه‌ها استفاده شده است. در پژوهش ما، علاوه بر این سه معیار از سه معیار دیگر نیز برای تشخیص حمله بهره برده می‌شود:

۴.  $p_{victim}(\hat{c}|x)$ : سطح اطمینان خود دسته‌بند قربانی از تصمیمی که در رابطه با یک نمونه اتخاذ کرده است می‌تواند خود به عنوان معیاری برای تشخیص حملات استفاده شود. به عنوان مثال قطعیت بسیار بالا برای یک نمونه (مثلاً احتمال 0.998 در یک دسته‌بندی 10 کلاسه) و یا قطعیت نسبتاً پایین (مثلاً 0.3 در همان دسته‌بندی 10 کلاسه) می‌تواند نمایانگر یک حمله‌ی تخاصمی باشد.

<sup>۱</sup> Grid Search

۵.  $JSD(p_D(x)||p_{victim}(x))$ : اختلاف بین احتمالات تولید شده توسط دسته‌بند قربانی و دسته‌بند جانبی یک ACGAN می‌تواند به عنوان معیاری برای تشخیص حمله مورد استفاده قرار بگیرد. برخلاف  $S_C$  که فقط از  $p_D(\hat{c}|x)$  استفاده می‌کند، در این معیار از کل توزیع احتمالی تولید شده استفاده می‌کنیم و برای اندازه‌گیری این اختلاف از دیورژانس (Jensen-Shannon (JS) نقطه‌ای بهره برده می‌شود. این دیورژانس که حالت متقارن دیورژانس KL است برای دو توزیع احتمالی  $P$  و  $Q$  به ترتیب زیر تعریف می‌شود:

$$JSD(P||Q) = \frac{1}{2}D_{KL}(P||M) + \frac{1}{2}D_{KL}(Q||M)$$

که در آن  $D_{KL}$  همان دیورژانس KL بوده و  $M = \frac{1}{2}(P + Q)$ .

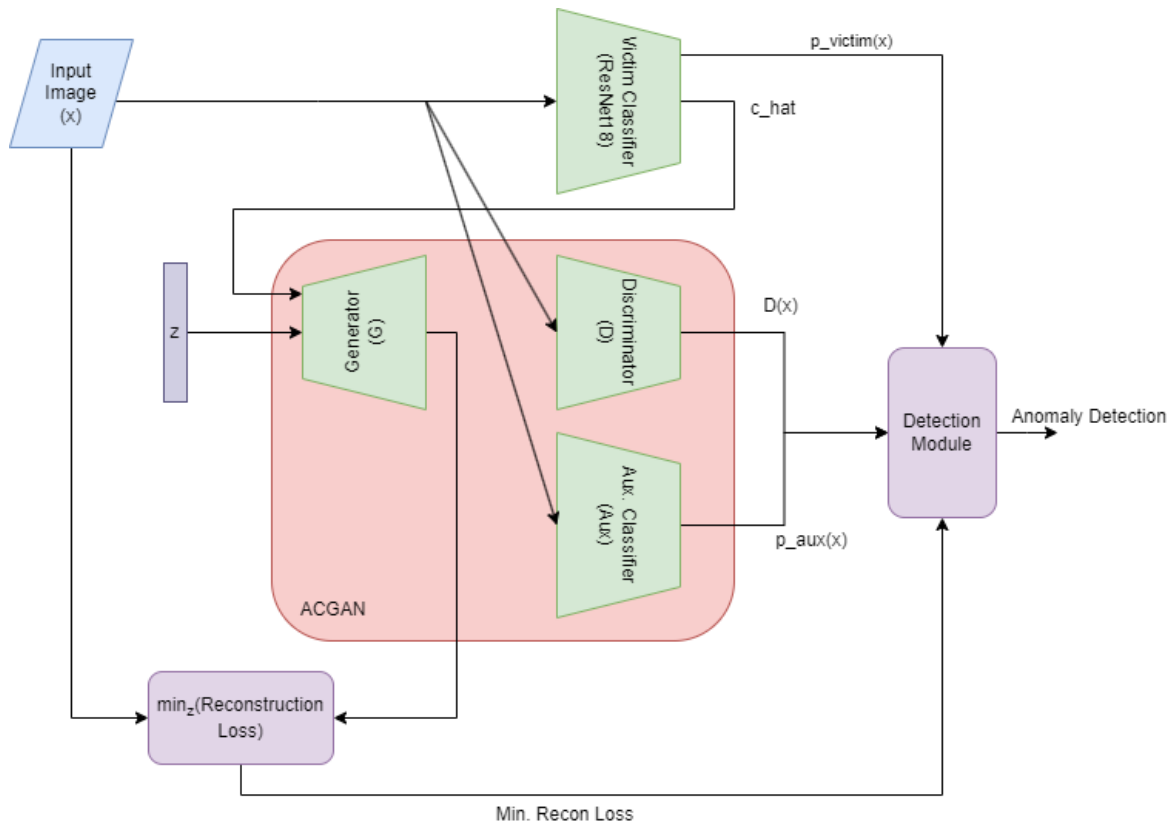
۶.  $\log(p_D(\hat{c}|x)) + \log(D(x))$ : احتمال توأم<sup>۱</sup> دو معیار معرفی شده پیشین می‌تواند به عنوان معیار جدیدی برای تشخیص استفاده شود. از آنجایی که تصمیم اتخاذ شده بر اساس  $S_C$  و  $S_R$ ، تفسیر مشابهی در هر دو معیار دارد، این احتمال توأم می‌تواند معیاری بهتری نسبت به هر کدام از آن‌ها به تنهایی ارائه دهد.

از طرف دیگر، در ARCANE برای تشخیص حمله از جستجوی شبکه‌ای برای پیدا کردن حدود آستانه استفاده نمی‌کنیم. در عوض تمامی این معیارها به صورت داده‌های جدولی<sup>۲</sup> به یک دسته‌بند XGBoost [۶۸] داده می‌شوند و تصمیم‌گیری نهایی توسط این دسته‌بند ثانویه صورت می‌گیرد. دورنمای کلی تشخیص حمله توسط ARCANE-GAN و ARCANE-Diff در شکل‌های ۳-۱، ۳-۲ و ۳-۳ نمایش داده شده است. پس از دریافت تصاویر ورودی، معیارهای ارائه شده محاسبه می‌شوند و در نهایت تمامی این معیارها به هم چسبانده شده و به دسته‌بند XGBoost برای تصمیم‌گیری نهایی داده می‌شوند.

