



لَهُ مَا  
يَرَى



دانشگاه صنعتی اصفهان  
دانشکده مهندسی برق و کامپیوتر

## آرکین: 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 تشكر

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 آرین  
تشکر

تحت عنوان

آرکین: 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 منشی 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، ارزش‌گذاری و سهم بندی خواهد شد. هر گونه بهره برداری از محتوا، نتایج یا اقدام برای تجاری‌سازی دستاوردهای این پایان نامه تنها با مجوز کتبی دانشگاه صنعتی اصفهان امکان‌پذیر است.

تندیم به

پروردگار عزیزم

متحکم ترین پشتوانه و حامیانم که وجودشان مشاهد ارش و انگزیره است...

و خواهرم

مشوقان و همراهان همیشگی...

# فهرست مطالب

<u>صفحه</u>	<u>عنوان</u>
	فهرست مطالب . . . . .
هشت	فهرست شکل‌ها . . . . .
۵	فهرست جداول . . . . .
دوازده	فهرست اختصارات . . . . .
سیزده	چکیده . . . . .
۱	
	<b>فصل اول : مقدمه</b>
۲	۱-۱ اهمیت مسئله . . . . .
۴	۱-۲ ساختار گزارش . . . . .
	<b>فصل دوم : پیشینه پژوهش و مبانی هوش مولد</b>
۵	۱-۲ مقدمه . . . . .
۵	۲-۲ حملات تخصصی . . . . .
۷	۱-۲-۲ دسته‌بندی حملات تخصصی . . . . .
۷	۱-۲-۲-۱ ای حمله . . . . .
۱۲	۱-۲-۲-۲ FGSM . . . . .
۱۳	۱-۲-۲-۲ CW . . . . .
۱۵	۱-۲-۲-۲ PGD . . . . .
۱۶	۳-۲ روش‌های دفاع در برابر حملات تخصصی . . . . .
۱۷	۱-۳-۲ روش‌های پاکسازی حمله . . . . .
۲۶	۲-۳-۲ روش‌های تشخیص حمله . . . . .
۳۵	۴-۲ مختصری در مورد هوش مولد . . . . .
۳۵	۱-۴-۲ شبکه‌های مولد تخصصی . . . . .
۳۹	۱-۴-۲ cGAN . . . . .
۴۱	۱-۴-۲ ACGAN . . . . .
۴۲	۱-۴-۲ ReACGAN . . . . .
۴۳	۲-۴-۲ مدل‌های انتشاری . . . . .
۴۵	۱-۴-۲ ۱-۴-۲ مدل‌های انتشاری هدایت شده (مشروط) . . . . .

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

۴۹	.....	۱-۳ مقدمه
۴۹	.....	۲-۳ بیان مسئله
۵۰	.....	۳-۳ روش پیشنهادی
۵۰	.....	۳-۳-۱ استفاده از مولد های قوی تر
۵۱	.....	۳-۳-۲ روش بهبود یافته برای تشخیص حمله
۵۵	.....	۳-۳-۳ روش بهبود یافته برای پاکسازی حمله

### فصل چهارم: شبیه سازی و نتایج ارزیابی

۵۷	.....	۱-۴ مقدمه
۵۷	.....	۲-۴ روش شبیه سازی
۵۹	.....	۳-۴ نتایج آزمون مدل ها
۵۹	.....	۳-۴-۱ سنجش عملکرد مدل های مولد
۶۰	.....	۳-۴-۲ سنجش عملکرد تشخیص حملات
۶۳	.....	۳-۴-۳ سنجش عملکرد پاکسازی حملات
۶۴	.....	۴-۴ بررسی و مقایسه نتایج

### فصل پنجم: نتیجه گیری و پیشنهادها

۶۶	.....	۱-۵ مقدمه
۶۶	.....	۲-۵ نتیجه گیری
۶۷	.....	۳-۵ پیشنهادها و کارهای آینده
۶۸	.....	واژه نامه انگلیسی به فارسی
۷۴	.....	واژه نامه فارسی به انگلیسی
۸۰	.....	مراجع
۸۵	.....	چکیده انگلیسی

## فهرست شکل‌ها

۱	۱-۱ نمونه حمله به یک سیستم خودران
۳	
۶	۱-۲ حمله تخاصمی به یک مدل تشخیص چهره
۹	۲-۱ حمله تخاصمی به یک متن
۸	۳-۱ نمونه خط لوله یک سیستم ماشین خودران
۱۳	۴-۱ نمونه حمله FGSM
۱۴	۵-۱ نمودار حساسیت موفقیت حمله و اندازه نویز تخاصمی بر حسب $c$ در حمله CW
۱۶	۶-۱ نمونه‌های تخاصمی تولید شده توسط $L_2$ -CW
۲۰	۷-۱ چارچوب نقطیر دفاعی
۲۱	۸-۱ چارچوب DeepCloak
۲۲	۹-۱ چارچوب MagNet
۲۳	۱۰-۱ چارچوب Defense-GAN
۲۴	۱۱-۱ چارچوب APE-GAN
۲۵	۱۲-۱ چارچوب ER-classifier
۲۶	۱۳-۱ چارچوب DiffPure
۲۷	۱۴-۱ معماری تشخیص دهنده Metzen
۲۸	۱۵-۱ دورنمای نحوه آموزش f-AnoGAN
۲۹	۱۶-۱ f-AnoGAN در زمان تست
۳۰	۱۷-۱ عملکرد چارچوب ACGAN-ADA در زمان تست
۳۳	۱۸-۱ تشكیل نمونه‌های تخاصمی در اطراف خمیدگی‌های مرز تصمیم یک دسته‌بند
۳۴	۱۹-۱ عملکرد SID در زمان تست
۳۵	۲۰-۱ دورنمای شبکه‌های مولد تخاصمی
۳۷	۲۱-۱ معماری شبکه DCGAN
۳۸	۲۲-۱ نمونه‌های تولید شده توسط یک GAN معمولی
۳۹	۲۳-۱ معماری cGAN
۴۰	۲۴-۱ نمونه‌های تولید شده توسط cGAN
۴۱	۲۵-۱ معماری ACGAN
۴۲	۲۶-۱ فضای embedding تصاویر در یک ReACGAN

۴۳	.....	زنجیره مارکوف نظریک DDPM	۲۷-۲
۴۵	.....	نمونه‌های تولید شده توسط DDPM	۲۸-۲
۴۶	.....	Classifier Guided Diffusion Model	۲۹-۲
۴۸	.....	Classifier-Free Guided Diffusion Model	۳۰-۲
۵۳	.....	۱-۳ نحوه تشخیص حمله توسط ARCANE-GAN	
۵۳	.....	۲-۳ نحوه تشخیص حمله توسط ARCANE-Diff	
۵۴	.....	۳-۳ دورنمای ساختار تصمیم‌گیرنده‌ی نهایی مبتنی بر XGBoost	
۵۶	.....	۴-۳ نحوه پاکسازی حمله توسط ARCANE-GAN	
۵۶	.....	۵-۳ نحوه پاکسازی حمله توسط ARCANE-Diff	
۶۲	.....	۱-۴ اهمیت ویژگی‌های استفاده شده در XGBoost	
۶۳	.....	۲-۴ نمونه‌هایی از پاکسازی‌های انجام شده توسط ARCANE	

## فهرست جداول

۱۰	۱-۲ مقایسه حملات جعبه سیاه و جعبه سفید
۱۷	۲-۲ مقایسه حملات CW و PGD, FGSM
۶۰	۱-۴ عملکرد مدل های مولد روی CIFAR10 و Tiny-ImageNet
۶۱	۲-۴ عملکرد تشخیص ARCANE-Diff و ARCANE-GAN طبق معیار pAUC-0.2
۶۳	۳-۴ دقت پاکسازی ARCANE-Diff و ARCANE-GAN
۶۴	۴-۴ مقایسه عملکرد پاکسازی ARCANE با سایر روش ها روی مجموعه داده CIFAR10
۶۴	۵-۴ مقایسه عملکرد پاکسازی ARCANE با سایر روش ها روی مجموعه داده Tiny-ImageNet
۶۵	۶-۴ مقایسه نتایج پاکسازی حملات روی مجموعه داده CIFAR10 و حمله CW

## فهرست اختصارات

### A

ARCANE ..... Adversarial Robustness using Class-conditionAl geNerative modEls

### 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  
GMM ..... Gaussian Mixture Model

**I**

IS ..... Inception Score

**J**

JS ..... Jensen-Shannon

**K**

KL ..... Kullback-Leibler

**L**

LPIPS ..... Learned Perceptual Image Patch Similarity

**M**

MSE ..... Mean Squared Error

**O**

ODE ..... Ordinary Differential Equation

**P**

PGD ..... Projected Gradient Descent

**R**

ROC ..... Receiver Operating Characteristic

**S**

SSIM ..... Structural Similarity Index Measure

**W**

WGAN ..... Wasserstein Generative Adversarial Network

**Y**

YOPO ..... You Only Propagate Once

## چکیده

با گسترش روز افزون استفاده از هوش مصنوعی و به خصوص ابزارهای یادگیری عمیق در مصارف روزمره، نیاز مبرمی به دفاع از چنین ابزارهایی در برابر حملات سایبری احتمالی وجود دارد. دسته‌ی مهمی از حملات سایبری روی مدل‌های یادگیری عمیق، حملات تخصصی نام دارند که با اعمال تغییرات نامحسوس به چشم غیر مسلح روی ورودی‌های چنین مدل‌های، خروجی‌های آن‌ها را دستخوش تغییر می‌کنند. در این پژوهش چارچوبی نوین به نام ARCANE برای مقابله با حملات تخصصی با استفاده از مدل‌های مولد مشروط بر کلاس، ارائه خواهد شد. نتایج تجربی نشان می‌دهند که این چارچوب می‌تواند در تشخیص و پاکسازی چنین حملات به ترتیب ۱۶.۶۲٪ و ۱۱.۸٪ بهتر از بهترین نتایج قبلی عمل کند.

**كلمات کلیدی:** ۱- هوش مصنوعی، ۲- یادگیری عمیق، ۳- مدل‌های مولد<sup>۱</sup>، ۴- دسته بند‌ها<sup>۲</sup>، ۵- حملات تخصصی<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 پیش‌رو بسط داده خواهد شد.

### ۱-۱ اهمیت مسئله

امروزه یادگیری عمیق<sup>۱</sup> به عنوان ابزاری قدرتمند 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>1</sup>Deep Learning

<sup>2</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 مدل‌های یادگیری ماشین دچار خطا در پیش‌بینی قیمت آینده یک سهم می‌شوند.

دسته خاصی از حملات سایبری<sup>۲</sup> اعمال پذیر روی دسته‌بند‌های مبتنی بر یادگیری ماشین، حملات تخاصمی هستند که مرکز اصلی این تحقیق می‌باشد. در فصل ۲ راجع به این حملات توضیح داده خواهد شد. در این پژوهش چارچوبی تحت عنوان ARCANE برای دفاع در برابر حملات تخاصمی از دو جنبه‌ی پاکسازی و تشخیص حملات ارائه خواهد شد. این چارچوب نسبت به چارچوب ACGAN-ADA [۴] که پیش‌تر در همین زمینه ارائه شد از چند لحظه تفاوت دارد:

۱. استفاده از مدل‌های مولدهای بمبود یافته: در ARCANE از مدل‌های مولد جدیدتر مانند ReACGAN [۵] و نیز مدل‌های انتشاری (DDPM [۶]) استفاده می‌شود که می‌تواند برخی از ایرادات ACGAN-ADA را مرتفع کند.

۲. استفاده از سنجه‌های جدید برای تشخیص حملات: همانطور که بعداً در بخش ۲-۳-۳ توضیح داده خواهد شد، در ARCANE از سه سنجه‌ی دیگر علاوه بر موارد استفاده شده در ACGAN-ADA برای تشخیص حملات استفاده می‌شود.

۳. استفاده از یک تصمیم‌گیرنده‌ی نهایی برای تشخیص حملات: به جای اتخاذ تصمیم بر اساس حدود آستانه‌ی بدست آمده با روش‌هایی مانند جستجوی شبکه‌ای<sup>۳</sup>، در ARCANE از یک مدل XGBoost برای



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

<sup>1</sup> Auto Trader

<sup>2</sup>Cyber Attacks

<sup>3</sup>Grid Search

اتخاذ تصمیم بر اساس سه سنجه‌ی بدست آمده از نمونه‌های ورودی استفاده می‌شود.

پس از آزمایش‌های صورت گرفته، نشان داده می‌شود که ARCANe می‌تواند هم در تشخیص و هم در پاکسازی، بهترین نتایج پیشین را با اختلاف زیادی شکست دهد و به طور متوسط در تشخیص و پاکسازی به ترتیب 16.62% و 11.8% بهتر از ACGAN-ADA عمل کرده کند.

## ۱-۲ ساختار گزارش

در ادامه این گزارش، در فصل ۲ پیشینه پژوهش و مبانی لازم برای درک بهتر فصل‌های آتی مورد بررسی قرار خواهد 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>۱</sup> های مناسب مورد بررسی قرار خواهد گرفت. در نهایت در فصل ۵ به جمع‌بندی نهایی و جهت‌های احتمالی تحقیقات آینده پرداخته خواهد شد.

## 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 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 [۹-۷]. 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 خواهد شد [۱۰]. تحقیقات انجام پذیرفته روی این دسته از حملات عموماً

<sup>1</sup> Adversarial Perturbation

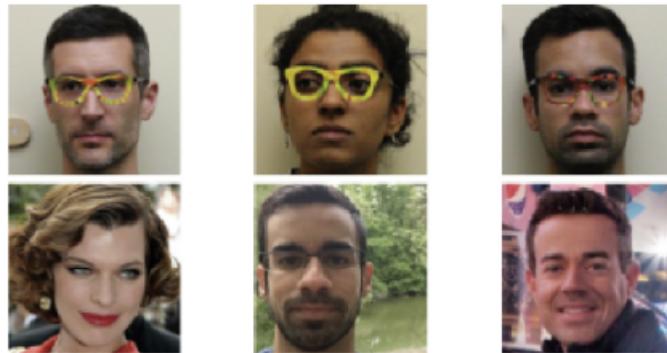
<sup>2</sup>Noise

<sup>3</sup>Victim

در حوزه تصویر می باشد. در این روش ها با تغییرات جزئی و نامحسوس روی مقادیر پیکسل<sup>۱</sup> های به خصوص، خروجی یک دسته بند 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 دسته بند ها، چه در حوزه تصویر و چه خارج از آن، باشند [۱۱، ۸]. در شکل ۱-۲ نمونه ای 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، چهره ای این افراد را به اشتباه به عنوان افراد نظیر در سطر پایین تشخیص داده است. همچنین در شکل ۲-۲ نمونه ای از یک حمله تخاصمی به یک مدل دسته بند 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 [۱۲] بررسی کرد. فرض کنیم که مدل دسته بند ( $f$ ) 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 ۲-۱: نمونه ای حمله تخاصمی به یک مدل تشخیص چهره [۱۰]. 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 **elekies 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 embarrassing1y foolish fo0l1sh** sexual innuendo littered throughout. This movie will make you laugh, but never intentionally. I repeat, never.

شکل ۲-۲: یک مدل دسته بند متن مبتنی بر CNN که با تغییرات جزئی در متن نظر ارسال شده، با وجود حفظ محتوای کلی نظر، دچار اشتباه شده است [۱۰].

<sup>1</sup>Pixel

<sup>2</sup>Sentiment Analysis

اکنون یک نمونه تخاصمی<sup>۱</sup> به ورودی دستکاری شده‌ی  $\hat{x}$  گفته می‌شود که طبق معادلات (۲-۲) و (۳-۲)

از افزودن مقدار ناچیزی  $\delta$  به ورودی اولیه  $x$  به دست می‌آید 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 دسته‌بندی حملات تخاصمی

امنیت یک مدل یادگیری ماشین با توجه به اهداف خصم‌مانه مورد نظر و قابلیت‌های مهاجم<sup>۲</sup> ارزیابی می‌شود [۸، ۷]. پیش از پرداختن به دسته‌بندی حملات تخاصمی، ابتدا راجع به مدل‌های تهدید<sup>۳</sup> در حوزه یادگیری ماشین با توجه به قدرت و دسترسی مهاجم بحث خواهد شد.

### ۱-۱-۲-۲ ی حمله

سطح حمله<sup>۴</sup> عبارتیست که به تمام روش‌های ممکن موجود برای یک مهاجم برای حمله به یک سیستم اطلاق می‌شود. یک سیستم تصمیم‌گیرنده مبتنی بر یادگیری ماشین را می‌توان عملاً به عنوان یک خط لوله<sup>۵</sup> برای پردازش داده‌های ورودی تصور کرد. بدین ترتیب، دنباله‌ای از عملیات ساده روی داده‌های ورودی در زمان استفاده از یک مدل یادگیری ماشین که چنین خط لوله‌ای را تشکیل می‌دهند، می‌توان به صورت زیر خلاصه کرد:

۱. جمع‌آوری داده‌های ورودی از سنسورها و یا انبارهای داده

۲. انتقال داده‌های جمع‌آوری شده در مرحله قبل به دامنه دیجیتال

۳. پردازش داده‌های 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 دنباله‌ای را می‌توان در شکل ۳-۲ مشاهده کرد.