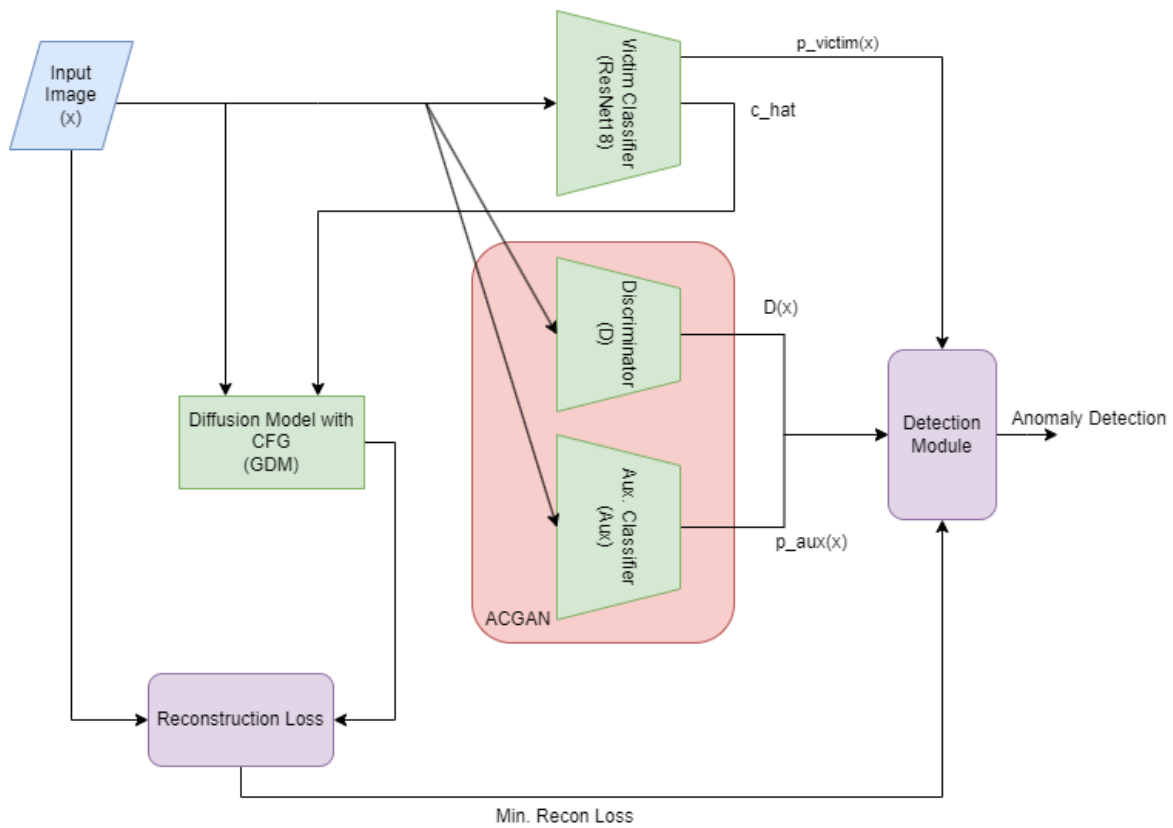
<sup>۱</sup>Joint

<sup>۲</sup>Tabular

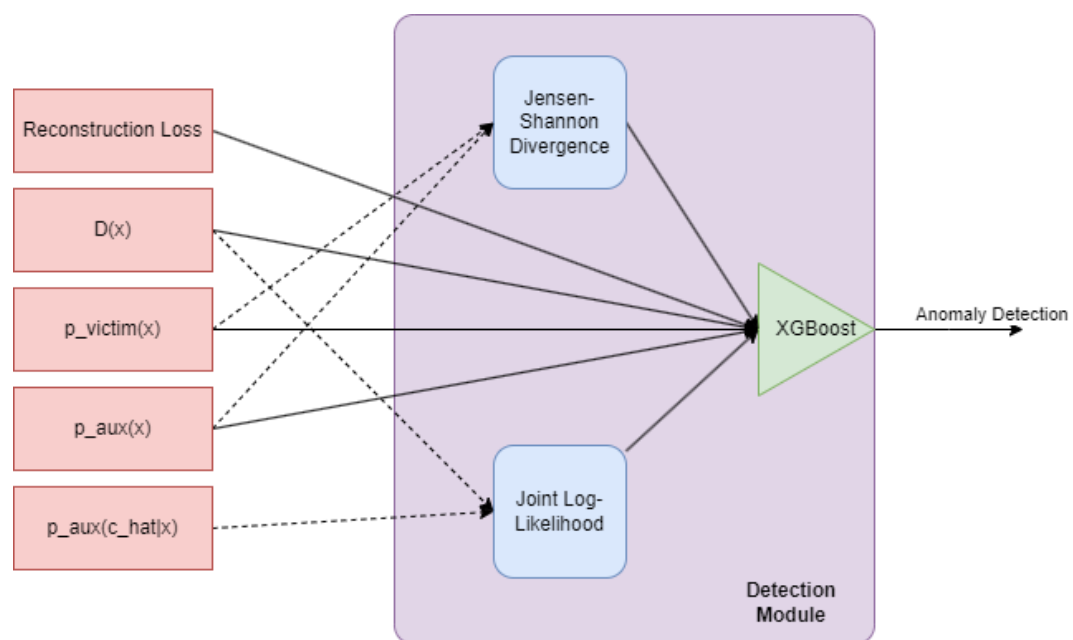




شکل ۳-۱: نحوه‌ی تشخیص حمله توسط ARCANE-GAN



شکل ۳-۲: نحوه‌ی تشخیص حمله توسط ARCANE-Diff



شکل ۳-۳: دورنمای ساختار تصمیم‌گیرنده نهایی مبتنی بر XGBoost

### ۳-۳-۳ روش بهبود یافته برای پاکسازی حمله

فرایند پاکسازی حملات شبیه چارچوب ارائه شده در ACGAN-ADA است با این تفاوت که مولدهای تقویت شده شرطی برای تولید نمونه‌های هدایت شده بهره خواهیم برد. علاوه بر آن پاکسازی نسبت به تمامی برجسب‌ها به صورت همزمان صورت می‌گیرد و در نهایت بین تمام نمونه‌های تولید شده، هر کدام که مقدار نرم اختلافش با نمونه‌ی تخصصی ورودی کمینه بود به عنوان نمونه‌ی پاکسازی شده برای دسته‌بندی نهایی به مدل قربانی داده خواهد شد که کلاس تمیز را تولید خواهد کرد.

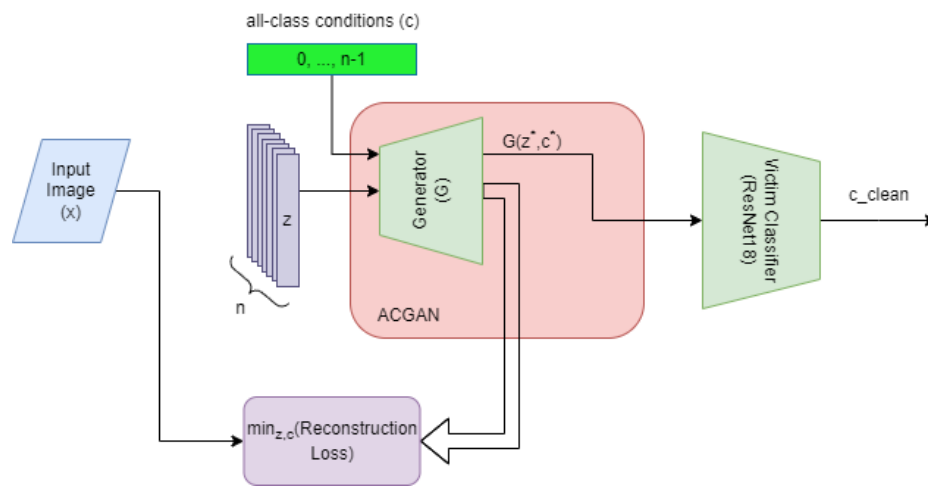
برای تولید تصاویر با استفاده از مولد ReACGAN از ایده‌ای شبیه Defense-GAN [۳۰] ولی به صورت شرطی بهره برده می‌شود. ابتدا یک بردار پنهان  $z$  با ابعاد مورد انتظار مولد تولید می‌شود. سپس  $n$  (تعداد کلاس‌ها) کپی از این بردار به صورت همزمان ولی با کلاس‌های متناظر مختلف به مولد داده می‌شوند. سپس این بردارها طی فرایند نزول گرادیان در راستای کمینه‌سازی نرم اختلاف خروجی مولد و تصویر ورودی، بهینه سازی می‌شوند.

برای تولید تصاویر توسط مدل انتشاری از روش ارائه شده در DiffPure [۳۳] اما به صورت مشروط، و روی تمام کلاس‌ها استفاده می‌کنیم. در واقع هر تصویر ورودی به اندازه  $t \leq T$  کام زمانی مورد تاثیر انتشار پیش‌رو قرار می‌گیرد و نویزی می‌شود. سپس به همان تعداد گام زمانی انتشار معکوس و مشروط بر تمام کلاس‌ها روی  $n$  (تعداد کلاس‌ها) کپی مختلف از تصویر، انجام خواهد شد تا به تصویری شبیه تصویر ورودی برسیم ولی احتمالاً بدون نویز تخصصی برسیم.