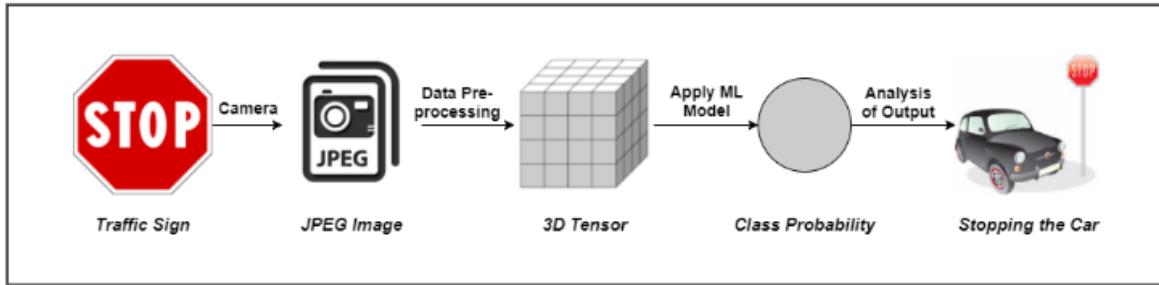
<sup>1</sup> Adversarial Sample

<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]

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 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، سطح 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 سطح حمله مورد استفاده به شرح زیر می باشد [۱۰-۷] :

۱. حمله گریزانه<sup>۲</sup>: این دسته از حملات، معمول ترین دسته ای حملات تخصصی هستند. در این حالت مهاجم در صدد گریزانی از یک سیستم امنیتی به وسیله ای 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 در این حمله ابتدا نمونه های آموزشی جدیدی که با هدف خاص انحراف مدل قربانی طراحی شده اند را به مجموعه داده های آموزشی می افزاید و در این راستا سعی می کند که فرایند آموزش مدل را مختل کرده و یا خروجی آن را پس از اتمام آموزش چار 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.

<sup>1</sup>Tensor

<sup>2</sup>Evasion Attack

<sup>3</sup>Poisoning Attack

۳. حمله اکتشافی<sup>۱</sup>: این حملات که با پیش فرض دسترسی جعبه سیاه<sup>۲</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 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 داده های آموزشی دریافت کند.

برای تعریف مدل تهدید نیازمند آن هستیم که نوع دسترسی مهاجم به قربانی را نیز در نظر بگیریم [۸، ۹]. اگر به مثال ماشین خودران باز گردیم، برای یک مهاجم قوی می توان دسترسی به معماری مدل استفاده شده و مجموعه داده های آموزشی را متصور شد در حالی که یک مهاجم ضعیف تن فقط احتمالاً به داده های زمان تست دسترسی دارد. با وجود این که هر دو مهاجم از یک سطح برای حمله به مدل استفاده می کنند، مهاجم اول به دلیل در اختیار داشتن اطلاعات کامل تر، قوی تر تلقی می شود.

بدین ترتیب، حمله ها را می توان بر اساس گستره دسترسی مهاجم به دو دسته مهم حملات جعبه سیاه و جعبه سفید<sup>۳</sup> دسته بندی کرد [۱۵، ۱۴، ۲].

- حملات جعبه سفید: در این دسته از حملات، قوی ترین دسترسی ممکن برای مهاجم فرض می شود. مهاجم در این نوع حمله از اطلاعات کامل راجع به مدل مورد استفاده برای تصمیم گیری (مثلاً معماری دقیق مدل، تابع هزینه مورد استفاده برای آموزش، گرادیان خروجی مدل نسبت به هر متغیر مطلوب مهاجم، وزن های مدل و غیره) برخوردار است. همچنین مهاجم از نحوه آموزش مدل (مثلاً الگوریتم بهینه سازی مورد استفاده) مطلع بوده و به مجموعه داده های آموزشی مدل دسترسی کامل دارد. با این مفروضات، مهاجم سعی می کند از نقاط ضعف مدل در فضای ویژگی<sup>۴</sup> مطلع شود و از آن های برای تخریب عملکرد مدل سوء استفاده کند.

- حملات جعبه سیاه: در این حملات که در نقطه مقابل حملات جعبه سفید قرار می گیرند، هیچ پیش فرض خاصی برای مهاجم در نظر گرفته نمی شود و نمایانگر یک سناریوی حمله محتمل تر است که در آن مهاجم سعی می کند با در اختیار داشتن اطلاعات محدود راجع به عملکرد مدل و خروجی های دریافتی از ورودی هایی که خودش به مدل ارسال می کند، از نقاط ضعف مدل پرده برداری کند.

در جدول ۱-۲ مقایسه ای 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

<sup>1</sup>Exploratory Attack

<sup>2</sup>Black Box

<sup>3</sup>White Box

<sup>4</sup>Feature Space

یک سیستم تصمیم‌گیرنده مبتنی بر یادگیری ماشین را می‌توان به موارد زیر خلاصه کرد:

۱. **تقلیل اطمینان<sup>۱</sup>:** در این حالت، مهاجم سعی می‌کند سطح اطمینان<sup>۲</sup> خروجی دسته‌بند را برای دسته تشخیص داده شده  $y_{pred}$  کاهش دهد در حالی که خروجی کلاس دسته‌بند دچار تغییر نشود. به عبارت دیگر:

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

۲. **دسته‌بندی اشتباه غیر هدفمند<sup>۳</sup>:** در این حالت، مهاجم در صدد تغییر خروجی مدل به هر کلاس  $\hat{y} \neq y_{pred}$  برمی‌آید.

۳. **دسته‌بندی اشتباه هدفمند<sup>۴</sup>:** در این حالت، مهاجم تلاش می‌کند خروجی مدل را به کلاس خاص مطلوب تغییر دهد.

هدف و تمرکز اصلی این پژوهش روی حملات تخاصمی گریزانه‌ی جعبه سفید است که از گرادیان خروجی

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

حملات جعبه سفید	حملات جعبه سیاه	
به دسته ای از حملات گفته می‌شود که در آن‌ها مهاجم اطلاعات کامل از معماری و ساختار درونی سیستم مورد حمله دارد.	به دسته ای از حملات گفته می‌شود که در آن‌ها ساختار درونی و طراحی سیستم مورد حمله از مهاجم مخفیست.	تعريف
نیازمند اطلاعات محروم‌انه راجع به سیستم مانند وزن‌های مدل مورد استفاده، معماری آن، کتابخانه‌های استفاده شده برای پیاده‌سازی وغیره می‌باشد.	نیازمند اطلاعات بنیادی راجع به سیستم قربانی و نحوه عملکرد آن نیست.	سطح اطلاعات قابل دسترسی
پیاده‌سازی پیچیده‌تر، اما به مراتب قوی‌تر حصول امنیت نسبی در برابر این حملات معمولاً وضعیت آرمانیست.	پیاده‌سازی معمولاً ساده‌تر، اما ضعیف‌تر. فرض واقع گرایانه‌تر راجع به مهاجمین احتمالی. کارآیی بالا در کشف ایرادات رفتاری مدل.	مزایا و معایب
این حملات از خصوصیات درونی مدل مورد استفاده سوء استفاده می‌کنند. به عنوان مثال حملات مبتنی بر گرادیان و یا استفاده از دانش قبلی راجع به ضعف‌های موجود در سیستم با فرض دانش کامل راجع به ساختار داخلی مدل استفاده شده در این دسته از حملات قرار می‌گیرند.	در این حملات معمولاً یک مدل محلی به آموزش داده می‌شود که بتواند رفتار مدل قربانی را تقلید کند. این کار با استفاده از تولید نمونه‌های ورودی ساختگی توسط مهاجم و استفاده از برچسب‌های خروجی مدل قربانی روی همین ورودی‌ها، صورت می‌گیرد.	استراتژی حمله

<sup>1</sup>Confidence Reduction

<sup>2</sup>Confidence

<sup>3</sup>Untargeted Misclassification

<sup>4</sup>Targeted Misclassification

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، 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 [۱۳، ۱۱-۷] ارجاع داده می‌شود.

به طور کلی، فرایند تولید یک نمونه تخاصمی را می‌توان به صورت معادله (۴-۲) فرمولبندی کرد [۱۶]:

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

که در آن  $x$  یک نمونه سالم و  $\mathcal{D}$  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 (۴-۲) 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 <sup>۱</sup> FGSM حاصل یکی از اولین پژوهش‌ها در زمینه حملات تخاصمی بود و امروزه به عنوان یک حمله‌ی سریع، ساده، اما نسبتاً ضعیف در میان حملات تخاصمی جعبه سفید شناخته می‌شود [۱۲]. این حمله معمولاً در حالت غیر هدفمند پیاده سازی می‌شود ولی پیاده سازی آن در حالت هدفمند نیز ممکن است. در مقابل این حمله یکی دیگر از معروف ترین حملات جعبه سفید، حمله <sup>۲</sup> CW است که در سال ۲۰۱۷ طراحی شد [۱۷] و هنوز یکی از قوی‌ترین حملات جعبه سفید شناخته شده با زمان اجرای معقول است که در هر دو نوع هدفمند و غیر هدفمند قابل پیاده‌سازی است. حمله <sup>۳</sup> PGD [۱۸] مستقیماً در این پژوهش برای آزمایش مدل‌ها مورد استفاده قرار نگرفته است ولی از آنجایی که در ادامه مباحث در بخش ۳-۲ به تعریف آن احتیاج است، این حمله نیز که یکی از حملات قوی و شناخته شده جعبه‌سفید در ادبیات حملات تخاصمیست، مورد بررسی قرار خواهد گرفت.

<sup>۱</sup>Fast Gradient Signed Method

<sup>۲</sup>Carlini-Wagner

<sup>۳</sup>Projected Gradient Descent

فرض کنید مدل  $f$  با پارامتر های  $\theta$  را در اختیار داشته باشیم. همچنین فرض کنید که  $(X, y)$  زوج های مرتبی از ورودی ها و خروجی های متناظر به  $f$  باشند و نیز تابع هزینه  $J$ . که مدل به وسیله آن و طی یک فرایند بهینه سازی، آموزش داده شده است. اکنون عبارت (۵-۲) را در نظر بگیرید:

$$\nabla_x J(f(x; \theta), y) \quad (5-2)$$

این عبارت مقدار گرادیان تابع هزینه آموزش مدل  $f$  را نسبت به **ورودی** مدل (و نه پارامتر های مدل،  $\theta$ ) نشان می دهد. بدین ترتیب اگر به هر نحو این گرادیان به ورودی اولیه افروده شود، ورودی تولید شده احتمالاً منجر به زیاد شدن مقدار تابع هزینه نهایی و موفقیت حمله خواهد شد. در [۱۲] نویسندهای از نویز تخاصمی ارائه شده در (۶-۲) استفاده می کنند و نمونه تخاصمی نهایی از معادله (۷-۲) بدست می آید.

$$\delta = \epsilon \cdot \text{sgn}(\nabla_x J(f(x; \theta), y)) \quad (6-2)$$

$$\hat{x} = x + \delta \quad (7-2)$$

که در آن مقدار  $\epsilon$  که کنترل کننده اندازه نویز تخاصمی بوده و بسته به مجموعه داده مورد استفاده و ضریب اطمینان لازم برای موفقیت حمله قابل تنظیم است. نویسندهای 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 اندازه هی برابر بسته به جهت گرادیان تابع هزینه نسبت به  $x$  کم و یا زیاد می شود به طوری که تصویر حاصل نسبت به برچسب 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 برچسب حقیقی و پیاده سازی یک حمله غیر هدفمند است. اگر بخواهیم با استفاده از همین روش یک حمله هدفمند را با برچسب مطلوب  $y_{desired}$  اجرا کنیم، کافیست که مقدار  $\delta$  به شکل معادله (۸-۲) تغییر داده شود:

$$\delta_{targeted} = -\epsilon \cdot \text{sgn}(\nabla_x J(f(x; \theta), \hat{y}_{desired})) \quad (8-2)$$

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% برچسب این تصویر را به درستی تشخیص دهد. اکنون با افزودن 0.007 از نویز تخاصمی تولید شده - که خود با سطح اطمینان 8.7% توسط مدل به عنوان "کرم لوله ای" دسته بندی شده - تصویر نهایی که همچنان به چشم غیر مسلح مانند تصویر اولیه است، با سطح اطمینان 99.3% توسط مدل به کلاس "میمون دست دراز" 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

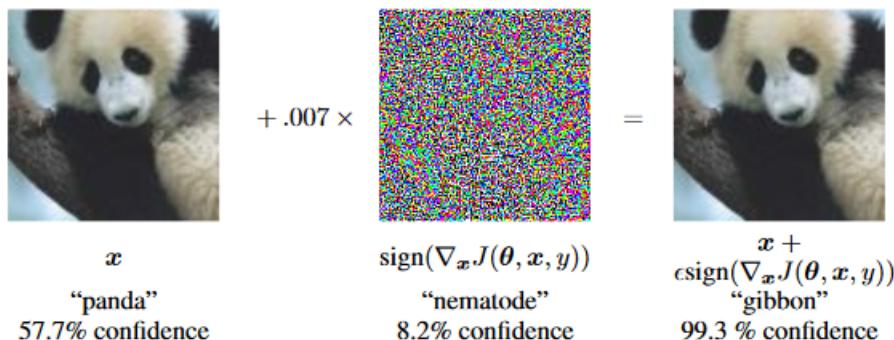
حمله‌ی دیگر مورد استفاده در این پژوهش، حمله‌ی CW است که در [۱۷] ارائه شده است. برای توضیح نحوه‌ی ساختن یک نمونه تخاصمی با این روش به معادله (۴-۲) باز می‌گردیم. در این حمله، از آنجایی که شرط  $t = C(x + \delta)$  بسیار غیرخطی و غیرقابل بهینه‌سازیست، این شرط باید ابتدا به طوری بازنویسی شود که بتوان آن را با استفاده از تکنیک‌های مرسوم بهینه‌سازی، اعمال کرد. برای این کار محققین این پژوهش تابع هدف  $f$  را طوری تعریف می‌کنند که شرط  $t = C(x + \delta)$  برقرار باشد اگر و تنها اگر  $f(x + \delta) \leq 0$ . اکنون به جای بیان مسئله به صورت

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

از فرمول‌بندی معادل (۱۰-۲) استفاده می‌شود:

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

که در آن  $c > 0$  مقداری ثابت است که قدرت حمله انجام شده را در ازای از دست دادن کمینگی  $\mathcal{D}(x, x + \delta)$  تنظیم می‌کند. این دو فرمول‌بندی معادل‌لند چرا که می‌توان نشان داد مقدار  $c > 0$  وجود دارد که راه حل بهینه در



شکل ۲-۴: نمونه حمله FGSM. در این حمله تصویر یک پاندا با افزودن مقدار ناچیزی نویز تخاصمی به عنوان یک میمون دست‌دراز شناخته شده است [۱۲].

معادله (۱۰-۲) با راه حل بهینه (۹-۲) برابر است. در [۱۷] از نرم<sup>۱</sup>

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

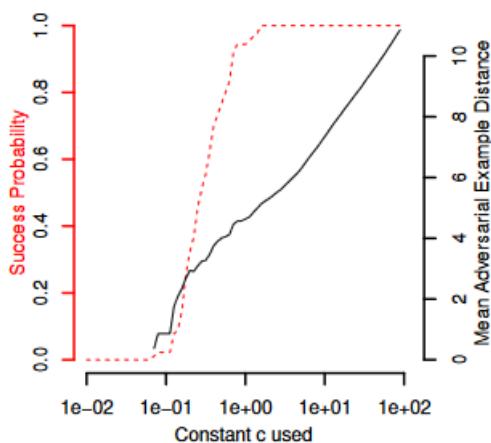
به عنوان معیار فاصله  $\mathcal{D}$  استفاده می شود. و بدین ترتیب معادله کلی نهایی حمله CW به صورت زیر در خواهد آمد:

$$\begin{aligned} & \text{minimize} \quad \|\delta\|_p + c \cdot f(x + \delta) \\ & \text{s.t. } x + \delta \in [0, 1]^n \end{aligned} \quad (11-2)$$

اکنون برای حل این مسئله با استفاده از روش های بهینه سازی تکراری، نیاز به انتخاب  $p$ ،  $c$  و تابع هزینه مناسب  $f$  است.

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

شرطی مانند  $x + \delta \in [0, 1]^n$  به نام محدودیت های جعبه‌ای<sup>۳</sup> معروف هستند. برای اجرا کردن این شرط، در حمله CW از یک تغییر متغیر استفاده می شود. به جای بهینه سازی مستقیم مقدار  $\delta$  در معادله اصلی، متغیر



شکل ۲-۵: نمودار حساسیت موفقیت حمله و اندازه نویز تخاصمی بر حسب مقدار ثابت  $c$  [۱۷]

<sup>1</sup>Norm

<sup>2</sup>Binary Search

<sup>3</sup>Box Constraints

### جديد $w$ معرفی میشود و مقدار

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

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

قوی ترین نوع حمله  $L_2$ -CW است که مسئله بهینه‌سازی آورده شده در رابطه‌ی (۱۲-۲) را حل می کند.

$$\text{minimize} \quad \left\| \frac{1}{2}(\tanh(w) + 1) - x \right\|_2^2 + c \cdot f\left(\frac{1}{2}(\tanh(w) + 1)\right) \quad (12-2)$$

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  $y = F(x) = \text{softmax}(Z(x))$ ) 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 (۱۲-۲) توسط بهینه‌ساز Adam [۱۹] حل می شود و در نهایت نمونه‌های تخاصمی به دست خواهد آمد. نمونه‌هایی از این حمله را می‌توان در شکل ۶-۲ مشاهده کرد. همانطور که می‌توان دید، بدون توجه به این که کلاس هدف دارای چه مقداریست تمامی نمونه‌های تخاصمی 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 [۱۸] 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 تخاصمی بین مهاجم و مدل قربانی بررسی می کنند که تحت عنوان مسئله نقطه‌زنی زیر به عنوان تابع هزینه مدل قربانی بیان می شود:

$$\min_{\theta} \max_{\delta \in S} \mathbb{E}_{(x,y) \sim \mathcal{D}} L(\theta, x + \delta, y) \quad (13-2)$$

که در آن وظیفه مهاجم حل مسئله بیشینه‌سازی درونی و وظیفه قربانی حل مسئله کمینه‌سازی برونویست. اکنون اگر فرمول (۶-۲) تشکیل حمله تخاصمی در حمله FGSM بازگردیم، می‌توان گفت که این حمله در واقع

می تواند به عنوان یک گام از حل مسئله بیشینه‌سازی مهاجم در فرمول‌بندی (۱۳-۲) 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، کافیست همین گام معرفی شده را چندین بار تکرار کنیم و حاصل را روی  $\epsilon$ -کره‌ی حول نمونه‌ی سالم بیافکنیم. این روش بھینه‌سازی 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  $\infty$  بهره بگیریم، تشکیل یک حمله تخاصمی در PGD از تکرار گام زیر به‌دست خواهد آمد:

$$x^{t+1} = \Pi_{x+\mathcal{S}}(x^t + \alpha \cdot \text{sgn}(\nabla_x L(\theta, x, y))) \quad (14-2)$$

که در آن  $\Pi_{x+\mathcal{S}}$  نشان دهنده عملیات افکنش روی  $\epsilon$ -کره‌ی  $S + x$  بوده و این‌بار  $\alpha$  نمایانگر نرخ یادگیری<sup>۳</sup> نزول گرادیان افکنده است. نکته قابل توجه در این رابطه آن است که عملیات افکنش حول  $\epsilon$ -کره‌ی 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 تخاصمی را - در صورت امکان - برآورده خواهد کرد.

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

### ۳-۲ روش‌های دفاع در برابر حملات تخاصمی

دفاع در برابر حملات تخاصمی از دو جنبه کلی مورد بررسی قرار می‌گیرد [۱۳]:

		Target Classification ( $L_2$ )									
		0	1	2	3	4	5	6	7	8	9
Source Classification	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	1	1	1	1	1	1	1
	2	2	2	2	2	2	2	2	2	2	2
	3	3	3	3	3	3	3	3	3	3	3
	4	4	4	4	4	4	4	4	4	4	4
	5	5	5	5	5	5	5	5	5	5	5
	6	6	6	6	6	6	6	6	6	6	6
	7	7	7	7	7	7	7	7	7	7	7
	8	8	8	8	8	8	8	8	8	8	8
	9	9	9	9	9	9	9	9	9	9	9

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

<sup>1</sup>Gradient Descent

<sup>2</sup>Projected Gradient Descent

<sup>3</sup>Learning Rate

۱. پاکسازی<sup>۱</sup>: در این حالت، هدف پاکسازی ورودی‌های دسته‌بند و یا مقاوم‌سازی دسته‌بند از لحاظ ساختاری در راستای اصلاح خروجی آن در هنگام بروز حمله است به طوری که به خروجی در زمان‌های دیگر آسیبی وارد نشود.

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

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

### ۳-۱-۱ روش‌های پاکسازی حمله

در ادبیات مرتبط با دفاع در برابر حملات تخاصمی، عبارت پاکسازی به دسته خاصی از روش‌های ممکن برای دفاع اطلاق می‌شود. در این پژوهش، برای سهولت دسته‌بندی، کمی از تعریف رسمی پاکسازی تخاصمی<sup>۳</sup> دور شده‌ایم و آن را مطابق تعریف ارائه شده در بخش ۳-۲ در نظر می‌گیریم. بنابراین پاکسازی تخاصمی خود در این دسته قرار خواهد گرفت. با این مقدمه، روش‌های پاکسازی را می‌توان در ابعاد زیر دسته‌بندی نمود [۱۳، ۹، ۸]:

۱. آموزش تخاصمی<sup>۴</sup>: یکی از ابتدایی ترین روش‌های دفاع که می‌تواند قابل اعمال روی هر شبکه عصبی مورد حمله باشد. در [۱۶] پیشنهاد شده است که انجام فرایند آموزش روی ترکیبی از نمونه‌های سالم و

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 همنجان موثر	قوت حمله
کند تر، زیرا علاوه بر چندین گام نزول گرادیان، نیاز به انجام جستجوی دودویی برای بهینه سازی 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 پیاده سازی 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 پیاده سازی حمله
جهیه سفید	جهیه سفید	جهیه سفید	نوع دسترسی مهاجم

<sup>1</sup>Purification

<sup>2</sup>Detection

<sup>3</sup>Adversarial Purification

<sup>4</sup>Adversarial Training

نمونه‌های تخصصی می‌تواند باعث منظم‌سازی<sup>۱</sup> شبکه عصبی آموزش دیده و پرورش توانایی برای مقابله در برابر حملات تخصصی باشد. در [۱۲] این نظریه با اعمال حمله FGSM روی مجموعه داده‌های آموزشی و افزودن این نمونه‌ها 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 پیش‌تر توضیح داده شد، در [۱۸] روشی نوین برای آموزش تخصصی ارائه شده است که در آن تابع ضرر<sup>۲</sup> به فرم یک مسئله بهینه‌سازی کمینه-بیشینه<sup>۳</sup> بیان شده است (رابطه (۱۳-۲)) در این رابطه، مسئله بیشینه‌سازی درونی سعی می‌کند که شبیه فرمول‌بندی ارائه شده در بخش ۲-۱-۲-۲ برای حمله FGSM، قوی‌ترین مقدار<sup>۴</sup> ممکن برای بزرگ کردن مقدار تابع هزینه پیدا شود. این در حالیست که در مسئله کمینه‌سازی برونوی هدف کمینه کردن مقدار این تابع ضرر تخصصی با تنظیم کردن پارامترهای مدل ( $\theta$ ) است. بدین ترتیب، اگر مقدار تلورانس<sup>۵</sup> حمله را در نظر بگیریم، مدل با نمونه‌های تخصصی در یک<sup>۶</sup>-کره حول هر نمونه‌ی سالم آموزش داده خواهد شد.

مهم‌ترین ایراد آموزش تخصصی این که 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 مجموعه داده‌های آموزشی استفاده می‌کنند. برای حل این مسئله روش Free Adversarial Training [۲۰] 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 نویز 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 [۱۸] مورد آزمایش قرار گرفته و نتایج امیدوار کننده‌ای داشته است. همچنین در YOPO<sup>۵</sup> [۲۱] 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 AT [۲۲]<sup>۶</sup> GI-AT

<sup>1</sup> Regularization

<sup>2</sup> Loss

<sup>3</sup> Min-max

<sup>4</sup> Tolerance

<sup>5</sup> You Only Propagate Once

<sup>6</sup> Geometry-aware Instance-reweighted Adversarial Training

و GI-AT [۲۳] اشاره کرد. در این نکته بهره گرفته می شود که نمونه دادهایی که نزدیک مرازهای تصمیم‌گیری یک مدل قرار دارند، بیشتر می توانند در موفقیت یک حمله تخصصی تاثیرگذار باشند. بدین ترتیب این روش با بهره گیری از راهکارهای استاندارد آموزش تخصصی، هر یک از نمونه دادهای آموزشی را بر مبنای این که تولید حمله‌ی تخصصی موفق از روی آن‌ها چقدر دشوار است، وزن‌دهی کرده و در فرایند پس‌انتشار<sup>۱</sup> دخیل می کند. نهایتاً در Fast Adversarial Training این نکته به صورت تجربی نشان داده می شود که آموزش تخصصی با استفاده از حمله تک-مرحله‌ای FGSM ولی با آغاز تصادفی<sup>۲</sup> (برخلاف [۲۰]) می تواند 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 [۲۰] به نتیجه مشابه دست بیابند.

۲. تغییرات در مدل قربانی و فرایند آموزش: پژوهش [۲۴] 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، می توانند 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 [۱۶] دارای اعوجاج<sup>۳</sup> بیشتری نسبت به نمونه‌های متناظر سالم دارند و دیگر قابل صرف نظر نیستند. در این پژوهش سه روش پیش‌پردازش برای مقابله با حمله L-BFGS مورد بررسی قرار گرفته‌اند:

(آ) تزریق نویز: افزودن نویز گاوی اندک به ورودی‌ها می‌تواند به تشخیص درست تعداد بیشتری از نمونه‌های تخصصی در ازای مقدار اندکی هدر رفت دقت دسته‌بندی بیانجامد.

(ب) خودرمزنگار<sup>۴</sup>: در این روش یک خودرمزنگار با هدف بازسازی نمونه‌های سالم از روی نمونه‌های تخصصی، آموزش داده شده است که می‌تواند از حمله جلوگیری کند.

(ج) خودرمزنگار نویز گیر: در این روش، شبیه روش قبلی، یک خودرمزنگار نویز گیر 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 گاوی با میانگین ۰ و انحراف معیار متغیر  $\sigma$  ترکیب می‌شود. نشان داده شده است که با قرار دادن  $0.1 = \sigma$  این خودکنگار می‌تواند به

<sup>1</sup> Backpropagation

<sup>2</sup> Random Initialization

<sup>3</sup> 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>4</sup> Autoencoder

خوبی خودگذار قبلی عمل کند.

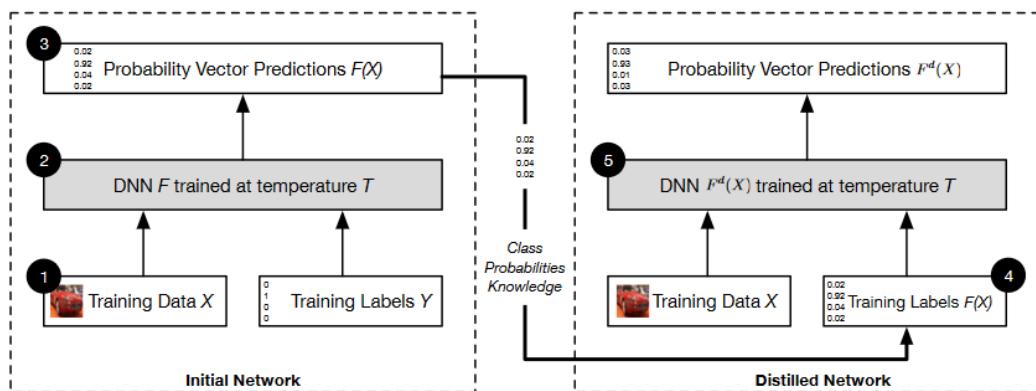
ایراد اصلی وارد به موارد ۲ ب و ۲ ج این است که با سری کردن خودزمگذار و مدل قربانی و حمله به مدل سری شده، همچنان می‌توان حمله موفقی داشت. این در حالیست که روش آن<sup>۲۲</sup> نمی‌توانست در برابر حملات جعبه‌سفید قوی‌تر که بعدها ارائه شدند (مانند PGD و CW) مقاومت کند.

ایده‌ی مهم دیگری که کمی بعدتر ارائه شد، ایده‌ی تقطیر دفاعی<sup>۱</sup> [۲۵] است. همانطور که در شکل ۷-۲ نشان داده شده است، ایده‌ی اصلی در این روش استفاده از پیش‌بینی‌های یک مدل از پیش‌آموزش داده شده روی مجموعه‌داده‌های سالم، به عنوان برچسب‌های جدید مدل تقطیر شده است. بدین ترتیب، مدل تقطیر شده برخلاف مدل اصلی دارای برچسب‌هایی 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 آینده ایده‌های قوی‌تری بر مبنای همین پژوهش ارائه شدند که مطالعه آن‌ها به خواننده واگذار می‌شود [۲۸-۲۶].

در [۲۹] 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 برچسب 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



شکل ۷-۲: چارچوب ارائه شده در روش تقطیر دفاعی [۲۵]

<sup>۱</sup>Defensive Distillation

بین پیش‌بینی مدل و برچسب واقعی تغییر چندانی نخواهد کرد و بنابراین نمونه‌های تخاصمی که در یک  $\epsilon$ -کره محدود می‌شوند، نخواهند توانست خروجی مدل قریب‌انی را تغییر دهند.

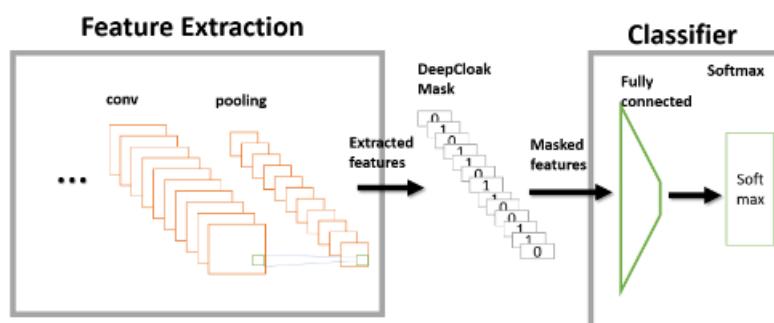
در روش DeepCloak [۳۰] که در شکل ۸-۲ نمایش داده شده، ایده مطرح شده آن است که با قرار دادن یک لایه ماسک‌کننده دقیقاً قبل از لایه خطی که تولید کننده logit های مدل، ویژگی‌های بی‌همیت در خروجی نهایی مدل را از صفر کرده و آن را نسبت به نویز تخاصمی مقاوم‌تر ساخت. آموزش این لایه با دادن ورودی‌های سالم و تخاصمی به مدل و encode کردن اختلاف بین ویژگی‌های آن‌ها در لایه پیشین، صورت می‌گیرد. قوت این روش در برابر حمله FGSM به صورت تجربی نشان داده شده است.

در ایده‌ای مشابه پژوهش قبلی و ترکیب آن با [۲۴]، در Random Self-Ensemble [۳۱] یک لایه نویز برای افزودن مقدار ناچیزی نویز ایزوتروپیک گاوی به ورودی‌های مدل و نیز خروجی‌های لایه‌های پنهان درون مدل قرار داده می‌شود:

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

مقادیر بزرگتر  $\sigma$  در این لایه مقاومت در برابر حملات تخاصمی 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، عملأً به ensemble ای از مدل‌ها دست پیدا کرد که بر اساس آن‌ها می‌توان تصمیم‌گیری‌های دقیق‌تری نسبت به نمونه‌های تخاصمی انجام داد.

۳. استفاده از شبکه‌های جانبی: در [۳۲] دو علت اصلی برای اشتباه دسته‌بندها در مواجهه با یک نمونه تخاصمی برشمرده شده است:



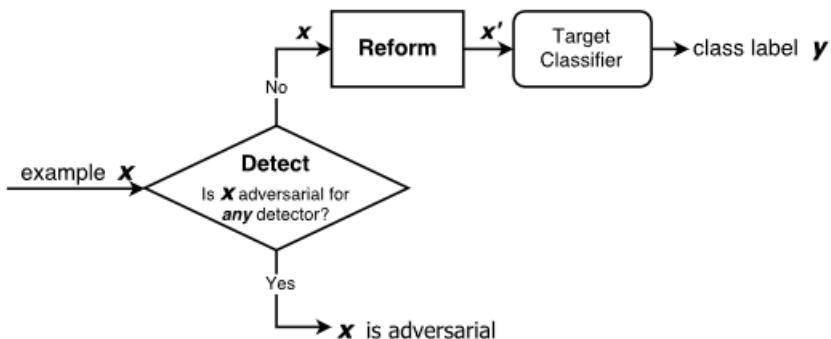
شکل ۸-۲: نحوه عملکرد [۳۰] DeepCloak

(آ) نمونه تخاصمی از مزهای 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 تخاصمی ممکن است تصویری باشد که اصلاً شامل یک عدد دست نویس نیست ولی دسته بند از آنجایی که مجبور به تولید خروجی است، دچار اشتباه خواهد شد.

(ب) نمونه تخاصمی به مزهای خمینه وظیفه مورد نظر خیلی نزدیک است. در این حالت که اکثر حملات تخاصمی جدید از آن استفاده می کنند، اگر دارای یک دسته بند باشیم، دسته بندی که در فضای اطراف یک نمونه تخاصمی قدرت تعییم پذیری 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 تشخیص دهنده استفاده می کند برای آن که فاصله یک نمونه زمان تست را با نمونه های آموزشی بسنجد. به عبارت دقیق تر، یک تشخیص دهنده تابع  $f : \mathcal{X} \rightarrow (0, 1)$  را یاد می گیرد که در آن  $\mathcal{X}$  مجموعه تمام نمونه های زمان تست است و خروجی این تشخیص دهنده معیاری از فاصله نمونه زمان تست با خمینه نمونه های سالم زمان آموزش است. و سپس برای برطرف کردن مشکل دوم، این چارچوب از یک بهساز<sup>۲</sup> برای تصحیح نمونه های دور از خمینه نمونه های سالم استفاده می شود. این شبکه بهساز توسط یک خود-رمزنگار پیاده سازی می شود. در صورت تشخیص نمونه تخاصمی توسط حداقل یکی از تشخیص دهنده ها، این نمونه به شبکه بهساز داده می شود و خروجی شبکه بهساز در نهایت به دسته بند می رسد (شکل ۹-۲).