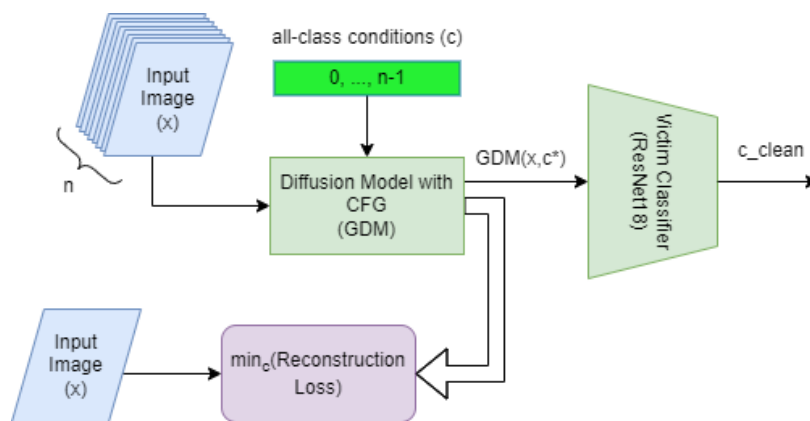
برای حصول اطمینان از تولید تصاویر با کیفیت از لحاظ ترکیب رنگ، فقط در مرحله‌ی پاکسازی، بهترین خروجی‌های هر دو مولد از یک فیلتر تطبیق هیستوگرام<sup>۱</sup> نیز عبور خواهند کرد تا مطمئن شویم ترکیب رنگ خروجی نهایی تا حد ممکن به تصویر ورودی نزدیک است. این امر باعث بهبود دقت پاکسازی می‌شود.

چارچوب کلی پاکسازی ARCANE-GAN و ARCANE-DIFF را در شکل‌های ۳-۴ و ۳-۵ مشاهده می‌کنید.

<sup>۱</sup>Histogram Matching



شکل ۳-۴: نحوه‌ی پاکسازی حمله توسط ARCANE-GAN



شکل ۳-۵: نحوه‌ی پاکسازی حمله توسط ARCANE-Diff

## فصل چهارم

### شبیه سازی و نتایج ارزیابی

#### ۴-۱ مقدمه

در این فصل ابتدا به بررسی نحوه پیاده سازی شبیه سازی و ابزارهای مورد استفاده خواهیم پرداخت. در ادامه نتایج آزمون مدل های ارائه شده ARCANE-GAN و ARCANE-Diff را ارائه کرده و نهایتاً مقایسه ای بین روش کنونی و روش های ارائه شده پیشین مورد بحث قرار خواهد گرفت.

#### ۴-۲ روش شبیه سازی

تمامی مدل ها و آزمایش های صورت گرفته روی بستر کتابخانه یادگیری عمیق معروف PyTorch [۶۹]<sup>۱</sup> و با استفاده از زبان برنامه نویسی پایتون<sup>۲</sup> پیاده سازی شده اند. برای آموزش مدل ReACGAN از مخزن PyTorch-StudioGAN [۷۰]<sup>۳</sup> با اندکی تغییرات استفاده شده است و نهایتاً مدل انتشاری هدایت شده مورد استفاده به کمک مخزن denoising-diffusion-pytorch<sup>۴</sup> پیاده سازی شده است. برای آموزش و آزمون مدل های مولد از مجموعه داده های CIFAR10 [۷۱] و Tiny-ImageNet [۷۲] استفاده شده است.

---

<sup>۱</sup><https://pytorch.org/>

<sup>۲</sup>Python

<sup>۳</sup><https://github.com/POSTECH-CVLab/PyTorch-StudioGAN>

<sup>۴</sup><https://github.com/lucidrains/denoising-diffusion-pytorch>

برای محاسبه‌ی سنجه‌ها از دو کتابخانه‌ی Scikit-learn [۷۳]<sup>۱</sup> و TorchMetrics [۷۴]<sup>۲</sup> استفاده شده است. پیاده‌سازی مدل دسته‌بند نهایی برای تشخیص حمله با استفاده از کتابخانه‌ی معروف XGBoost [۶۸]<sup>۳</sup> انجام شده است. نهایتاً برای پیاده‌سازی حملات تخصصی از کتابخانه‌ی Torchattacks<sup>۴</sup> با کمی تغییرات، بهره‌برداری شده است.

برای آزمون مدل‌ها تعداد ۲۰۰ نمونه از هر مجموعه داده به صورت تصادفی اما متعادل در بین تمام کلاس‌های ممکن انتخاب شدند و سپس از روی این ۲۰۰ نمونه به ازای هر روش حمله یک نمونه‌ی تخصصی ایجاد شده است. به عنوان مثال، برای مجموعه داده‌ی CIFAR10 ۲۰۰ نمونه‌ی سالم، از هر کلاس ۲۰ نمونه، انتخاب شده است و سپس ۲۰۰ نمونه حمله‌ی FGSM و ۲۰۰ نمونه حمله‌ی CW متناظر با این ۲۰۰ نمونه‌ی سالم ایجاد شده و برای آزمون مدل‌ها کنار گذاشته می‌شوند. در زمان آزمون مدل‌ها برای وظایف پاکسازی و تشخیص حملات، به ازای هر نوع حمله، تمام نمونه‌های آن حمله به همراه نمونه‌های سالم به چارچوب مورد آزمایش داده خواهند شد و نتایج روی مجموع ۴۰۰ نمونه (۲۰۰ نمونه سالم و ۲۰۰ نمونه تخصصی) برای هر حمله گزارش شده است. همچنین، برای آموزش تشخیص دهنده‌ی نهایی XGBoost از ۱۰۰۰ نمونه‌ی سالم و تعداد مساوی نمونه‌ی تخصصی از هر نوع حمله استفاده شده است. برای حصول اطمینان از عادلانه بودن نتایج، مطمئن شده ایم که مجموعه داده‌ی مورد استفاده برای آموزش XGBoost اشتراکی با مجموعه داده آزمون مدل‌ها که پیش‌تر توضیح داده شد، ندارد. برای دستیابی به بهترین دقت ممکن روی تشخیص دهنده‌ی XGBoost از روش اعتبار سنجی ضربدری<sup>۵</sup> ۵- لایه با جستجوی شبکه‌ای روی فرآپارامترهای XGBoost صورت گرفته است و سپس مدل‌های دارای بهترین AUC انتخاب شده‌اند.

آموزش و آزمون مدل‌ها روی سیستمی با یک عدد کارت گرافیک NVIDIA GeForce RTX 3090 و ۳۲ گیگابایت رم DDR4 که توسط دانشگاه صنعتی اصفهان در اختیار ما قرار گرفته بود، صورت پذیرفتند.

برای اطلاعات بیشتر راجع به فرآپارامتر<sup>۶</sup> های استفاده شده برای هر مدل، می‌توانید به مخزن ARCANE<sup>۷</sup> مراجعه کنید.

<sup>۱</sup><https://scikit-learn.org/stable/index.html>

<sup>۲</sup><https://lightning.ai/docs/torchmetrics/stable/index.html>

<sup>۳</sup><https://xgboost.readthedocs.io/>

<sup>۴</sup><https://github.com/Harry24k/adversarial-attacks-pytorch>

<sup>۵</sup>Cross-validation

<sup>۶</sup>Hyperparameter

<sup>۷</sup><https://github.com/Adversarian/arcane>

## ۳-۴ نتایج آزمون مدل ها

در این بخش به بررسی نتایج بدست آمده طی آزمون مدل های ارائه شده و چارچوب های ARCANE-GAN و ARCANE-Diff از سه جنبه ی عملکرد مدل های مولد، عملکرد تشخیص و در نهایت عملکرد پاکسازی حملات خواهیم پرداخت.