یکی از مهم ترین ایده های مطرح شده در زمینه استفاده از شبکه های جانبی چارچوب Defense-GAN [۳۳]



شکل ۹-۲: نحوه عملکرد MagNet [۳۲]

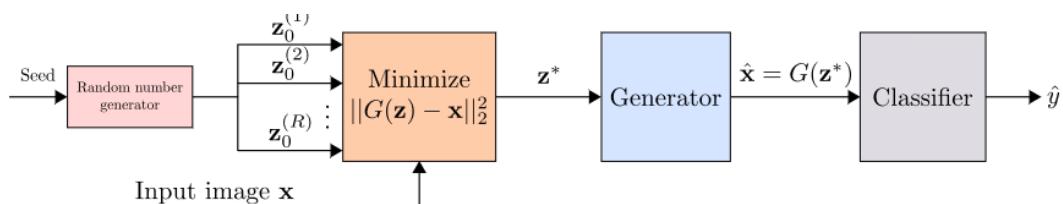
<sup>1</sup>Manifold  
<sup>2</sup>Reformer

است. در این پژوهش برای پاکسازی یک نمونه تخاصمی از یک شبکه مولد تخاصمی<sup>۱</sup> بهره برده می‌شود. در این چارچوب که در شکل ۱۰-۲ نمایش داده شده است، ابتدا یک WGAN<sup>۲</sup> 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 مولد، افکنده می‌شود. به عبارت دقیق‌تر، ابتدا  $R$  نمونه بردار نهفته<sup>۴</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  $G(z^*)$  نزدیک ترین نمونه‌ی ساختگی ممکن توسط مولد به نمونه‌ی احتمالاً تخاصمی ورودی 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 تخاصمی کمک خواهد کرد. یکی از نقاط قوت Defense-GAN ماهیت غیرخطی آن به دلیل وجود یک حلفه 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 تخاصمی مورد 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>۵</sup> [۳۴] مجدداً از یک شبکه مولد تخاصمی برای پاکسازی نویز تخاصمی استفاده شده است. این چارچوب که در شکل ۱۱-۲ نمایش داده شده است، از یک خودمزگذار به عنوان مولد و یک شبکه ممیز تشکیل شده است. هدف نهایی این چارچوب آن است که مولد  $G$  طوری



شکل ۱۰-۲ : دورنمای عملکرد چارچوب [۳۳] Defense-GAN

<sup>۱</sup>Generative Adversarial Network (GAN)

<sup>۲</sup>Wasserstein Generative Adversarial Network

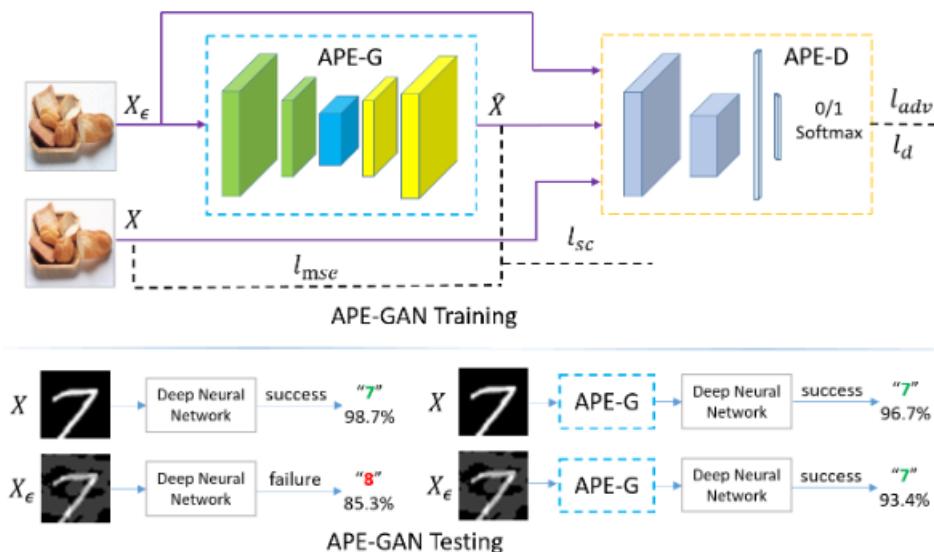
<sup>۳</sup>برای توضیحات مقدماتی راجع به GAN ها، می‌توانید به بخش ۱-۴-۲ مراجعه کنید

<sup>۴</sup>Latent

<sup>۵</sup>Adversarial Perturbation Elimination Generative Adversarial Network

آموزش ببیند که بتوان نویزهای ناچیز تخصصی را از روی ورودی تخصصی احتمالی، حذف کند، بدون آن که خروجی مدل در زمان دریافت ورودی‌های سالم دچار تغییر محسوسی بشود. برای تحقق این هدف از یک ساختار تخصصی و برقراری یک بازی خصم‌مانه 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 متضطر است. در نهایت این مولد قادر به تولید نمونه‌های پاکسازی شده‌ای خواهد بود که توسط ممیز غیر قابل تشخیص 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 تحويل داده می‌شوند.

در [۳۵] استفاده از یک دسته‌بند با Embedding منظم شده<sup>۱</sup> 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 ۱۲-۲ نشان داده شده است، هر دسته‌بند را می‌توان به صورت دو زیرشبکه متصور شد: یک Encoder که وظیفه آن استخراج ویژگی از ورودی دسته‌بند

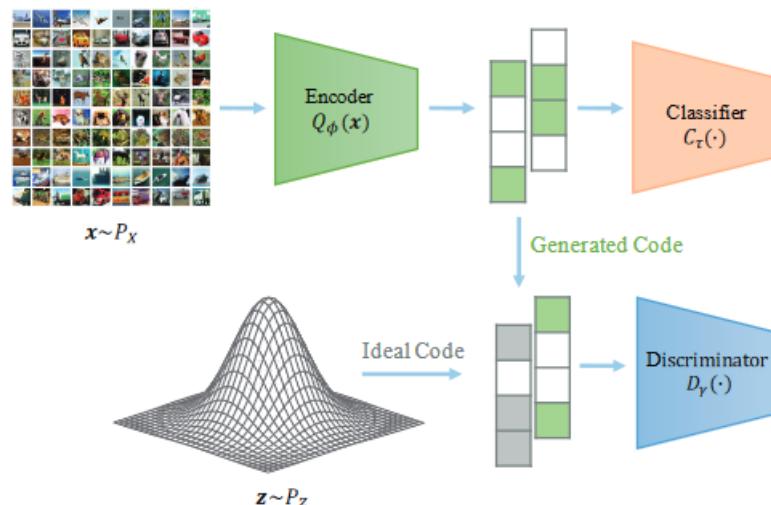


شکل ۱۱-۲: دورنمای APE-GAN در زمان آموزش و تست [۳۴]

<sup>۱</sup>Embedding Regularized Classifier

است و یک هد دسته‌بند برای تولید خروجی کلاس نهایی. اکتون 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 تخاصمی کاهش خواهد یافت.

یکی از جدیدترین ایده‌های مطرح شده در زمینه پاکسازی نمونه‌های تخاصمی DiffPure [۳۶] است. در این چارچوب از یک مدل انتشاری<sup>۳</sup> برای پاکسازی نویز تخاصمی تزریق شده به یک تصویر استفاده می‌شود<sup>۴</sup>. نحوه عملکرد این چارچوب در شکل ۱۳-۲ نمایش داده شده است. مدل‌های انتشاری از دو فرایند انتشار پیش‌رو<sup>۵</sup> و انتشار معکوس<sup>۶</sup> تشکیل می‌شوند. ایده‌ی اصلی DiffPure آن است که با اعمال انتشار پیش‌رو روی ورودی تخاصمی احتمالی و به تبع آن افزودن مقداری نویز ایزوتروپیک گاوی به آن، تاثیر نویز تخاصمی 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 تخاصمی دست یافته شود. این ایده به صورت تجربی و همچنین با شهود ریاضی در این مقاله مورد بررسی و به اثبات رسیده است.



شکل ۱۲-۲ : دورنمایی از [۳۵] ER-classifier

<sup>۱</sup>Prior

<sup>۲</sup>Discriminator

<sup>۳</sup>Diffusion Model

<sup>۴</sup> برای اطلاعات بیشتر راجع به مدل‌های انتشاری می‌توانید به بخش ۲-۴-۲ مراجعه کنید

<sup>۵</sup>Forward Diffusion

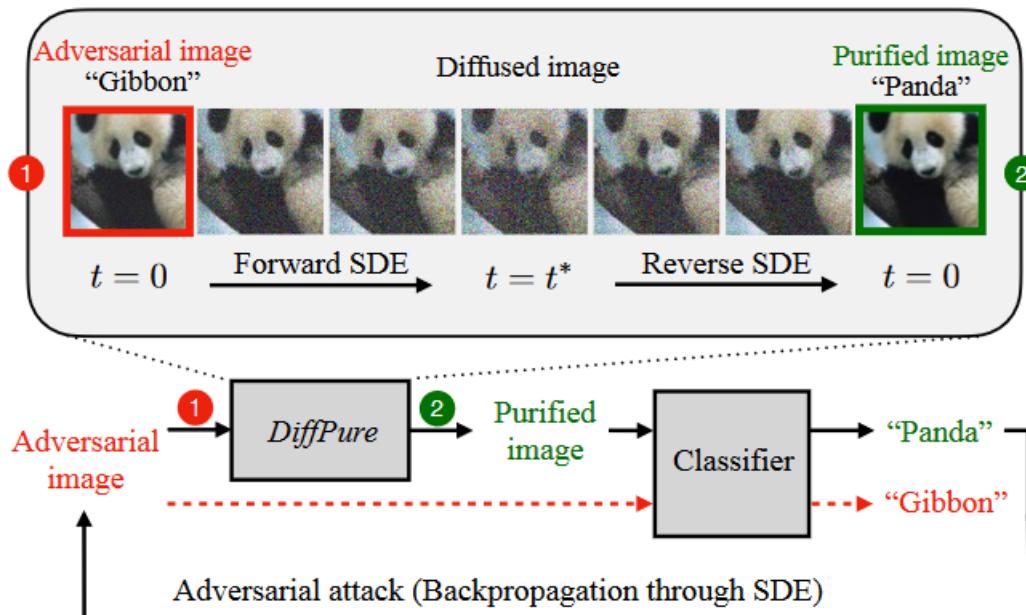
<sup>۶</sup>Reverse Diffusion

## ۲-۳-۲ روش‌های تشخیص حمله

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

۱. **تشخیص دهنده‌های 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 [۳۷] 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 تعیین‌پذیری کمی دارد و خصوصاً نسبت به نوع حمله مورد استفاده شده بسیار حساس است. در این پژوهش که از حملات FGSM و JSMA [۳۸] برای ارزیابی مدل استفاده شده است، نتایج تجربی نشان می‌دهند که تشخیص دهنده‌هایی که روی حملات FGSM شده‌اند، نمی‌توانند در برابر حملات JSMA به خوبی دفاع کنند و بر عکس.

به طور مشابه در [۳۹] از یک زیر-شبکه تشخیص دهنده 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 زیر‌شبکه تشخیص دهنده و نیز ادامه دسته‌بند، داده خواهند شد. این معماری در شکل ۱۴-۲ نشان داده شده است. برای آموختن این شبکه، ابتدا دسته‌بند اصلی روی نمونه‌های



شکل ۲-۱۴: نحوه عملکرد [۳۹] DiffPure

<sup>۱</sup>Hidden Layers

سالم آموزش داده خواهد شد. سپس، زیرشبکه‌های تشخیص دهنده در لایه‌های میانی به دسته‌بند اضافه می‌شوند و نمونه‌های تخاصمی به تعداد برابر نمونه‌های آموزشی، تشکیل خواهد شد. در نهایت وزن‌های دسته‌بند اصلی freeze شده و فقط زیرشبکه‌های تشخیص دهنده روی نمونه‌های سالم و تخاصمی به عنوان یک دسته‌بند دو- کلاسه، آموزش داده خواهد شد. در [۴۰] ایده‌ی مشابه ولی با ارتباط بین زیرشبکه‌های دسته‌بند با الهام گرفتن از ایده‌ی Boosting برای دسته‌بندی آبشراری<sup>۱</sup> [۴۱]، مطرح شده است.

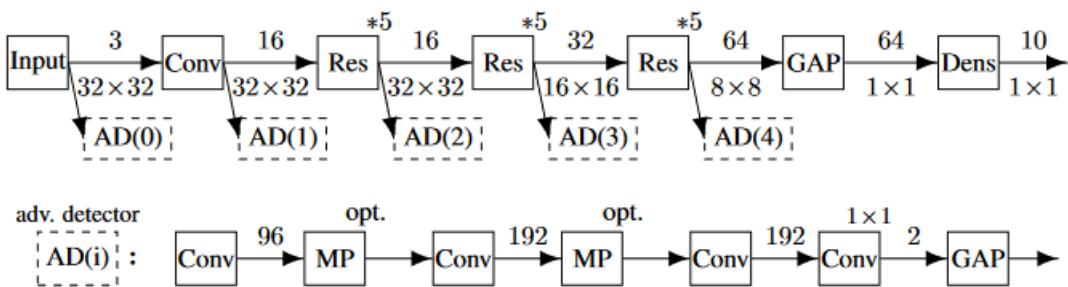
در [۴۲] چارچوبی تحت عنوان I-defender مطرح شده است که در آن توزیع احتمالی خروجی‌های لایه‌های کاملاً متصل<sup>۲</sup> یک مدل دسته‌بند در حین دسته‌بندی نمونه‌های سالم و تخاصمی مورد بررسی قرار گرفته است. در این پژوهش نشان داده شده است که توزیع احتمالی لایه‌های پنهان کاملاً متصل نه تنها برای کلاس‌های مختلف مجموعه‌داده آموزش، تفاوت دارد، بلکه برای یک کلاس در حالت سالم و تخاصمی 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 مجموعه‌داده مورد استفاده،

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

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 برای هر کلاس، می‌توان از  $TH_c$

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

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



شکل ۱۴-۲: معماری تشخیص دهنده [۳۹] Metzen

<sup>1</sup>Cascade

<sup>2</sup>Fully Connected

<sup>3</sup>Gaussian Mixture Model

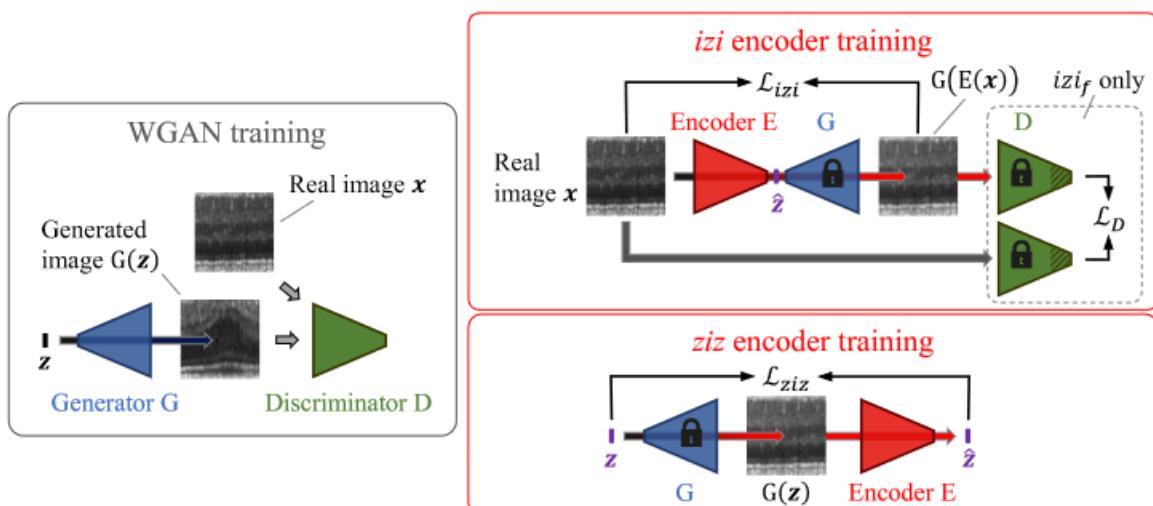
در ML-LOO [۴۳] مفهومی تحت عنوان Feature Attribution برای تشخیص حملات تخاصمی مطرح می‌شود. این مفهوم که برای هر ورودی  $x \in \mathbb{R}^d$  با  $\Phi(x) : \mathbb{R}^d \rightarrow \mathbb{R}^d$  نشان نمایش داده شده است، معیاری از میزان تاثیر هر ویژگی از ورودی (برای تصاویر، هر پیکسل از تصویر) در خروجی دسته‌بندی نهای است. 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 (مثلًاً ۰) جایگزین شده است، محاسبه می‌شود. به عبارت دقیق‌تر:

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

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

f-AnoGAN [۴۴] چارچوب جالب توجه دیگری با استفاده شبکه‌های مولد تخاصمیست 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 ۱۵-۲ آمده است، ابتدا یک WGAN 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 سنجه‌ی  $A_R(x)$  و  $A_D(x)$  استفاده می‌شود (شکل ۱۶-۲):

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



شکل ۲-۱۵: دورنمای نحوه آموزش [۴۴] f-AnoGAN

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 می‌رود پس از آموزش رمزگذار  $E$  روی داده‌های سالم، در حالت ایده‌آل، شبکه‌های  $G$  و  $E$  توابع عکس نظری یک دیگر باشند و بنابراین

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

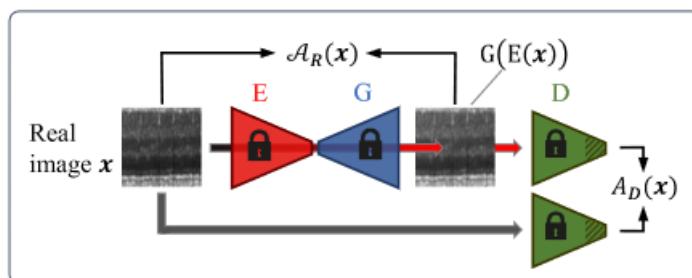
ولی اگر در نمونه  $x$  آنومالی وجود داشته باشد، از آنجایی که  $E$  فقط می‌تواند به فضای پنهان نمونه‌های سالم رمز کند، نُرم اختلاف دو مقدار  $x$  و  $G(E(x))$  می‌تواند سنجه خوبی برای تشخیص آنومالی باشد.

از طرف دیگر

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

که در آن  $(\cdot)$  ویژگی‌های یکی از لایه‌های میانی شبکه ممیز است، با استدلالی مشابه  $A_R(x)$  و با الهام گرفتن از روش تطبیق ویژگی<sup>۲</sup> [۴۵] به عنوان سنجه دیگر برای تشخیص آنومالی استفاده می‌شود. در نهایت ترکیب این دو سنجه امتیاز  $(x)$  را بدست خواهد داد که از آن و با کمک یک حد آستانه تنظیم شده می‌توان برای تشخیص آنومالی استفاده کرد.

در نهایت نگاهی به روش ارائه شده در [۴] 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<sup>۵</sup> ابتدا یک شبکه ACGAN [۴۶] روی نمونه‌های سالم آموزش داده می‌شود. سپس از تمام بخش‌های این شبکه برای ایجاد سنجه‌هایی برای تشخیص حمله استفاده می‌شود.