### ۱-۳-۴ سنجش عملکرد مدل های مولد

هر دو مدل مولد روی split های آموزشی دو مجموعه داده ی CIFAR10 و Tiny-ImageNet آموزش داده شدند. برای انتخاب بهترین مدل های مولد از معیارهای Fréchet Inception Distance (FID) [۷۵] و Inception Score (IS) [۴۲] برای سنجش کیفیت تصاویر تولید شده، استفاده شده است. در ادامه به طور مختصر هر یک را مورد بررسی قرار خواهیم داد.

• **IS:** یکی از اولین معیار قدرتمند برای سنجش عملکرد مدل های مولد. برای محاسبه ی این معیار از خروجی یک دسته بند معروف از پیش آموزش داده شده به نام InceptionV3<sup>۱</sup> روی تعداد نسبتاً زیادی از تصاویر تولید شده توسط مدل مولد، استفاده می شود. مقدار IS در دو صورت بیشینه خواهد شد (حالت مطلوب):

۱. این که آنتروپی برجسب های پیش بینی شده توسط مدل InceptionV3 برای نمونه های تولید شده توسط مدل مولد مورد آزمون، کمینه باشد. به عبارت دقیق تر، دسته بند InceptionV3 می تواند با قطعیت بالایی یک برجسب برای هر نمونه ورودی پیش بینی کند. به طور شهودی، در این حالت می توان گفت نمونه های تولید شده ویژگی های نمونه های واقعی را به خوبی در بر می گیرند، به طوری که به راحتی توسط یک دسته بند نسبتاً قوی، در دسته های متناظرشان قرار می گیرند.

۲. این که برجسب های خروجی تولید شده توسط دسته بند، به طور برابری در همه ی کلاس های ممکن حضور داشته باشند. به عبارت دیگر، آنتروپی برجسب های خروجی دسته بند برای نمونه های تولید شده نسبت به کل دسته های ممکن، بیشینه باشد. به طور شهودی، در این حالت می توان انتظار داشت که نمونه های تولید شده دارای تنوع خوبی هستند.

• **FID:** این معیار فاصله ی فِرشه<sup>۲</sup> بین توزیع بردار ویژگی های استخراج شده توسط دسته بند InceptionV3 از مجموعه داده ی واقعی و نمونه های ساختگی تولید شده توسط یک مدل مولد را محاسبه می کند. از آنجایی که این یک معیار فاصله است، بنابراین مقادیر کمتر برای آن نمایانگر نزدیک تر بودن توزیع مجموعه داده

<sup>۱</sup><https://en.wikipedia.org/wiki/Inceptionv3>

<sup>۲</sup>Fréchet

جدول ۴-۱: عملکرد مدل های مولد روی CIFAR10 و Tiny-ImageNet

CIFAR10			Tiny-ImageNet			
$\uparrow$ IS	$\downarrow$ FID	# Training Steps	$\uparrow$ IS	$\downarrow$ FID	# Training Steps	
10.08	7.28	200000	18.48	15.73	200000	ReACGAN
9.17	3.62	100000	19.15	8.81	300000	GDM
11.54	—	—	34.11	—	—	Real Data

حقیقی و نمونه‌های ساختگیست و بنابراین مطلوب ماست. اگر بتوان توزیع آماری ویژگی های استخراج شده توسط دسته‌بند را برای نمونه‌های حقیقی و ساختگی توسط دو گاوسی چند متغیره مدل کرد، برای محاسبه فاصله‌ی فرشه فرم بسته‌ی زیر موجود است:

$$d_F(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\mu', \Sigma')) = \|\mu - \mu'\|_2^2 + \text{tr}(\Sigma + \Sigma' - 2(\Sigma\Sigma')^{\frac{1}{2}})$$

بنابراین مقدار FID معمولاً زمانی کاهش پیدا می‌کند که نمونه‌های تولید شده توسط مدل مولد - نسبت به مجموعه‌ی داده‌های اصلی - از تنوع خوبی برخوردار باشند.

در عمل، با وجود این که سنجه‌ی FID به عنوان جایگزینی برای IS معرفی شد، این دو سنجه اکنون با هم برای سنجش عملکرد مدل‌ها به کار می‌روند. به عنوان یک قاعده‌ی کلی مدل‌های که دارای FID کمی هستند نمونه‌های متنوع‌تری تولید می‌کنند در حالی که مدل‌هایی که دارای IS بالا هستند، نمونه‌های با کیفیت تری در هر کلاس تولید خواهند کرد. به علاوه، سنجه‌ی FID برای برقراری مقایسه‌ای بین مجموعه داده‌های واقعی و نمونه‌های ساختگی به کار می‌رود در حالی که سنجه‌ی IS یک آنالیز آماری صرف فقط روی نمونه‌های یک مجموعه از داده هاست (معمولاً نمونه‌های ساختگی) [۶۵].

در جدول ۴-۱ عملکرد تولید تصاویر مدل های ReACGAN و Guided Diffusion Model (GDM) را طبق دو معیار IS و FID مشاهده می‌کنید.

#### ۴-۳-۲ سنجش عملکرد تشخیص حملات

یکی از مهم‌ترین سنجه‌هایی که برای تسک‌های تشخیص مورد استفاده قرار می‌گیرد، سطح زیر نمودار<sup>۱</sup> Receiver Operating Characteristic (ROC) است. برای تطبیق نتایج به دست آمده با ACGAN-ADA از سطح زیر نمودار جزئی 0.2 (pAUC-۲.۰) استفاده شده است. علت ذکر شده برای این انتخاب در مقاله‌ی ACGAN-ADA اهمیت بیشتر تشخیص های صورت گرفته در نواحی‌ای از نمودار است که نرخ مثبت کاذب<sup>۲</sup> کم می‌باشد، چرا که نرخ مثبت کاذب کم در عمل برای یک تشخیص دهنده بسیار مطلوب است و نواحی دیگر نمودار ROC

<sup>۱</sup>Area Under Curve

<sup>۲</sup>False Positive

<sup>۳</sup>FPR (False Positive Rate)



در مقام یک سیستم دفاعی، از اهمیت ناچیزی برخوردارند [۴۳]. به عبارت ریاضی اگر AUC به صورت زیر تعریف شده باشد:

$$AUC = \int_{x=0}^1 ROC(x)dx$$

آنگاه  $pAUC-U$  به صورت زیر تعریف خواهد شد:

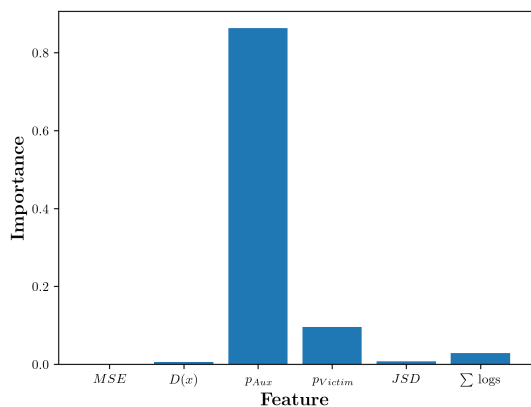
$$pAUC - U = \int_{x=0}^U ROC(x)dx$$

. مشخص است که در این حالت بیشینه مقدار ممکن برای  $pAUC-0.2$  برابر همان 0.2 خواهد بود چرا که در بهترین حالت، این مقدار برابر مساحت مستطیلی با اضلاع 0.2 و 1 خواهد بود که از نقطه‌ی بالا سمت چپ نمودار ROC عبور می کند.