شکل ۲-۱۶: نحوه عملکرد f-AnoGAN در زمان تست [۴۴]

<sup>۱</sup>Reconstruction Error

<sup>۲</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 خطا خواهیم کرد

<sup>۴</sup>Class Conditional

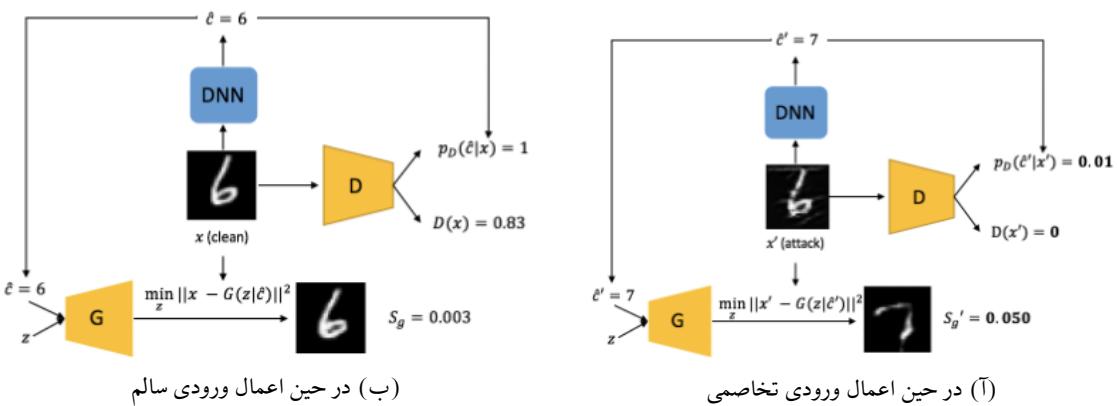
به طور مشخص، سنجه‌های مورد استفاده موارد زیر هستند:

$$S_R = D(x)$$

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

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

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 از داده‌های سالم است. این سنجه حملاتی را که نویز تخاصمی آن‌ها دارای نرم 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 ۱۷-۲ نشان داده شده است. در حین اعمال ورودی سالم (شکل ۱۷-۲ ب) مقدار  $p_D(\hat{c}|x)$  بالا بوده در حالی که مقدار  $S_g$  ناچیز است. در مقابل، در حین اعمال ورودی تخاصمی (شکل ۱۷-۲ ت) دقیقاً بر عکس این حالت اتفاق خواهد افتاد. در ادامه این پژوهش ترکیب‌های مختلفی از این سنجه‌های مربوط به زیرشبکه ممیز ( $S_R$  و  $S_C$ ) استفاده ممکن گزارش شده، حالت D-AD است که فقط از سنجه‌های مربوط به زیرشبکه ممیز ( $S_R$  و  $S_C$ ) استفاده



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

<sup>۱</sup>Posterior

<sup>۲</sup>Auxiliary Classifier

می‌کند.

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

برای تشخیص نمونه‌های تخاصمی که دور از خمینه نمونه‌های واقعی قرار دارند روشی تحت عنوان -KD [۴۷] پیشنهاد شد. در این روش از تخمین چگالی هسته‌ای<sup>۱</sup> (KDE) برای تخمین چگالی 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 مجھول 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  $(\cdot, \cdot) K_\sigma$  یکتابع 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 شدن، نمونه‌های زمان تست ابتدا به صورت موقت به دسته‌بند مورد دفع 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 [۴۸] 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 نرم نویز تخاصمی اعمال شده به نمونه باید محدود شده باشد، انتظار است که

---

<sup>۱</sup>Kernel Density Estimation

این نمونه‌های تخصصی از کلاس نمونه‌ی سالم متناظر شان نیز چندان دور نخواهد 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>  $KL^2$  [۴۹] بین معیارهای تخمین زده شده توسط خود دسته‌بند و مدل‌های تخمین چگالی و یک حد آستانه مناسب، برای تشخیص حملات استفاده می‌شود.

در [۵۰] روش دیگری تحت عنوان Mahalanobis Detector (MD) مطرح شده است. فرض کنید یک 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 گاوی چند متغیره پیروی می‌کند، می‌توان یک دسته‌بند مولد تعریف کرد. به عبارت دقیق‌تر، می‌توان  $C$  توزیع گاوی 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>۳</sup> بین نمونه  $x$  و نزدیک ترین گاوی مشروط بر کلاس موجود از بین  $C$  گاوی 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 [۵۱] روش آماری دیگری برای تشخیص نمونه‌های تخصصی ارائه شده است. اگر logit های پیش از اعمال softmax در یک دسته‌بند با بردار  $\vec{f}_y(x)$  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

<sup>1</sup>Divergence

<sup>2</sup>Kullback-Leibler

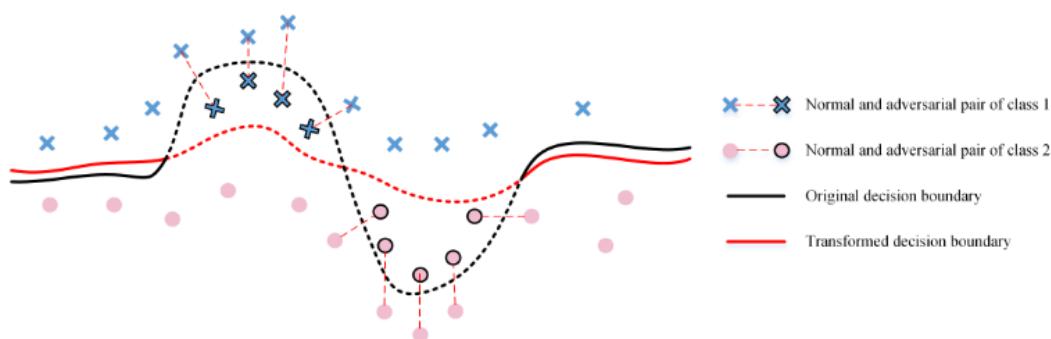
<sup>3</sup>Mahalanobis Distance

نیز برای دو کلاس  $y$  و  $z$  به صورت زیر تعریف شوند:

$$f_{y,z}(x) = f_z(x) - f_y(x) \quad (15-2)$$

اکنون، ایده‌ی اصلی مطرح شده در این پژوهش آن است که این مقدار log-odds جفتی چگونه با افروزن مقدار کمی نویز  $\eta$  به ورودی  $x$  تغییر می‌کند هنگامی که  $y = y_{true}$  و اگر برجسب‌های 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 متفاوت باشد. همچنین، اگر فرض شود که نویز تخاصمی افزوده شده به نمونه‌های سالم نسبت به نویز  $F(\hat{x}) = \hat{y} \neq F(x)$  مقاوم نیست، بنابراین اگر نمونه تخاصمی  $\hat{x} = x + \delta$  در اختیار باشد که  $y = f_{\hat{y},y}(\hat{x} + \eta) > f_{\hat{y},y}(\hat{x})$  چرا که نویز کوچک  $\eta$  اضافه شده مقدار کمی از تاثیر نویز تخاصمی  $\delta$  را از بین برده و مقدار log-odds جفتی بین کلاس به اشتباہ تشخیص داده شده توسط دسته‌بند و کلاس اصلس را طبق معادله (15-2) افزایش می‌دهد. بنابراین از امید ریاضی این مقدار log-odds جفتی (روی  $\eta$  های مختلف) می‌توان به عنوانی معیاری برای تشخیص نمونه‌های تخاصمی با استفاده از یک حد آستانه مطلوب بهره برد.

در چارچوبی تحت عنوان SID<sup>۱</sup> [۵۲] این ایده مطرح شده است که نمونه‌های تخاصمی از هل دادن نمونه‌های سالم به سمت دیگر مرز تصمیم در جاهایی که مرز تصمیم یک دسته‌بند دارای تلاطم زیادیست، تشکیل می‌شوند(شکل ۱۸-۲). 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 تخاصمی شد بدون آن که مرز تصمیم در نقاط کم تلاطم دچار تغییرات



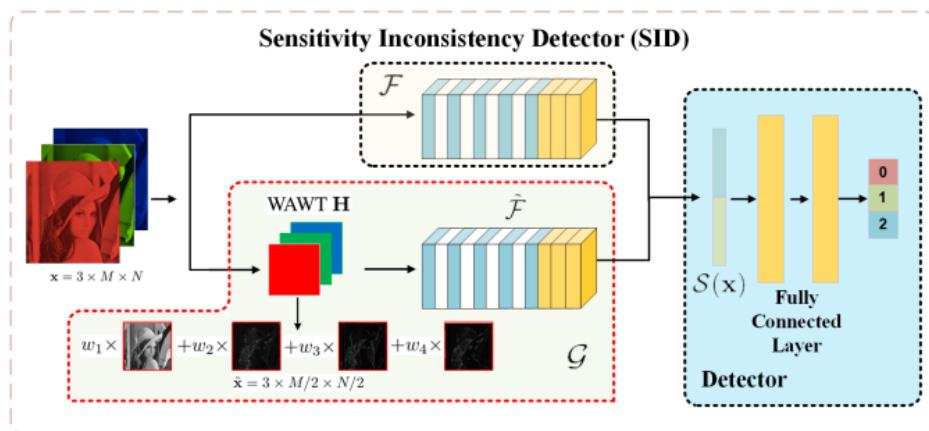
شکل ۱۸-۲: تشکیل نمونه‌های تخاصمی در اطراف خمیدگی‌های مرز تصمیم یک دسته‌بند [۵۲]

<sup>۱</sup>Sensitivity Inconsistency Detector

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

$$\min_{\mathcal{G}} \max_{x \in X} \|\mathcal{F}(x) - \mathcal{G}(x)\|_2^2, \quad \text{s.t. } \|\mathcal{F}(\hat{x}) - \mathcal{G}(\hat{x})\|_2^2 \geq \xi, \quad \forall \hat{x} \in \hat{X} \quad (16-2)$$

که در آن  $X$  مجموعه نمونه‌های سالم،  $\hat{X}$  مجموعه نمونه‌های تخاصمی 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 بر اساس اختلاف مرز تصمیم‌های ایجاد شده توسط دو دسته‌بند، شبکه تشخیص دهنده بتواند سالم و یا تخاصمی بودن یک نمونه‌ی ورودی را تشخیص دهد.



شکل ۱۹-۲: عملکرد SID در زمان تست [۵۲]

<sup>1</sup> Slack Variable

<sup>2</sup> Weighted Average Wavelet Transform (WAWT)

## ۴-۲ مختصری در مورد هوش مولد

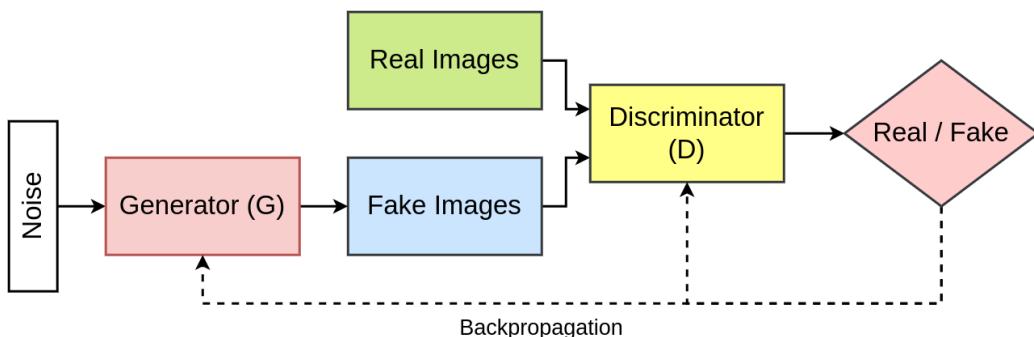
هوش مولد<sup>۱</sup> عبارتیست که به یکی از زیر شاخه های هوش مصنوعی اطلاق می شود که تمرکز اصلی آن تولید رسانه های مختلف از جمله متن، تصویر، فیلم و مدل های سه بعدی وغیره است. در هوش مولد از محتوای موجود کنونی و برای آموزش مدل هایی استفاده می شود که توانایی خلق محتوای جدید را داشته باشند بدون آن که لزوماً نمونه های آموزشی را دقیقاً در خروجی تکرار کنند. در حقیقت مدل های مولد تخمینی پارامتری از توزیع احتمالی پیچیده محتوای که قرار است تولید کنند را یاد میگیرند 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 حوضه‌ی 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 [۵۳] توسط آقای Goodfellow و همکارانش، در یکی از هوشمندانه ترین استفاده های نظریه‌ی 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 ۲۰-۲ نشان داده شده است، ایده‌ی کلی GAN ها برقراری یک بازی مجموع-صفرا<sup>۳</sup> بین دو شبکه به نام های مولد<sup>۴</sup> و ممیز است. هدف شبکه های مولد در حالت ایده‌آل تولید تصاویری از توزیع تصاویر حقیقی موجود است. برآورده کردن این هدف به کمک بازی ایجاد شده میان مولد و ممیز صورت می‌گیرد.



شکل ۲۰-۲: دورنمای شبکه های مولد تخصصی

<sup>1</sup>Generative AI

<sup>2</sup>Unsupervised Learning

<sup>3</sup>Zero-sum

<sup>4</sup>Generator

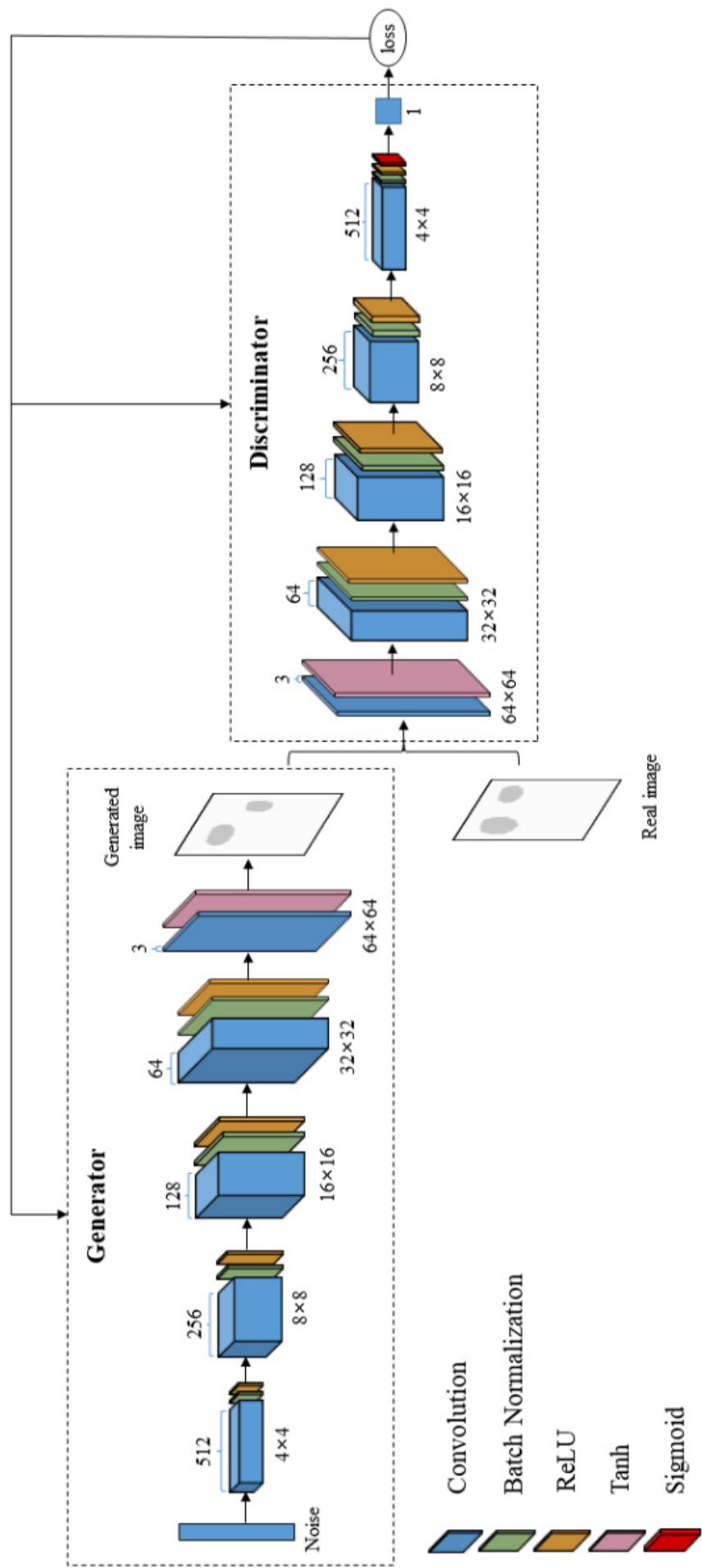
مولد تلاش می‌کند با تولید نمونه‌هایی که به نمونه‌های واقعی شبیه هستند، ممیز را فریب دهد. از طرف دیگر، ممیز در تلاش است که با تقویت خودش در برابر فریب خوردن از مولد مصون بماند و همچنان بتواند نمونه‌های ساختگی و واقعی را از هم تفکیک کند. به طور دقیق‌تر، مولد  $G$  و ممیز  $D$  بازی کمینه–بیشینه زیر را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

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (17-2)$$

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 بنده 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 [۵۳]).

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 ۲۱-۲ مشاهده می‌شود، مولد یک DCGAN از سری کردن چندین لایه Upsampling (که در اولین نمونه‌های DCGAN صرفاً با استفاده از کانولوشن ترانهاده<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 عملًا معیاری از احتمال واقعی بودن تصویر دریافت شده توسط ممیز است. شبکه‌های ممیز با گرفتن دسته‌ای از نمونه‌های واقعی و ساختگی مولد که دارای برچسب‌های 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 ۲۲-۲ 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 [۵۴، ۵۵]:

<sup>۱</sup>Transposed Convolution

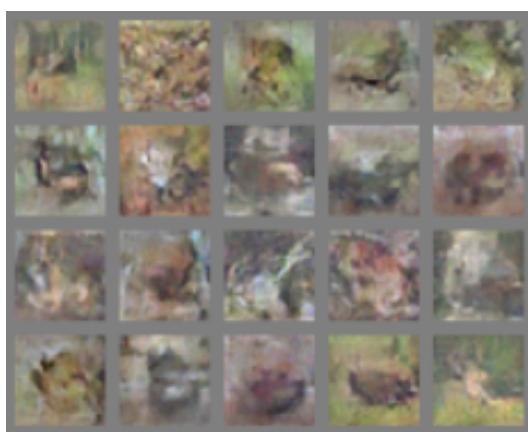


شکل ۲-۱: معماری شبکه DCGAN

۱. فروپاشی مُد<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<sup>۲</sup> 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 مدل گفته می‌شود.

۲. عدم همگرایی: رسیدن به نقطه‌ی تعادل تعادل نَش<sup>۳</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 گاهًا 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 فرآیند آموزش: حالت مقابل فروپاشی مدل 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 ۲-۲: نمونه‌های تولید شده توسط یک GAN معمولی [۵۳]

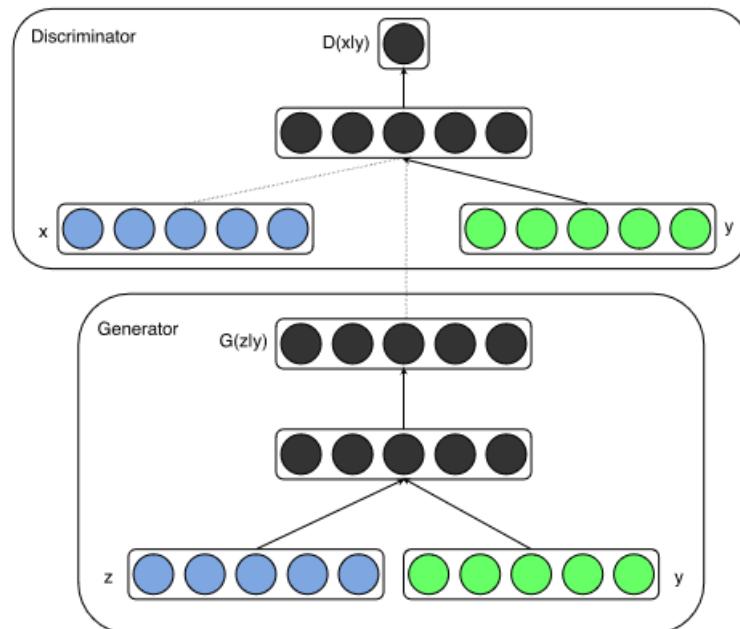
<sup>1</sup>Mode Collapse<sup>2</sup>Nash Equilibrium

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 مذکور شد اما نشان داده شده است که می‌تواند روی بهبود همگرایی و پایداری فرایند آموزش نیز تاثیرگذار باشد [۵۶، ۵۷]، مشروط کردن خروجی مولد به برچسب مورد انتظار است. این ایده منجر به معرفی ساختارهای جدیدی به نامهای GAN<sup>c</sup><sup>2</sup> و کمی بعدتر ACGAN شد که در ادامه کمی بیشتر این دو ساختار را مورد بررسی قرار خواهیم داد.

#### cGAN ۱-۱-۴-۲

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 [۵۸] محققین معماری cGAN<sup>۳</sup> را معرفی کردند. ایده‌ی تزریق یک شرط به ورودی‌های مولد و ممیز در



شکل ۲-۲: معماری cGAN [۵۸]

<sup>1</sup>Vanishing Gradient

<sup>2</sup>Conditional Generative Adversarial Network

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 ۲۳-۲ برای تزریق این شرط در هر دو شبکه‌ی مولد و ممیز یک لایه‌ی 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 خواهد کرد. در این وضعیت مولد  $G$  و ممیز  $D$  توزیع‌های شرطی  $G(z|y)$  و  $D(x|y)$  را تخمین می‌زنند و بازی (۱۷-۲) 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 ۲۴-۲ 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 ۲۴-۲: تاثیر تغییر برچسب ورودی مولد در نمونه‌های تولید شده توسط یک cGAN. هر سطر از شکل متناظر با یک برچسب ورودی در مولد است [۵۸].

<sup>1</sup>Concatenate

در [۴۶] 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 مُدهای 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 ۲۵-۲ 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. احتمال واقعی بودن نمونه‌ی ورودی.  $P(S|x)$

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

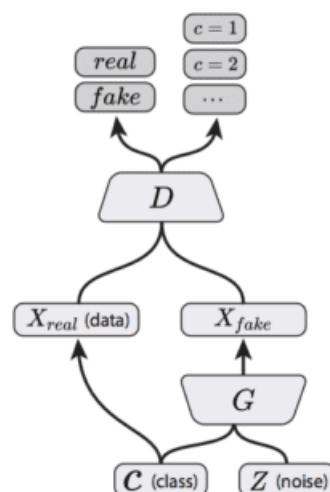
به تبع این موضوع،تابع هزینه‌ی آموزش یک ACGAN از دو بخش تشکیل می‌شود:

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

و

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

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 ترقیب به تشخیص کلاس درست ورودی دریافت شده خواهد کرد.



شکل ۲۵-۲: معماری ACGAN

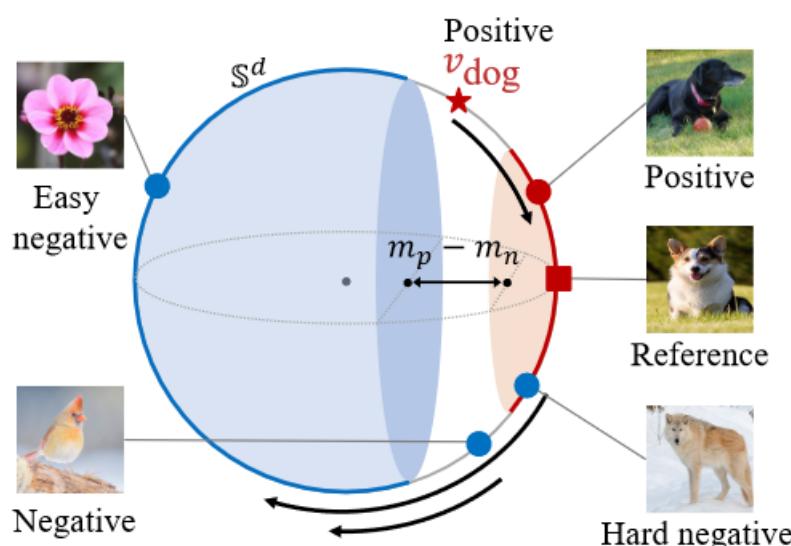
بدین ترتیب ممیز در راستای کمینه کردن  $L_C - L_S$  و مولد در راستای کمینه کردن  $L_C + L_S$  آموزش خواهد دید. با وجود بهبودهای این روش نسبت به GAN، این معماری همچنان خالی از ایراد نبود. یکی از اصلی‌ترین ایراد‌های وارد به این معماری، عدم توانایی مولد در یادگیری هنگامیست که تعداد کلاس‌های مجموعه داده‌ها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

#### ReACGAN ۳-۱-۴-۲

در [۵] معماری جدیدی برای تولید مشروط تصاویر به نام ReACGAN در راستای رفع دو ایراد اساسی ACGAN‌ها ارائه شد. محققین در این پژوهش، دو ایراد اصلی معماری ACGAN را چنین بر می‌شمارند:

۱. عدم پایداری آموزش زمانی که تعداد کلاس‌های مجموعه 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 ابرکره‌ی 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 D2D-CE<sup>۱</sup> به عنوان تابع هزینه‌ی آموزش جدید مدل معرفی می‌شود. هدف این تابع هزینه، استفاده از ویژگی‌های مستر در خود نمونه‌های آموزش با در نظر گرفت تفاوت‌ها و شباهت‌های آن‌ها نسبت به فاصله‌ی آن‌ها در فضای embedding است. همانطور که در شکل



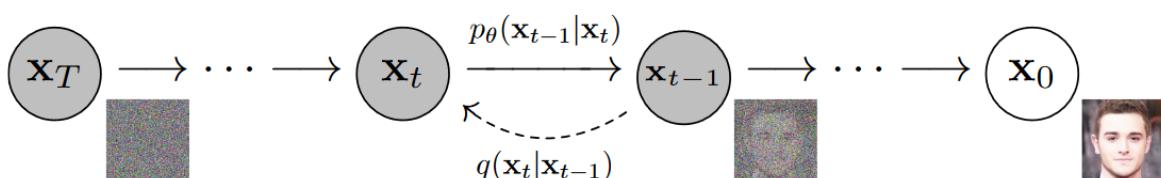
شکل ۲۶-۲: فضای embedding تصاویر در یک

<sup>۱</sup>Data-to-Data Cross-Entropy

۲۶-۲ مشاهده می‌شود فضای embedding یک ReACGAN برخلاف روش‌های قبلی (مانند ACGAN) روی یک ابرکرهی  $\mathbb{S}^d$  قرار می‌گیرد. از فواید مهم تابع هزینه‌ی D2D-CE که در همین شکل مشخص هستند می‌توان به استخراج نمونه‌های منفی دشوار و سرکوب کردن نمونه‌های مثبت و منفی آسان اشاره کرد. اگر تصویری به عنوان مرجع<sup>۱</sup> در اختیار باشد، مثلاً تصویری با کلاس سگ در شکل ۲۶-۲، یک نمونه‌ی منفی دشوار، نمونه‌ای خواهد بود که از لحاظ ویژگی‌های تصویری بسیار مشابه کلاس مرجع باشد (مثلاً تصویر یک گرگ). از طرف دیگر، یک نمونه‌ی منفی آسان در نقطه‌ی مقابل یک منفی دشوار قرار می‌گیرد (مثلاً تصویر یک گل). نهایتاً یک نمونه‌ی مثبت آسان، نمونه‌ایست که مانند یک منفی دشوار، ویژگی‌های مشترک زیادی با تصویر مرجع داشته باشد ولی برچسب آن نیز با برچسب مرجع برابر باشد. برای ایجاد تنوع در نمونه‌های تولید شده D2D-CE تلاش می‌کند که embedding نمونه‌های مثبت آسان را بیش از حد به هم نزدیک نکند تا کمی واریانس در خروجی‌های مولد باقی بماند. همچنین از طرف دیگر برای این‌که نمونه‌های تولید شده در حین داشتن تنوع، همچنان متناظر با برچسب‌های صحیح‌شان باشند.

## ۲-۴-۲ مدل‌های انتشاری

دسته‌ی دیگری از مدل‌های مولد که اخیراً معرفی شده‌اند و به سرعت در حال گسترش هستند، مدل‌های انتشاری هستند. ایده‌ی مدل‌های 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 مدل‌های انتشاری در سال ۲۰۱۵ و تقریباً به صورت همزمان با GAN منتشر شد [۵۹]، آوازه این مدل‌ها اولین بار در ۲۰۲۰ پس از انتشار مقاله معروف [۶] تحت عنوان مدل‌های انتشار احتمالی حذف کننده نویز<sup>۳</sup> بر سر زبان‌ها آمد. مدل‌های انتشاری یک زنجیره مارکوف<sup>۴</sup> از گام‌های انتشار را تعریف می‌کنند که به تدریج مقداری نویز تصادفی به یک نمونه می‌افزایند و سپس یاد می‌گیرند که عکس این فرایند انتشار را انجام دهند تا بتوانند با شروع از نویز به یک نمونه‌ی ساختگی که به توزیع نمونه‌های واقعی نزدیک است، برسند (شکل ۲۷-۲).



شکل ۲۷-۲: زنجیره مارکوف نظیر یک DDPM [۶]

<sup>1</sup>Reference

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

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

<sup>4</sup>Markov Chain

همان‌طور که پیش‌تر توضیح داده شد، مدل‌های انتشاری از دو فرایند انتشار پیش‌رو و انتشار معکوس تشکیل می‌شوند. در ادامه به طور مختصر به توزیع این دو فرایند پرداخته خواهد شد [۶۱، ۶۰]:

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

$$q(x_t|x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t \mathbf{I}\right) \quad q(x_{1:T}|x_0) = \prod_{t=1}^T q(x_t|x_{t-1}) \quad (18-2)$$

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

۲. انتشار معکوس: حال اگر بتوان به نحوی معکوس پروسه‌ی انتشار پیش‌رو  $q$  را یاد گرفت و به جای نمونه‌گیری از  $(x_{t-1}|x_t)$  از  $q$  نمونه برداری کرد، می‌توان با شروع از نمونه‌ای از یک گاویسی ایزوتروپیک و اعمال  $T$  گام زمانی معکوس، به نمونه‌ای از توزیع واقعی  $(x)$  رسید. می‌توان نشان داد که اگر  $\beta_t$  به اندازه کافی کوچک باشد، انتشار معکوس  $(x_{t-1}|x_t)$   $q$  نیز تقریباً یک گاویسی است. ولی از آنجایی که محاسبه‌ی  $(x_{t-1}|x_t)$   $q$  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:

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^T p_\theta(x_{t-1}|x_t) \quad p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}|\mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \quad (19-2)$$

در [۶] نشان داده می‌شود که می‌توان به جای تخمین زدن پارامترهای گاویسی 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  $T \leq t \leq 1$  است.

از آنجایی که هدف آموزشی مدل‌های انتشاری مطابق تخمین بیشینه درست‌نمایی<sup>۱</sup> انجام می‌شود که به مراتب ساختار یافته تراز بازی کمینه-بیشینه بین مولد و ممیز در یک GAN است، آموزش مدل‌های انتشاری پایدار‌تر 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 گاووسی تخمین زد و بنابراین به مقدار  $T$  (تعداد کل گام‌های زمانی فرایند انتشار) باید بزرگ باشد (مثلًاً ۱۰۰۰ گام) و در نتیجه برای تولید هر نمونه به تعداد قابل توجهی forward pass نیاز است که می‌تواند هزینه برباشد. با این وجود، نمونه گیری 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 ۲۸-۲ 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 توصیف کننده 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>۲</sup>: ایده‌ی ارائه شده در [۶۴] این بود که ابتدا یک دسته‌بند  $f_\phi(y|x_t)$



شکل ۲۸-۲: نمونه‌های تولید شده توسط DDPM روی مجموعه داده Celeb-HQ [۶]

<sup>1</sup>Maximum Likelihood Estimation

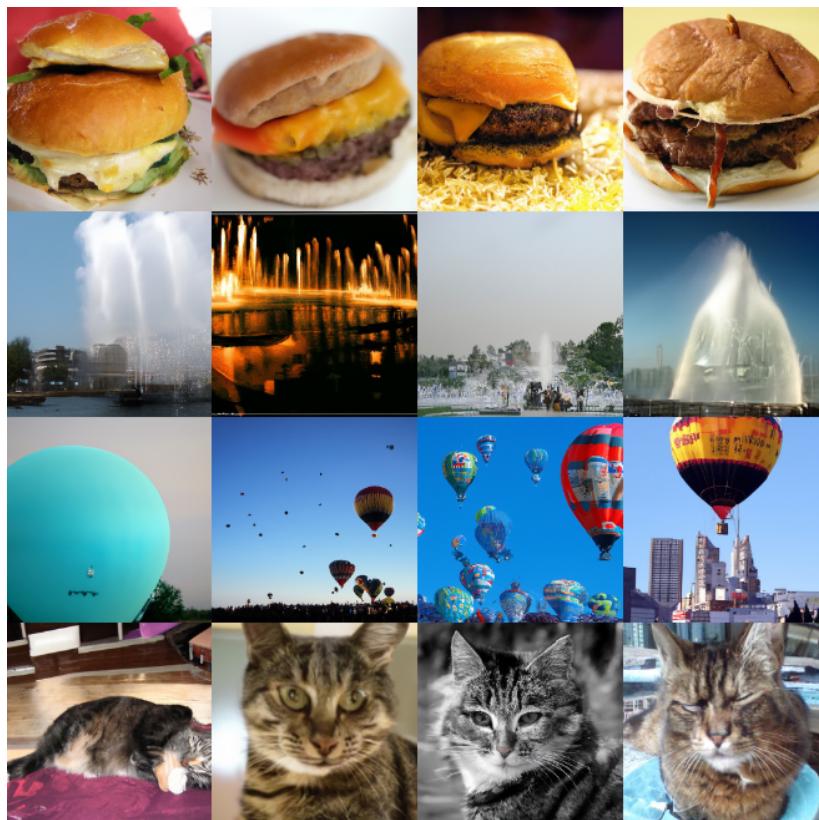
<sup>2</sup>Classifier Guided Diffusion Model

روی تصاویر نویزی  $x_t$  آموزش داده شده و سپس از لگاریتم گرادیان خروجی این دسته‌بند نسبت به تصویر 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 می‌شود. 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 دقیق‌تر خواهیم داشت:

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

که 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 می‌کند. در شکل ۲۹-۲ 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



شکل ۲۹-۲: نمونه‌های تولید شده توسط Classifier Guided Diffusion Classifier. روی مجموعه داده ImageNet استفاده شده به عنوان شرط برای مدل انتشاری در سطرها به ترتیب از بالا به پایین: چیزبرگر، آبنما، بالون، گربه tabby [۹۴]

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

توزیع  $(x|y)$  و  $p_\theta(x|y)$  را یادگرفت، آنگاه می‌توان از ترکیب نمونه‌های این دو توزیع، نمونه‌های جدیدی 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 تنها‌ی آن است که خروجی‌های این توزیع معمولاً بسیار شبیه هم هستند چرا که ویژگی‌های کلی متاظر 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  $W \times H$  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 نمونه اصلی ناچیز است. به عبارت دقیق تر 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 پاسازی - مطابق تعاریف ارائه شده در بخش ۳-۲ - چنین حملاتی است.

### ۳-۳ روش پیشنهادی

روش پیشنهادی ما بر مبنای روش ACGAN-ADA که در [۴] ارائه شده است و پیش تر در بخش ۲-۳-۲ راجع به آن بحث شد، بیان‌گذاری می‌شود. در این پژوهش، روش پیشین ACGAN-ADA از چند جهت بهبود داده شده است که در نهایت منجر به چارچوب پیشنهادی ما یعنی ARCANE<sup>۱</sup> خواهد شد. در ادامه به بررسی این تغییرات می‌پردازیم.

#### ۳-۳-۱ استفاده از مولدهای قوی تر

یکی از ایرادات مطرح شده توسط خود محققین ACGAN-ADA، کمبودهای ذاتی معماری ACGAN در مدل‌سازی آماری داده‌های دارای مُدهای زیاد است. ایراداتی که در ابتدای بخش ۳-۱-۴-۲ 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 ۳-۱-۴-۲) و یک مدل انتشاری شرطی (بخش ۱-۲-۴-۲) 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 نمونه‌های تولید شده، نمونه‌های نزدیک تری به توزیع واقعی نمونه‌ها تولید کند.