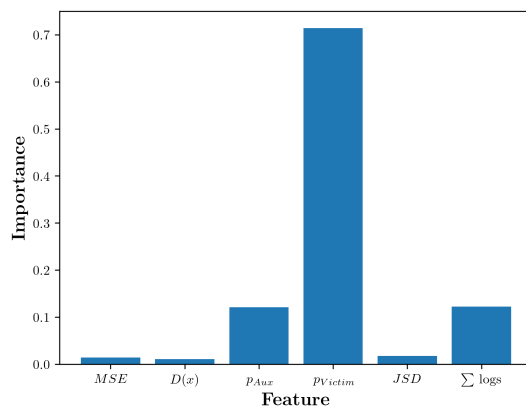
نتایج عملکرد تشخیص ARCANE-Diff و ARCANE-GAN در جدول ۲-۴ آورده شده اند. همچنین برای بررسی تاثیر ویژگی‌های اضافه شده برای تصمیم‌گیرنده XGBoost نهایی، به تفکیک مجموعه‌ی داده‌ها و حملات، اهمیت هر ویژگی در شکل ۴-۱ آمده است. در این شکل مشاهده می شود که با وجود برتری ویژگی  $p_{victim}(\hat{c}|x)$  برای حملات FGSM و  $p_{aux}(\hat{c}|x)$  برای حملات CW باقی ویژگی‌ها نیز در اکثر موارد در تصمیم نهایی XGBoost بی تاثیر نبوده اند.

جدول ۲-۴: عملکرد تشخیص ARCANE-Diff و ARCANE-GAN طبق معیار  $pAUC-0.2$

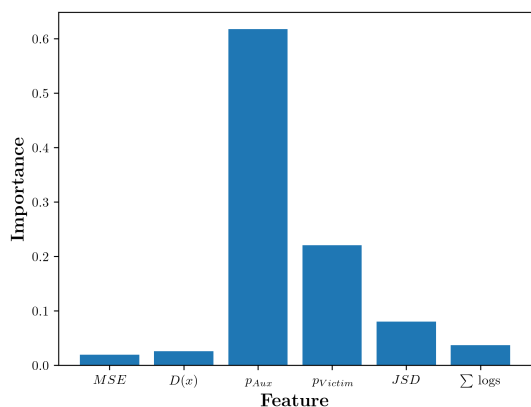
Tiny-ImageNet		CIFAR10		
CW	FGSM	CW	FGSM	
0.19132	0.19856	0.19995	0.18661	ARCANE-GAN
0.1912	0.19859	0.19995	0.18564	ARCANE-Diff



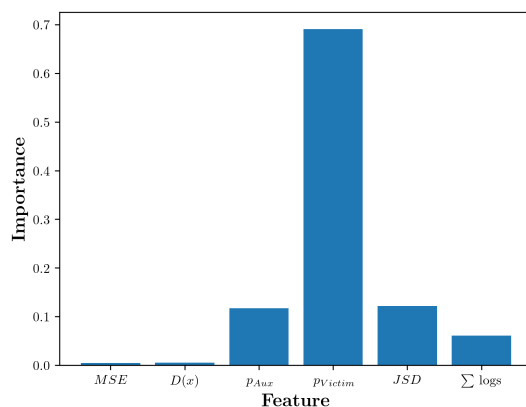
ARCANE-GAN, CIFAR10, CW (ب)



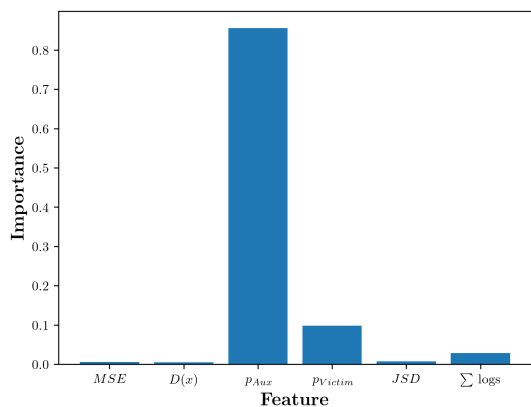
ARCANE-GAN, CIFAR10, FGSM (ب)



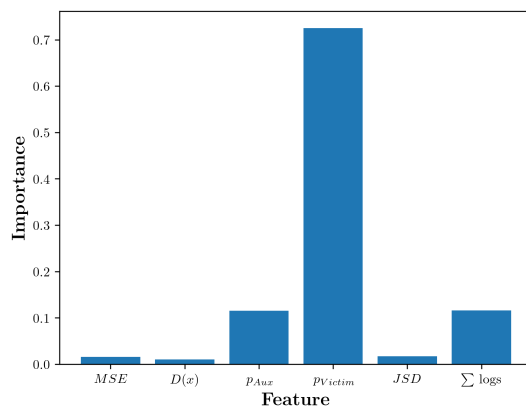
ARCANE-GAN, Tiny-ImageNet, CW (د)



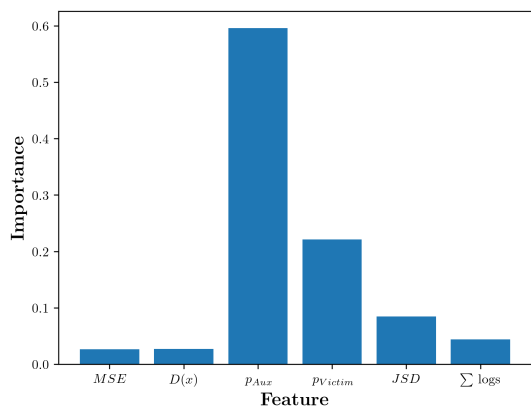
ARCANE-GAN, Tiny-ImageNet, FGSM (ج)



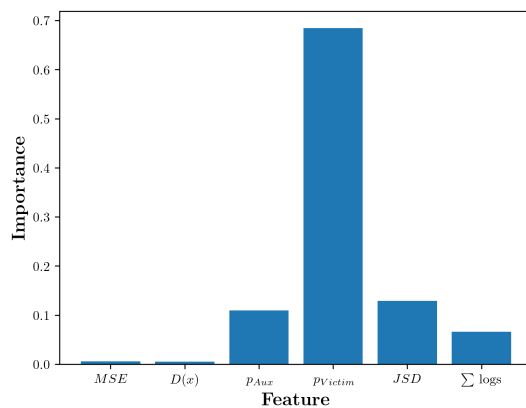
ARCANE-Diff, CIFAR10, CW (و)



ARCANE-Diff, CIFAR10, FGSM (ز)



ARCANE-Diff, Tiny-ImageNet, CW (ح)



ARCANE-Diff, Tiny-ImageNet, FGSM (ز)

شکل ۴-۱: اهمیت ویژگی‌های استفاده شده در XGBoost به تفکیک مجموعه‌ی داده و نوع حمله‌ی صورت گرفته

#### ۴-۳-۳ سنجش عملکرد پاکسازی حملات

برای سنجش عملکرد پاکسازی از دقت پاکسازی استفاده شده است. در دو حالت عملیات پاکسازی موفقیت آمیز تلقی می‌شود:

۱. نمونه سالم بوده و برچسب پیش بینی شده توسط مدل قربانی قبل و پس از پاکسازی برابر است.
  ۲. نمونه تخاصمی بوده و برچسب پیش بینی شده توسط مدل قربانی پس از پاکسازی یا با برچسب پیش بینی شده یا با برچسب واقعی نمونه‌ی سالم متناظر با این نمونه تخصصمی برابر است.
- از آنجایی که مجموعه داده‌های آماده شده برای آزمون عملکرد پاکسازی مدل از تعدادی نمونه‌ی سالم به همراه نمونه‌های تخصصمی متناظرشان به همان تعداد تشکیل شده است، بنابراین، این مجموعه‌ی داده‌ها متعادل بوده و سنجشی دقت در آزمون عملکرد پاکسازی با معنیست.
- در جدول ؟؟ نتایج مربوط به پاکسازی را برای ARCANE-Diff و ARCANE-GAN مشاهده می‌کنید.