<sup>۱</sup> Adversarial Robustness using Class-conditionAl geNerative modEls

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

در روش اصلی مبنای این پژوهش، ACGAN-ADA، برای تشخیص حملات تخصصی 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 منزلهٔ بزرگ بودن میزان نویز تخصصی تزریق شده به نمونهٔ ورودی و مواجهه با یک نمونهٔ تخصصی 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 نظری دسته‌بند قربانی نباشد، احتمالاً با یک نمونهٔ تخصصی 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 نمونهٔ ورودی کوچک باشد بنابراین احتمالاً نمونهٔ ورودی در توزیع آماری شرطی یادگیری شده توسط مولد قرار دارد و سالم است. این معیار، همان معیار معروف MSE<sup>۱</sup> است که به طور معمول در پاکسازی‌های مبتنی بر بهینه‌سازی استفاده شده است [۴، ۳۳].<sup>۲</sup>

در ACGAN-ADA برای تعیین حد آستانه برای هر کدام از معیارهای ارائه شده از روش جستجوی شبکه‌ای روی آستانه‌ها استفاده شده است. در پژوهش ما، علاوه بر این سه معیار دیگر نیز برای تشخیص حمله بهره بردۀ 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 ۰.۹۹۸ در یک دسته‌بندی ۱۰ کلاسه) و یا قطعیت نسبتاً پایین (مثلاً ۰.۳ در همان دسته‌بندی ۱۰ کلاسه) می‌تواند نمایانگر یک حملهٔ تخصصی باشد.

<sup>۱</sup>Mean Squared Error  
<sup>۲</sup>لازم به ذکر است که در این پژوهش از معیار SSIM و ترکیب آن با MSE نیز برای محاسبهٔ  $S_g$  استفاده شد که نتیجه‌ی بهتری نسبت به MSE به تنهایی نداشت.

۵.  $JSD(p_D(x) || p_{victim}(x))$ : اختلاف بین احتمالات تولید شده توسط دسته‌بند قربانی و دسته‌بند جانبی  $S_C$  یک ACGAN می‌تواند به عنوان معیاری برای تشخیص حمله مورد استفاده قرار بگیرد. برخلاف  $C$  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) است. این دیورژانش که حالت متقاضی دیورژانس KL است برای دو توزیع احتمالی  $P$  و  $Q$  به ترتیب زیر تعریف می‌شود:

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

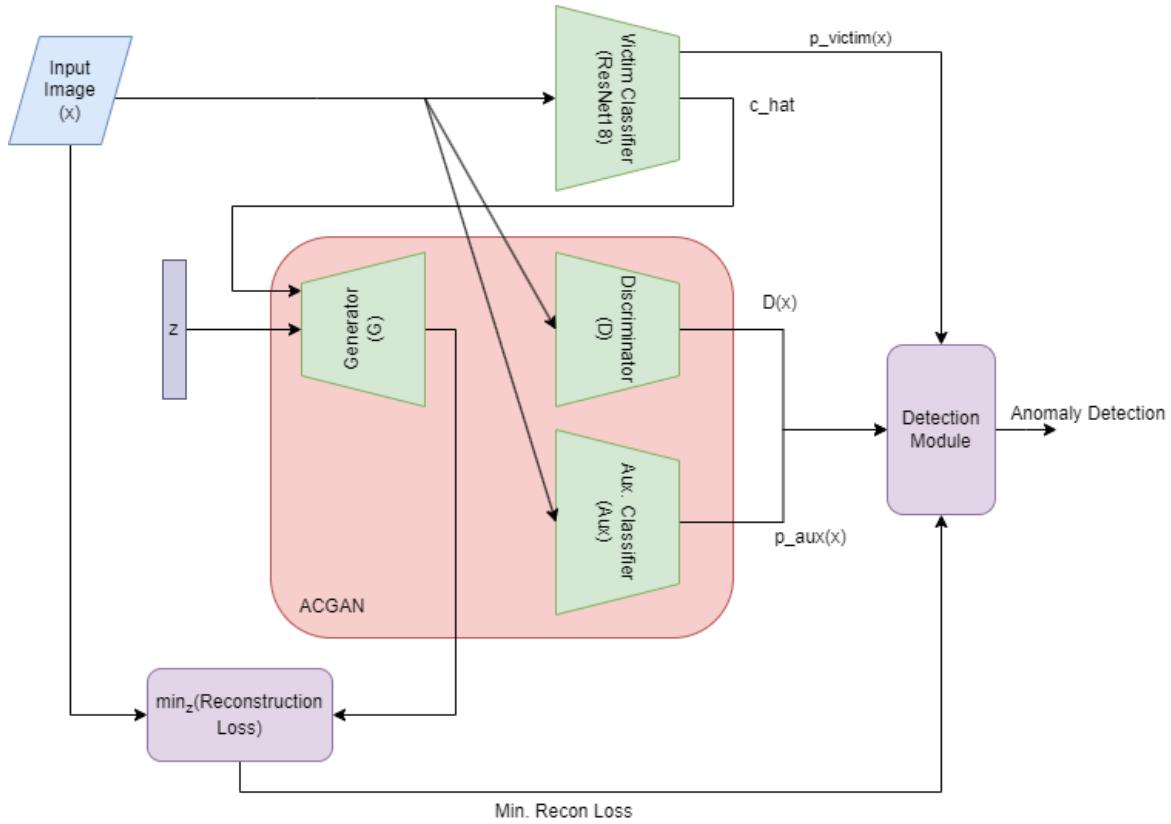
که در آن  $D_{KL}$  همان دیورژانش KL بوده و  $M = \frac{1}{2}(P + Q)$ .

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 تنها‌ی ارائه دهد.

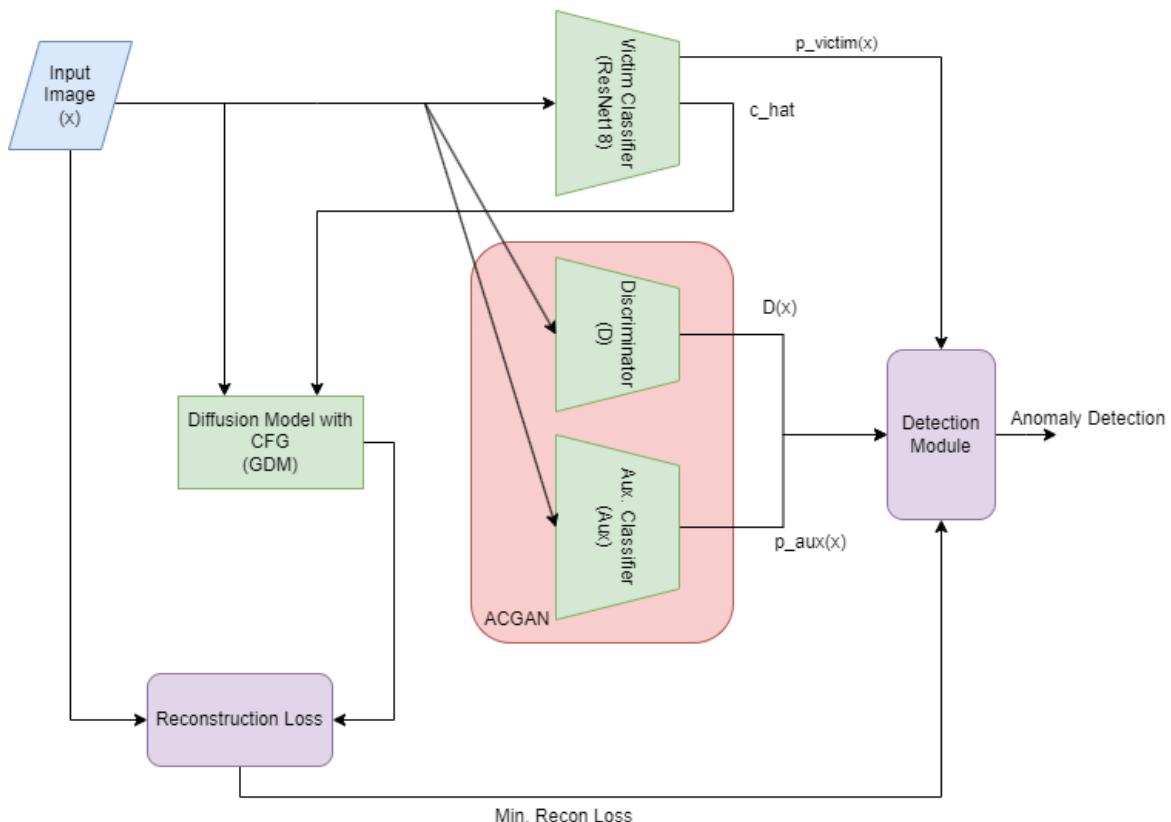
از طرف دیگر، در ARCANE برای تشخیص حمله از جستجوی شبکه‌ای برای پیدا کردن حدود آستانه استفاده نمی‌کنیم. در عوض تمامی این معیارها به صورت داده‌های جدولی<sup>۲</sup> به یک دسته‌بند XGBoost [۶۸] داده می‌شوند و تصمیم‌گیری نهایی توسط این دسته‌بند ثانویه صورت می‌گیرد. دورنمای کلی تشخیص حمله توسط تصاویر ورودی، 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>1</sup>Joint

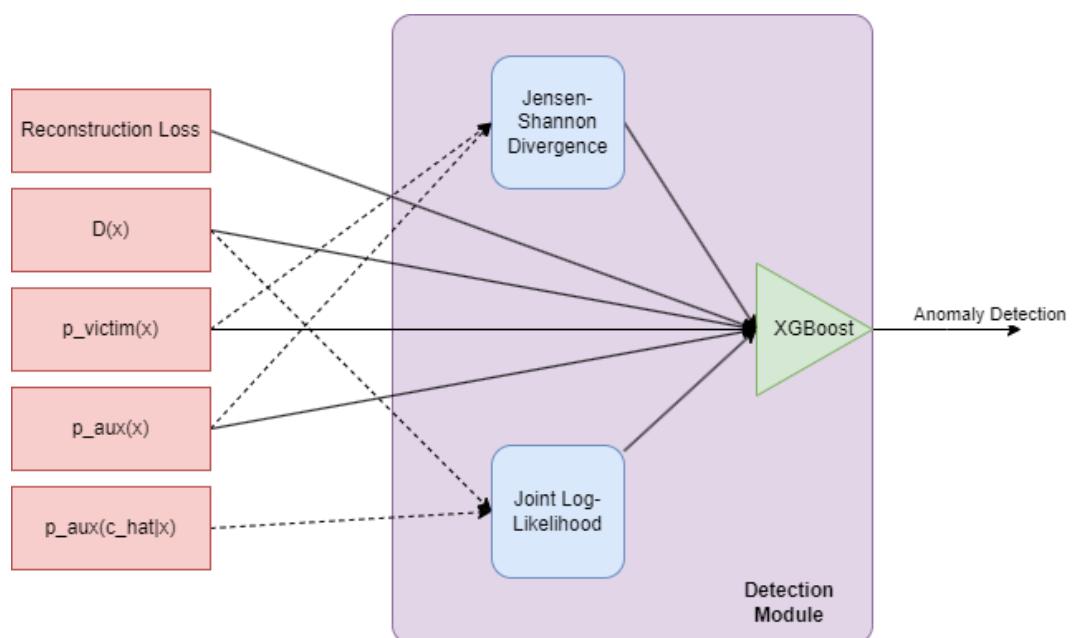
<sup>2</sup>Tabular



شكل ۱-۳: نحوی تشخیص حمله توسط ARCANE-GAN



شكل ۲-۳: نحوی تشخیص حمله توسط ARCANE-Diff



شکل ۳-۳: دورنمای ساختار تصمیم‌گیرنده‌ی 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 برچسب‌ها به صورت همزمان صورت می‌گیرد و در نهایت بین تمام نمونه‌های تولید شده، هر کدام که مقدار نرم اختلافش با نمونه‌ی تخاصمی ورودی کمینه بود به عنوان نمونه‌ی پاکسازی شده برای دسته‌بندی نهایی به مدل قربانی داده خواهد شد که کلاس تمیز را تولید خواهد کرد.

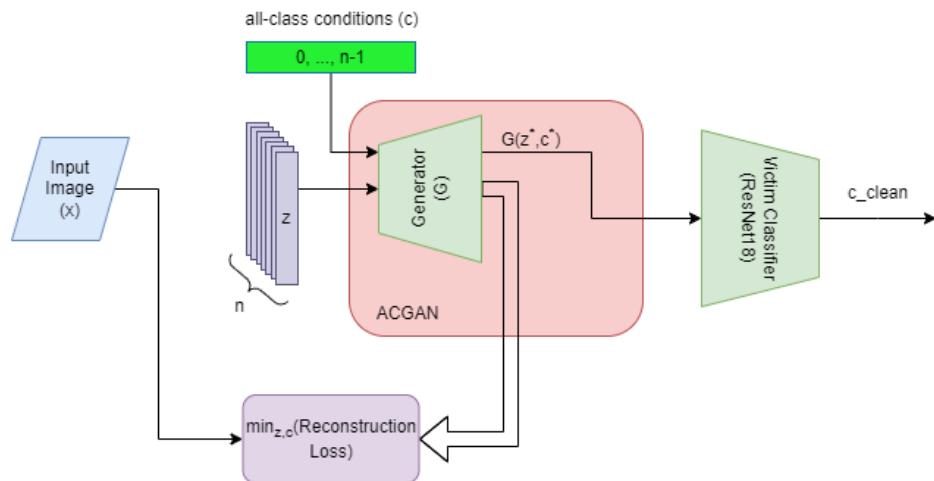
برای تولید تصاویر با استفاده از مولد ReACGAN از ایده‌ای شبیه Defense-GAN [۳۳] ولی به صورت شرطی بهره برد می‌شود. ابتدا یک بردار پنهان  $\mathbf{z}$  با ابعاد مورد انتظار مولد تولید می‌شود. سپس  $n$  (تعداد کلاس‌ها) کپی از این بردار به صورت همزمان ولی با کلاس‌های متناظر مختلف به مولد داده می‌شوند. سپس 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 [۳۶] 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 تخاصمی برسیم.

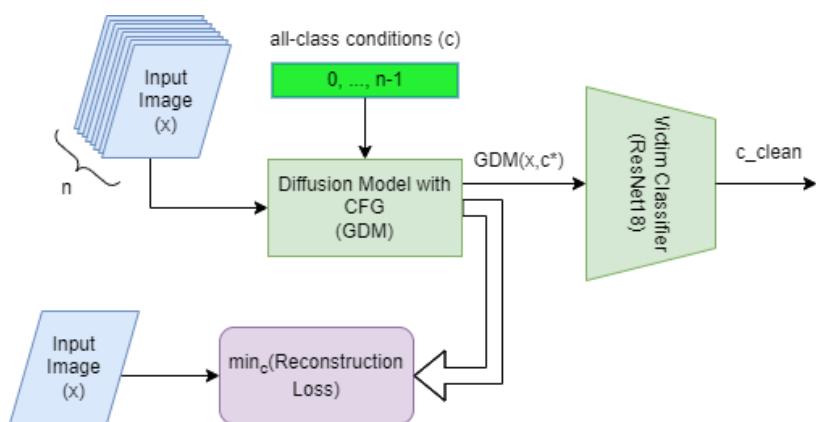
برای حصول اطمینان از تولید تصاویر با کیفیت از لحاظ ترکیب رنگ، فقط در مرحله‌ی پاکسازی، بهترین خروجی‌های هر دو مولد از یک فیلتر تطبیق هیستوگرام<sup>۱</sup> نیز عبور خواهند کرد تا مطمئن شویم ترکیب رنگ خروجی نهایی تا حد ممکن به تصویر ورودی نزدیک است. این امر باعث بهبود دقیق پاکسازی می‌شود. چارچوب کلی پاکسازی ARCANE-GAN و ARCANE-DIFF را در شکل‌های ۴-۳ و ۵-۳ مشاهده می‌کنید.

---

<sup>۱</sup>Histogram Matching



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



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

## ۱ - ۴ مقدمه

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

## ۲ - ۴ روش شبیه سازی

تمامی مدل‌ها و آزمایش‌های صورت گرفته روی بستر کتابخانه یادگیری عمیق معروف PyTorch [۶۹]<sup>۱</sup> و با استفاده از زبان برنامه‌نویسی پایتون<sup>۲</sup> پیاده سازی شده‌اند. برای آموزش مدل ReACGAN از مخزن-PyTorch [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 مدل‌های مولد از مجموعه داده‌های CIFAR10 [۷۱] و Tiny-ImageNet [۷۲] استفاده شده است.

<sup>1</sup><https://pytorch.org/>

<sup>2</sup>Python

<sup>3</sup><https://github.com/POSTECH-CVLab/PyTorch-StudioGAN>

<sup>4</sup><https://github.com/lucidrains/denoising-diffusion-pytorch>

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

برای آزمون مدل‌ها تعداد ۱۰۰۰ نمونه از هر مجموعه داده به صورت تصادفی اما متعادل در بین تمام کلاس‌های ممکن انتخاب شدند و سپس از روی این ۱۰۰۰ نمونه به ازای هر روش حمله یک نمونه‌ی تخاصمی ایجاد شده است. به عنوان مثال، برای مجموعه داده‌ی CIFAR10 ۱۰۰۰ نمونه‌ی سالم، از هر کلاس ۱۰۰ نمونه، انتخاب شده است و سپس ۱۰۰۰ نمونه حمله‌ی FGSM و ۱۰۰۰ نمونه حمله‌ی CW متناظر با این ۱۰۰ 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 ۲۰۰۰ نمونه (۱۰۰۰ نمونه سالم و ۱۰۰۰ نمونه تخاصمی) برای هر حمله گزارش شده است. همچنین، برای آموزش تشخیص دهنده‌ی نهایی XGBoost از ۱۰۰ نمونه سالم و تعداد مساوی نمونه‌ی تخاصمی 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 اعتبار‌سنجی ضربدری<sup>۵</sup> – لایه با جستجوی شبکه‌ای روی فرآپارامترهای XGBoost صورت گرفته است و سپس مدل‌های دارای بهترین AUC انتخاب شده اند. همچنین، در تمامی حملات، قربانی مورد حمله یک مدل دسته‌بند ۱۸ ResNet آموزش داده متناظر آموزش دیده است (دقیقاً مطابق [۴]). 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 مدل‌ها

در این بخش به بررسی نتایج بدست آمده طی آزمون مدل‌های ارائه شده و چارچوب‌های ARCANE-GAN و ARCANE-Diff از سه جنبهٔ عملکرد مدل‌های مولد، عملکرد تشخیص و در نهایت عملکرد پاکسازی حملات خواهیم پرداخت.