#### ۴-۴ بررسی و مقایسه نتایج

## فصل پنجم

### نتیجه‌گیری و پیشنهادها

۱-۵ نتیجه‌گیری  
۲-۵ پیشنهادها

## واژه‌نامه انگلیسی به فارسی

### A

Adversarial Attacks	حملات تخاصمی
Adversarial Perturbation	اختلال تخاصمی
Adversarial Purification	پاکسازی تخاصمی
Adversarial Sample	نمونه تخاصمی
Adversarial Training	آموزش تخاصمی
ARCANE	آرکین
Area Under Curve	سطح زیر نمودار
Attack Surface	رویه‌ی حمله
Attacker	مهاجم
Auto Trader	تاجر خودکار
Autoencoder	خودرمزگذار
Auxiliary Classifier	دسته‌بند اضافی

**B**

Backpropagation	پس انتشار
Binary Search	جستجوی دودویی
Black Box	جعبه سیاه
Box Constraints	محدودیت های جعبه ای

**C**

Cascade	آبشاری
Class Conditional	مشروط بر کلاس
Classifier Guided Diffusion Model	مدل انتشاری هدایت شده با دسته بند
Classifiers	دسته بند ها
Confidence	اطمینان
Confidence Reduction	تقلیل اطمینان
Cyber Attacks	حملات سایبری

**D**

Deep Learning	یادگیری عمیق
Defensive Distillation	تقطیر دفاعی
Denoising Diffusion Probabilistic Model (DDPM)	مدل های انتشار احتمالی حذف کننده نویز
Detection	تشخیص
Diffusion Model	مدل انتشاری
Diffusion Model with Classifier-Free Guidance	مدل انتشاری با هدایت بدون دسته بند
Discriminator	ممیز
$\text{Distortion}(x, \hat{x}) = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2$ where $x, \hat{x} \in \mathbb{R}^n$	اعوجاج
Divergence	دیورژانس

**E**

Evasion Attack	حمله گریزانه
----------------	--------------

Exploratory Attack ..... حمله اکتشافی

## F

False Positive ..... مثبت کاذب

Feature Matching ..... تطبیق ویژگی

Feature Space ..... فضای ویژگی

Forward Diffusion ..... انتشار پیش رو

Fréchet ..... فرِشه

Fully Connected ..... کاملاً متصل

## G

Generative Adversarial Network (GAN) ..... شبکه مولد تخاصمی

Generative AI ..... هوش مولد

Generative Models ..... مدل های مولد

Generator ..... مولد

Gradient Descent ..... نزول گرادیان

Grid Search ..... جستجوی شبکه‌ای

## H

Hidden Layers ..... لایه‌های پنهان

Histogram Matching ..... تطبیق هیستوگرام

Hyperparameter ..... فرایارامتر

## J

Joint ..... توأم

## K

Kernel Density Estimation ..... تخمین چگالی هسته‌ای

**L**

Latent	نهفته
Learning Rate	نرخ یادگیری
Loss	ضرر

**M**

Mahalanobis Distance	فاصله ماهالانوبیس
Manifold	خمینه
Markov Chain	زنجیره مارکوف
Maximum Likelihood Estimation	تخمین بیشینه درست‌نمایی
Metric	سنجه
Min-max	کمینه-بیشینه
Mode Collapse	فروپاشی مُد

**N**

Nash Equilibrium	تبادل نَش
Noise	نویز
Non-equilibrium Thermodynamics	ترمودینامیک غیرتعادلی
Norm	نرم

**P**

Pipeline	خط لوله
Pixel	پیکسل
Poisoning Attack	حمله مسموم‌کننده
Posterior	پسین
Prior	پیشین
Projected Gradient Descent	نزول گرادیان افکنده
Purification	پاکسازی



پایتون ..... Python

## R

آغاز تصادفی ..... Random Initialization

خطای بازسازی ..... Reconstruction Error

مرجع ..... Reference

بهبود ..... Reformer

منظم سازی ..... Regularization

انتشار معکوس ..... Reverse Diffusion

## S

ماشین خودران ..... Self-driving Car

تحلیل احساسات ..... Sentiment Analysis

متغیر لنگی ..... Slack Variable

## T

جدولی ..... Tabular

دسته بندی اشتباه هدفمند ..... Targeted Misclassification

تنتور ..... Tensor

مدل های تهدید ..... Threat Models

تلورانس ..... Tolerance

کانولوشن ترانهاد ..... Transposed Convolution

## U

یادگیری بدون نظارت ..... Unsupervised Learning

دسته بندی اشتباه غیر هدفمند ..... Untargeted Misclassification

## V

گرادیان محو شونده ..... Vanishing Gradient

Victim ..... قربانی

## W

White Box ..... جعبه سفید

## Z

Zero-sum ..... مجموع-صفر

## واژه‌نامه فارسی به انگلیسی

۱

Cascade .....	آبشاری
ARCANE .....	آرکین
Random Initialization .....	آغاز تصادفی
Adversarial Training .....	آموزش تخاصمی
Adversarial Perturbation .....	اختلال تخاصمی
Confidence .....	اطمینان
Distortion( $x, \hat{x}$ ) = $\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2$ where $x, \hat{x} \in \mathbb{R}^n$ .....	اعوجاج
Forward Diffusion .....	انتشار پیش‌رو
Reverse Diffusion .....	انتشار معکوس

ب

Reformer .....	به‌ساز
----------------	--------

پ

Purification .....	پاکسازی
--------------------	---------

Adversarial Purification .....	پاکسازی تخصصی
Python .....	پایتون
Backpropagation .....	پس انتشار
Posterior .....	پسین
Prior .....	پیشین
Pixel .....	پیکسل

## ت

Auto Trader .....	تاجر خودکار
Sentiment Analysis .....	تحلیل احساسات
Maximum Likelihood Estimation .....	تخمین بیشینه درست‌نمایی
Kernel Density Estimation .....	تخمین چگالی هسته‌ای
Non-equilibrium Thermodynamics .....	ترمودینامیک غیر تعادلی
Detection .....	تشخیص
Feature Matching .....	تطبیق ویژگی
Histogram Matching .....	تطبیق هیستوگرام
Nash Equilibrium .....	تعادل نش
Defensive Distillation .....	تقطیر دفاعی
Confidence Reduction .....	تقلیل اطمینان
Tolerance .....	تلورانس
Tensor .....	تَنسور
Joint .....	توأم

## ج

Tabular .....	جدولی
Binary Search .....	جستجوی دودویی
Grid Search .....	جستجوی شبکه‌ای
White Box .....	جعبه سفید

جعبه سیاه ..... Black Box

## ح

حملات تخصصی ..... Adversarial Attacks

حملات سایبری ..... Cyber Attacks

حمله اکتشافی ..... Exploratory Attack

حمله گریزانه ..... Evasion Attack

حمله مسموم‌کننده ..... Poisoning Attack

## خ

خط لوله ..... Pipeline

خطای بازسازی ..... Reconstruction Error

خمینه ..... Manifold

خودرمزگذار ..... Autoencoder

## د

دسته بند ها ..... Classifiers

دسته‌بند اضافی ..... Auxiliary Classifier

دسته‌بندی اشتباه غیر هدفمند ..... Untargeted Misclassification

دسته‌بندی اشتباه هدفمند ..... Targeted Misclassification

دیورژانس ..... Divergence

## ر

رویه‌ی حمله ..... Attack Surface

## ز

زنجیره مارکوف ..... Markov Chain

## س

سطح زیر نمودار ..... Area Under Curve

سنجه ..... Metric

## ش

شبکه مولد تخصصی ..... Generative Adversarial Network (GAN)

## ض

ضرر ..... Loss

## ف

فاصله ماها لانویس ..... Mahalanobis Distance

فراپارامتر ..... Hyperparameter

فرشه ..... Fréchet

فروپاشی مُد ..... Mode Collapse

فضای ویژگی ..... Feature Space

## ق

قربانی ..... Victim

## ک

کاملاً متصل ..... Fully Connected

کانولوشن ترانهاده ..... Transposed Convolution

کمینه-بیشینه ..... Min-max

## گ

گرادیان محو شونده ..... Vanishing Gradient

## ل

لایه‌های پنهان ..... Hidden Layers

## م

Self-driving Car	ماشین خودران
Slack Variable	متغیر لنگی
False Positive	مثبت کاذب
Zero-sum	مجموع-صفر
Box Constraints	محدودیت های جعبه‌ای
Diffusion Model	مدل انتشاری
Diffusion Model with Classifier-Free Guidance	مدل انتشاری با هدایت بدون دسته‌بند
Classifier Guided Diffusion Model	مدل انتشاری هدایت شده با دسته‌بند
Generative Models	مدل های مولد
Denoising Diffusion Probabilistic Model (DDPM)	مدل های انتشار احتمالی حذف کننده نویز
Threat Models	مدل های تهدید
Reference	مرجع
Class Conditional	مشروط بر کلاس
Discriminator	ممیز
Regularization	منظم سازی
Generator	مولد
Attacker	مهاجم

## ن

Learning Rate	نرخ یادگیری
Norm	نرم
Gradient Descent	نزول گرادیان
Projected Gradient Descent	نزول گرادیان افکنده
Adversarial Sample	نمونه تخاصمی
Noise	نویز
Latent	نهفته

Generative AI ..... هوش مولد

Unsupervised Learning ..... یادگیری بدون نظارت

Deep Learning ..... یادگیری عمیق





## مراجع

- [1] Wu, H., Yunas, S., Rowlands, S., Ruan, W., and Wahlstrom, J., “Adversarial Driving: Attacking End-to-End Autonomous Driving”, in *2023 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1–7, June 2023.
- [2] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., and Swami, A., “Practical Black-Box Attacks against Machine Learning”, Mar. 2017.
- [3] Nehemya, E., Mathov, Y., Shabtai, A., and Elovici, Y., “Taking Over the Stock Market: Adversarial Perturbations Against Algorithmic Traders”, Sept. 2021.
- [4] Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., and Mukhopadhyay, D., “Adversarial Attacks and Defences: A Survey”, Sept. 2018.
- [5] Costa, J. C., Roxo, T., Proença, H., and Inácio, P. R. M., “How Deep Learning Sees the World: A Survey on Adversarial Attacks & Defenses”, May 2023.
- [6] Khamaiseh, S. Y., Bagagem, D., Al-Alaj, A., Mancino, M., and Alomari, H. W., “Adversarial Deep Learning: A Survey on Adversarial Attacks and Defense Mechanisms on Image Classification”, *IEEE Access*, Vol. 10, pp. 102266–102291, 2022.
- [7] Ren, K., Zheng, T., Qin, Z., and Liu, X., “Adversarial Attacks and Defenses in Deep Learning”, *Engineering*, Vol. 6, pp. 346–360, Mar. 2020.
- [8] Sun, L., Tan, M., and Zhou, Z., “A survey of practical adversarial example attacks”, *Cybersecurity*, Vol. 1, p. 9, Dec. 2018.
- [9] Goodfellow, I. J., Shlens, J., and Szegedy, C., “Explaining and Harnessing Adversarial Examples”, Mar. 2015.
- [10] Li, Y., Cheng, M., Hsieh, C.-J., and Lee, T. C. M., “A Review of Adversarial Attack and Defense for Classification Methods”, *The American Statistician*, Vol. 76, pp. 329–345, Oct. 2022.
- [11] Qiu, H., Custode, L. L., and Iacca, G., “Black-box adversarial attacks using Evolution Strategies”, in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 1827–1833, July 2021.
- [12] Zhou, M., Gao, X., Wu, J., Liu, K., Sun, H., and Li, L., “Investigating White-Box Attacks for On-Device Models”, Mar. 2024.

- [13] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R., “Intriguing properties of neural networks”, Feb. 2014.
- [14] Carlini, N. and Wagner, D., “Towards Evaluating the Robustness of Neural Networks”, Mar. 2017.
- [15] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A., “Towards Deep Learning Models Resistant to Adversarial Attacks”, Sept. 2019.
- [16] Kingma, D. P. and Ba, J., “Adam: A Method for Stochastic Optimization”, Jan. 2017.
- [17] Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., and Goldstein, T., “Adversarial Training for Free!”, Nov. 2019.
- [18] Zhang, D., Zhang, T., Lu, Y., Zhu, Z., and Dong, B., “You Only Propagate Once: Accelerating Adversarial Training via Maximal Principle”, Nov. 2019.
- [19] Zhang, J., Zhu, J., Niu, G., Han, B., Sugiyama, M., and Kankanhalli, M., “Geometry-aware Instance-reweighted Adversarial Training”, May 2021.
- [20] Wong, E., Rice, L., and Kolter, J. Z., “Fast is better than free: Revisiting adversarial training”, Jan. 2020.
- [21] Gu, S. and Rigazio, L., “Towards Deep Neural Network Architectures Robust to Adversarial Examples”, Apr. 2015.
- [22] Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A., “Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks”, Mar. 2016.
- [23] Zi, B., Zhao, S., Ma, X., and Jiang, Y.-G., “Revisiting Adversarial Robustness Distillation: Robust Soft Labels Make Student Better”, Aug. 2021.
- [24] Cui, J., Liu, S., Wang, L., and Jia, J., “Learnable Boundary Guided Adversarial Training”, Aug. 2021.
- [25] Chen, E.-C. and Lee, C.-R., “LTD: Low Temperature Distillation for Robust Adversarial Training”, June 2023.
- [26] Ross, A. and Doshi-Velez, F., “Improving the Adversarial Robustness and Interpretability of Deep Neural Networks by Regularizing Their Input Gradients”, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32, Apr. 2018.
- [27] Gao, J., Wang, B., Lin, Z., Xu, W., and Qi, Y., “DeepCloak: Masking Deep Neural Network Models for Robustness Against Adversarial Samples”, Apr. 2017.
- [28] Liu, X., Cheng, M., Zhang, H., and Hsieh, C.-J., “Towards Robust Neural Networks via Random Self-ensemble”, July 2018.
- [29] Meng, D. and Chen, H., “MagNet: A Two-Pronged Defense against Adversarial Examples”, in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, (Dallas Texas USA), pp. 135–147, ACM, Oct. 2017.
- [30] Samangouei, P., Kabkab, M., and Chellappa, R., “Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models”, May 2018.
- [31] Jin, G., Shen, S., Zhang, D., Dai, F., and Zhang, Y., “APE-GAN: Adversarial Perturbation Elimination with GAN”, in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Brighton, United Kingdom), pp. 3842–3846, IEEE, May 2019.
- [32] Li, Y., Min, M. R., Lee, T., Yu, W., Kruus, E., Wang, W., and Hsieh, C.-J., “Towards Robustness of Deep Neural Networks via Regularization”, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, (Montreal, QC, Canada), pp. 7476–7485, IEEE, Oct. 2021.
- [33] Nie, W., Guo, B., Huang, Y., Xiao, C., Vahdat, A., and Anandkumar, A., “Diffusion Models for Adversarial Purification”, May 2023.
- [34] Gong, Z., Wang, W., and Ku, W.-S., “Adversarial and Clean Data Are Not Twins”, Apr. 2017.

- [35] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A., “The Limitations of Deep Learning in Adversarial Settings”, Nov. 2015.
- [36] Metzen, J. H., Genewein, T., Fischer, V., and Bischoff, B., “On Detecting Adversarial Perturbations”, Feb. 2017.
- [37] Li, X. and Li, F., “Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics”, Oct. 2017.
- [38] Saberian, M. J. and Vasconcelos, N., “Boosting Classifier Cascades”, *Advances in Neural Information Processing Systems*, Vol. 23, pp. 2047–2055, 2010.
- [39] Zheng, Z. and Hong, P., “Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks”, in *Advances in Neural Information Processing Systems* (Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., eds.), Vol. 31, Curran Associates, Inc., 2018.
- [40] Yang, P., Chen, J., Hsieh, C.-J., Wang, J.-L., and Jordan, M. I., “ML-LOO: Detecting Adversarial Examples with Feature Attribution”, June 2019.
- [41] Schlegl, T., Seeböck, P., Waldstein, S. M., Langs, G., and Schmidt-Erfurth, U., “F-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks”, *Medical Image Analysis*, Vol. 54, pp. 30–44, May 2019.
- [42] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X., “Improved Techniques for Training GANs”, June 2016.
- [43] Wang, H., Miller, D. J., and Kesidis, G., “Anomaly Detection of Adversarial Examples using Class-conditional Generative Adversarial Networks”, May 2022.
- [44] Odena, A., Olah, C., and Shlens, J., “Conditional Image Synthesis With Auxiliary Classifier GANs”, July 2017.
- [45] Feinman, R., Curtin, R. R., Shintre, S., and Gardner, A. B., “Detecting Adversarial Samples from Artifacts”, Nov. 2017.
- [46] Miller, D. J., Wang, Y., and Kesidis, G., “When Not to Classify: Anomaly Detection of Attacks (ADA) on DNN Classifiers at Test Time”, June 2018.
- [47] Kullback, S. and Leibler, R. A., “On Information and Sufficiency”, *The Annals of Mathematical Statistics*, Vol. 22, pp. 79–86, Mar. 1951.
- [48] Lee, K., Lee, K., Lee, H., and Shin, J., “A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks”, Oct. 2018.
- [49] Roth, K., Kilcher, Y., and Hofmann, T., “The Odds are Odd: A Statistical Test for Detecting Adversarial Examples”, May 2019.
- [50] Tian, J., Zhou, J., Li, Y., and Duan, J., “Detecting Adversarial Examples from Sensitivity Inconsistency of Spatial-Transform Domain”, Mar. 2021.
- [51] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., “Generative Adversarial Networks”, June 2014.
- [52] Saad, M. M., O’Reilly, R., and Rehmani, M. H., “A survey on training challenges in generative adversarial networks for biomedical image analysis”, *Artificial Intelligence Review*, Vol. 57, p. 19, Jan. 2024.
- [53] Saxena, D. and Cao, J., “Generative Adversarial Networks (GANs): Challenges, Solutions, and Future Directions”, *ACM Computing Surveys*, Vol. 54, pp. 1–42, Apr. 2022.
- [54] Mohebbi Moghaddam, M., Boroomand, B., Jalali, M., Zareian, A., Daeijavad, A., Manshaei, M. H., and Krunz, M., “Games of GANs: Game-theoretical models for generative adversarial networks”, *Artificial Intelligence Review*, Vol. 56, pp. 9771–9807, Sept. 2023.
- [55] Mounjid, O. and Guo, X., “Convergence of GANs Training: A Game and Stochastic Control Methodology”, Dec. 2021.
- [56] Mirza, M. and Osindero, S., “Conditional Generative Adversarial Nets”, Nov. 2014.

- [57] Kang, M., Shim, W., Cho, M., and Park, J., “Rebooting ACGAN: Auxiliary Classifier GANs with Stable Training”, Nov. 2021.
- [58] Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S., “Deep Unsupervised Learning using Nonequilibrium Thermodynamics”, Nov. 2015.
- [59] Ho, J., Jain, A., and Abbeel, P., “Denoising Diffusion Probabilistic Models”, Dec. 2020.
- [60] Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B., and Yang, M.-H., “Diffusion Models: A Comprehensive Survey of Methods and Applications”, Feb. 2024.
- [61] Croitoru, F.-A., Hondru, V., Ionescu, R. T., and Shah, M., “Diffusion Models in Vision: A Survey”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 45, pp. 10850–10869, Sept. 2023.
- [62] Ronneberger, O., Fischer, P., and Brox, T., “U-Net: Convolutional Networks for Biomedical Image Segmentation”, May 2015.
- [63] Song, J., Meng, C., and Ermon, S., “Denoising Diffusion Implicit Models”, Oct. 2022.
- [64] Dhariwal, P. and Nichol, A., “Diffusion Models Beat GANs on Image Synthesis”, June 2021.
- [65] Ho, J. and Salimans, T., “Classifier-Free Diffusion Guidance”, July 2022.
- [66] Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M., “GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models”, Mar. 2022.
- [67] Bao, F., Li, C., Sun, J., and Zhu, J., “Why Are Conditional Generative Models Better Than Unconditional Ones?”, Dec. 2022.
- [68] Chen, T. and Guestrin, C., “XGBoost: A Scalable Tree Boosting System”, June 2016.
- [69] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S., “PyTorch: An imperative style, high-performance deep learning library”, in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA: Curran Associates Inc., 2019.
- [70] Kang, M., Shin, J., and Park, J., “StudioGAN: A taxonomy and benchmark of gans for image synthesis”, *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2023.
- [71] Krizhevsky, A., “Learning multiple layers of features from tiny images”, *University of Toronto*, May 2012.
- [72] Le, Y. and Yang, X., “Tiny ImageNet Visual Recognition Challenge”, *Stanford University*, 2017.
- [73] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E., “Scikit-learn: Machine learning in Python”, *Journal of Machine Learning Research*, Vol. 12, pp. 2825–2830, 2011.
- [74] Nicki Skafte Detlefsen, Jiri Borgev, Justus Schock, Ananya Harsh, Teddy Koker, Luca Di Liello, Daniel Stancl, Changsheng Quan, Maxim Grechkin, and William Falcon, “TorchMetrics - measuring reproducibility in PyTorch”, Feb. 2022.
- [75] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S., “GANs trained by a two time-scale update rule converge to a local nash equilibrium”, in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS’17, (Red Hook, NY, USA), pp. 6629–6640, Curran Associates Inc., 2017.

# Athena: A Framework for Adversarial Robustness using Class-conditional Generative Models

Arian Tashakkor

tashakkor.a@ec.iut.ac.ir

Feb 19, 2024

Department of Electrical and Computer Engineering  
Isfahan University of Technology, Isfahan 84156-83111, Iran

Degree: M.Sc.

Language: Farsi

**Supervisor: Prof. Mohammad Hossein Manshaei (manshaei@ece.iut.ac.ir)**

## Abstract

Lorem ipsum, or lipsum as it is sometimes known, is dummy text used in laying out print, graphic or web designs. Lorem ipsum, or lipsum as it is sometimes known, is dummy text used in laying out print, graphic or web designs. Lorem ipsum, or lipsum as it is sometimes known, is dummy text used in laying out print, graphic or web designs. Lorem ipsum, or lipsum as it is sometimes known, is dummy text used in laying out print, graphic or web designs.

**Key Words:** Deep Learning, Machine Learning, Adversarial Robustness, Adversarial Attacks



**Isfahan University of Technology**

Department of Electrical and Computer Engineering

# **Athena: A Framework for Adversarial Robustness using Class-conditional Generative Models**

A Thesis

Submitted in partial fulfillment of the requirements  
for the degree of Master of Science

**by**

**Arian Tashakkor**

Evaluated and Approved by the Thesis Committee, on Jan 01, 2022

1. Mohammad Hossein Manshaei, Prof. (Supervisor)
2. XYZ, Prof. (Examiner)
3. XYZ, Assist. Prof (Examiner)

Dr. Behzad Nazari, Department Graduate Coordinator