#### ۴-۱ سنجش عملکرد مدل‌های مولد

هر دو مدل مولد روی آموزشی split آموزش داده شدند. برای انتخاب بهترین مدل‌های مولد از معیارهای Inception Score [۷۵] Fréchet Inception Distance (FID) و (IS) [۴۵] 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 مدل‌های مولد. برای محاسبهٔ این معیار از خروجی یک دسته‌بند معروف از پیش آموزش داده شده به نام InceptionV3<sup>۱</sup> روی تعداد نسبتاً زیادی از تصاویر تولید شده توسط مدل مولد، استفاده می‌شود. مقدار IS در دو صورت بیشینه خواهد شد (حالت مطلوب):

۱. این که آنتروپی برچسب‌های پیش‌بینی شده توسط مدل InceptionV3 برای نمونه‌های تولید شده توسط مدل مولد آزمون، کمینه باشد. به عبارت دقیق‌تر، دسته‌بند InceptionV3 می‌تواند با قطعیت بالایی یک برچسب برای هر نمونه ورودی پیش‌بینی کند. به طور شهودی، در این حالت می‌توان گفت نمونه‌های تولید شده ویژگی‌های نمونه‌های واقعی را به خوبی در بر می‌گیرند، به طوری که به راحتی توسط یک دسته‌بند نسبتاً قوی، در دسته‌های متناظرشان قرار می‌گیرند.

۲. این که برچسب‌های خروجی تولید شده توسط دسته‌بند، به طور برابری در همهٔ کلاس‌های ممکن حضور داشته باشند. به عبارت دیگر، آنتروپی برچسب‌های خروجی دسته‌بند برای نمونه‌های تولید شده نسبت به کل دسته‌های ممکن، بیشینه باشد. به طور شهودی، در این حالت می‌توان انتظار داشت که نمونه‌های تولید شده دارای تنوع خوبی هستند.

**FID:** این معیار فاصلهٔ فریشه<sup>۲</sup> بین توزیع بردار ویژگی‌های استخراج شده توسط دسته‌بند InceptionV3 از مجموعه‌دادهٔ واقعی و نمونه‌های ساختگی تولید شده توسط یک مدل مولد را محاسبه می‌کند. از آنجایی که این یک معیار فاصله است، بنابراین مقادیر کمتر برای آن نمایانگر نزدیک‌تر بودن توزیع مجموعه داده

<sup>1</sup><https://en.wikipedia.org/wiki/Inceptionv3>

<sup>2</sup>Fréchet

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

CIFAR10			Tiny-ImageNet			
↑IS	↓FID	# Training Steps	↑IS	↓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 نمونه های ساختگیست و بنابراین مطلوب ماست. اگر بتوان توزیع آماری ویژگی های استخراج شده توسط دسته بند را برای نمونه های حقیقی و ساختگی توسط دو گاووسی چند متغیره مدل کرد، برای محاسبه فاصله های فرشه فرم بسته هی (۱-۴) موجود است:

$$d_F(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\mu', \Sigma')) = \|\mu - \mu'\|_2^2 + \text{tr} \left( \Sigma + \Sigma' - 2(\Sigma \Sigma')^{\frac{1}{2}} \right) \quad (1-4)$$

بنابراین مقدار FID معمولاً زمانی کاهش پیدا می کند که نمونه های تولید شده توسط مدل مولد - نسبت به مجموعه داده های اصلی - از تنوع خوبی برخوردار باشند.

در عمل، با وجود این که سنجه FID به عنوان جایگزینی برای IS معرفی شد، این دو سنجه اکنون با هم برای سنجش عملکرد مدل ها به کار می روند. به عنوان یک قاعده هی کلی مدل های که دارای FID کمی هستند نمونه های متنوع تری تولید می کنند در حالی که مدل هایی که دارای IS بالا هستند، نمونه های با کیفیت تری در هر کلاس تولید خواهند کرد. به علاوه، سنجه FID برای برقراری مقایسه های بین مجموعه داده های واقعی و نمونه های ساختگی به کار می رود در حالی که سنجه IS یک آنالیز آماری صرف فقط روی نمونه های یک مجموعه از داده هاست (ممولاً نمونه های ساختگی) [۶۵].

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

## ۲-۳-۴ سنجش عملکرد تشخیص حملات

یکی از مهم ترین سنجه هایی که برای تسلک های تشخیص مورد استفاده قرار می گیرد، سطح زیر نمودار<sup>۱</sup> Receiver Operating Characteristic (ROC) است. برای تطبیق نتایج به دست آمده با ACGAN-ADA از سطح زیر نمودار ACGAN-ADA (pAUC-۰.۰۰) استفاده شده است. علت ذکر شده برای این انتخاب در مقاله ای اهمیت بیشتر تشخیص های صورت گرفته در نواحی ای از نمودار است که نرخ مثبت کاذب<sup>۲</sup> کم می باشد، چرا که نرخ مثبت کاذب کم در عمل برای یک تشخیص دهنده بسیار مطلوب است و نواحی دیگر نمودار ROC در

<sup>۱</sup> Area Under Curve

<sup>۲</sup> False Positive

<sup>۳</sup> FPR (False Positive Rate)

مقام یک سیستم دفاعی، از اهمیت ناچیزی برخوردارند [۴]. 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:

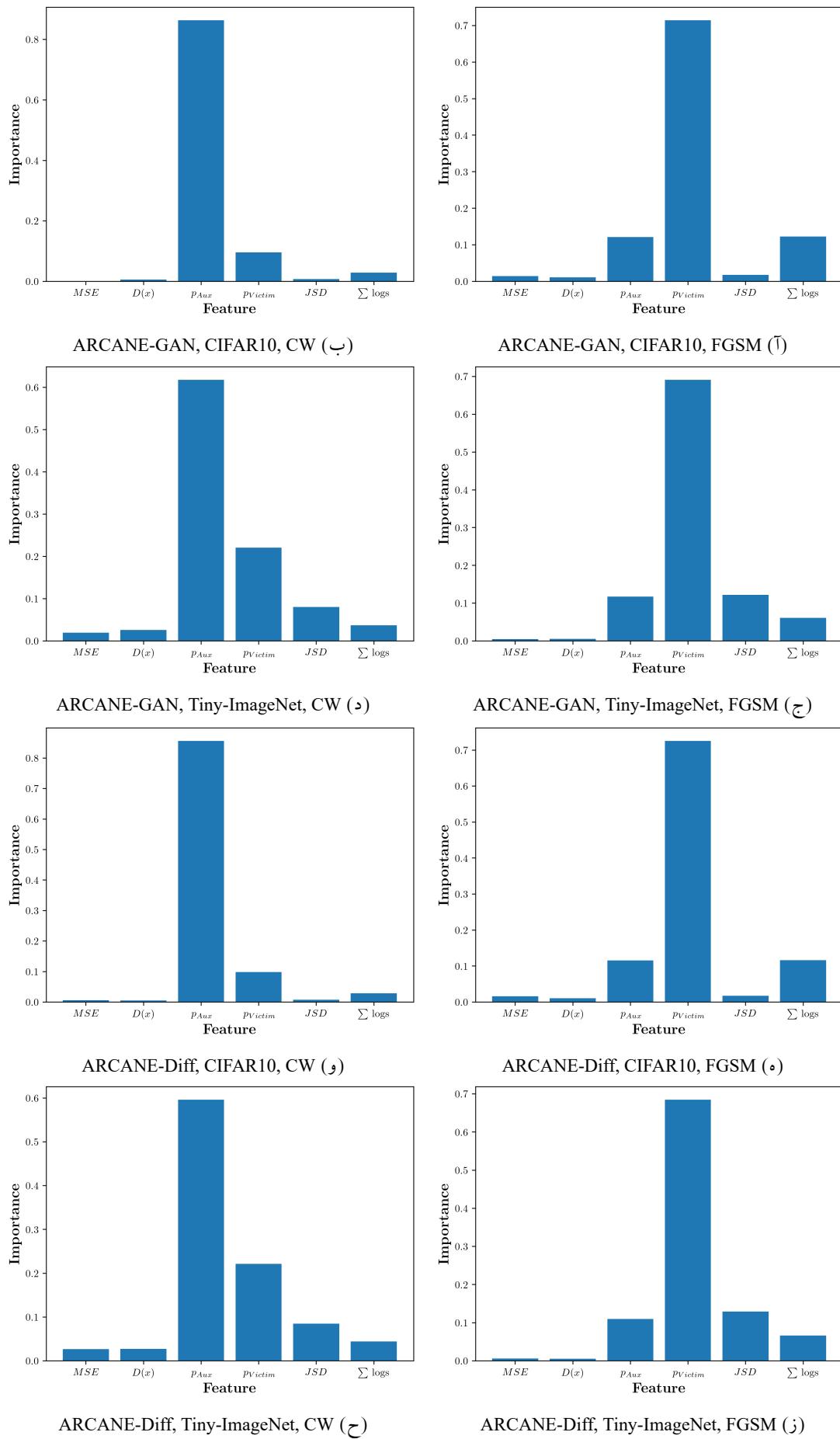
$$pAUC - U = \int_{x=0}^U ROC(x)dx$$

. مشخص است که در این حالت بیشینه‌ی مقدار ممکن برای  $pAUC-0.2$  برابر همان ۰.۲ خواهد بود چرا که در بهترین حالت، این مقدار برابر مساحت مستطیلی با اضلاع ۰.۲ و ۱ خواهد بود که از نقطه‌ی بالا سمت چپ نمودار ROC عبور می‌کند.

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

جدول ۲-۴: عملکرد تشخیص ARCANE-Diff و ARCANE-GAN طبق معیار ۰.۲ pAUC

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



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

### ۳-۳-۴ 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 پیش‌بینی شده توسط مدل قربانی قبل و پس از پاکسازی برابر است.

۲. نمونه تخاصمی بوده و برچسب پیش‌بینی شده توسط مدل قربانی پس از پاکسازی یا با برچسب پیش‌بینی شده یا با برچسب واقعی نمونه‌ی سالم متناظر با این نمونه تخاصمی برابر است.

از آنجایی که مجموعه داده‌های آماده شده برای آزمون عملکرد پاکسازی مدل از تعدادی نمونه‌ی سالم به همراه نمونه‌های تخاصمی متاظرshan به همان تعداد تشکیل شده است، بنابراین، این مجموعه داده‌ها متعادل بوده و سنجه‌ی دقت در آزمون عملکرد پاکسازی با معنیست.

به عنوان مثال در شکل ۲-۴ دو نمونه از پاکسازی‌های انجام شده توسط ARCANE-GAN و ARCANE-Diff روی یک نمونه از مجموعه داده‌ی CIFAR10 با برچسب واقعی "اسب" و برچسب تخاصمی "قورباغه" آمده است.

همچنین در جدول ۳-۴ نتایج مربوط به پاکسازی برای ARCANE-GAN و ARCANE-Diff آمده است.

جدول ۳-۴: دقت پاکسازی ARCANE-GAN و ARCANE-Diff

Tiny-ImageNet		CIFAR10	
CW	FGSM	CW	FGSM
0.861	0.784	0.915	0.835
0.953	0.8423	0.965	0.8475

ARCANE-GAN      ARCANE-Diff



(ب) پاکسازی توسط ARCANE-GAN



(آ) پاکسازی توسط ARCANE-Diff

شکل ۴-۲: نمونه‌هایی از پاکسازی‌های انجام شده توسط ARCANE-Diff و ARCANE-GAN. در هر تصویر شکل سمت چپ نمونه‌ی سالم، شکل وسط نمونه تخاصمی و شکل سمت راست نمونه پاکسازی شده را نشان می‌دهد.

۴ - ۴ نتایج مقایسه و بررسی

در ابتدا به بررسی عملکرد تشخیص حملات توسط ARCANE خواهیم پرداخت. نتایج ARCANE در برابر سایر روش‌ها روی مجموعه داده CIFAR10 در جدول ۴-۴ آمده است که در آن بهترین نتایج به صورت برجسته و دومین بهترین نتایج به صورت زیر خط دار آمده‌اند. همانطور که مشاهده می‌شود، ARCANE می‌تواند تمامی روش‌های ارائه شده را با اختلاف قابل توجهی بهبود دهد. همچنین، عملکرد تشخیص ARCANE-GAN و ARCANE-Diff در تشخیص حملات بسیار نزدیک بوده ولی روی حمله FGSM، روش ARCANE-GAN پیشین یعنی عملکرد بهتری از خود نشان می‌دهد. به طور کلی چارچوب ARCANE بهترین روش تشخیص پیشین است. ACGAN-ADA را روی مجموعه داده CIFAR10 با 6.27% و 2.58% بهبود برای حملات به ترتیب CW و FGSM شکست داده است.

جدول ۴-۴: مقایسه عملکرد پاکسازی ARCANE با سایر روش‌ها روی مجموعه داده CIFAR10

CW	FGSM	
0.0576	0.0566	[FF] f-AnoGAN
0.0533	0.1642	[FV] KD
0.1042	0.1783	[D0] MD
0.0910	0.0436	[D1] ODDS
0.1489	0.1388	[D2] SID
0.1593	0.1782	[FA] ADA
<u>0.1881</u>	0.1819	<sup>1</sup> [F] ACGAN-ADA
<b>0.1999</b>	<b>0.1866</b>	ARCANE-GAN
<b>0.1999</b>	<u>0.1856</u>	ARCANE-Diff

در ادامه مطابق آزمایش های انجام شده در [۴] برای آزمون نتیجه های تشخیص روی Tiny-ImageNet فقط روش های ARCANE، ACGAN-ADA و f-AnoGAN با MD، ADA، KD مقایسه شده اند. نتایج این مقایسه در جدول ۵-۴ آمده است.

مطابق انتظار، از آنجایی که یکی از اهداف اصلی ARCANE حل مشکل مجموعه داده‌ها با تعداد کلاس‌های زیاد است، اختلاف ARCANE با بهترین روش پیشین ACGAN-ADA در مجموعه داده‌ی ۲۰۰ کلاسه‌ی-Tiny

جدول ۴-۵: مقایسه عملکرد پاکسازی ARCANE با سایر روش‌ها روی مجموعه داده‌ی Tiny-ImageNet

CW	FGSM	
0.0571	0.0655	[FF] f-AnoGAN
0.0542	0.1168	[FV] KD
0.0918	0.1104	[D.] MD
0.1312	0.1385	[FA] ADA
0.1532	0.1496	[F] ACGAN-ADA
<b>0.1913</b>	<u>0.1985</u>	ARCANE-GAN
<u>0.1912</u>	<b>0.1986</b>	ARCANE-Diff

<sup>۱</sup> برای ACGAN-ADA از بین چندین روش ارائه شده در مقاله، نتیجه‌ی بهترین روش در اینجا گذارش شده است.

مشهود تر است. در این مجموعه داده، ARCANE توانسته است روش ACGAN-ADA را با 24.87% ImageNet و 32.75% FGSM شکست دهد.

در ادامه به مقایسه نتایج پاکسازی حملات خواهیم پرداخت. مطابق [۴]، مقایسه‌ای بین نتایج پاکسازی سه چارچوب Defense-GAN، ACGAN-ADA و ARCANE روی مجموعه داده CIFAR10 و تحت حمله CW در جدول ۶-۴ به عمل آمده است.<sup>۱</sup> همانطور که از جدول بر می‌آید و مطابق انتظار، روش ARCANE-Diff که مجهز به یک مدل انتشاری شرطیست با اختلاف ۷۹.۱۱ درصدی نسبت به pix2pix منجر به دقت پاکسازی بهتری می‌شود. علت عملکرد بهتر ARCANE-Diff نسبت به ARCANE-GAN احتمالاً روش متفاوت ARCANE-Diff و نیز توانایی ذاتی مدل‌های انتشاری در از بین بردن نویز به واسطه طراحی و نحوه آموزش آنها در نظر گرفت. همچنین به نظر می‌رسد که رابطه مستقیمی بین عملکرد مولد در تولید تصاویر واقعی و متنوع، و دقت پاکسازی چارچوب‌هایی که از آن مولد به عنوان عامل پاک کننده استفاده می‌کنند، وجود دارد.

جدول ۶-۶: مقایسه نتایج پاکسازی حملات روی مجموعه داده CIFAR10 و حمله CW

Purification Accuracy	
0.3274	[۳۳] Defense-GAN
0.8423	[۴] ACGAN-ADA
0.8632	<sup>۲</sup> [۷۶] pix2pix
0.875	ARCANE-GAN
<b>0.965</b>	<b>ARCANE-Diff</b>

<sup>۱</sup> به دلیل پر هزینه بودن انجام عملیات پاکسازی روی مجموعه داده Tiny-ImageNet نتایج روش‌های دیگر روی این مجموعه داده موجود نبوده و بنابراین در این جدول مقایسه گزارش نشده است. برای دیدن نتایج پاکسازی ARCANE روی Tiny-ImageNet می‌توانید به جدول ۳-۴ مراجعه کنید.

<sup>۲</sup> روش pix2pix همان روش ACGAN-ADA است که به جای ACGAN در آن از یک مدل مولد تخصصی شرطی قوی‌تر به نام pix2pix استفاده شده است.

## فصل پنجم

### نتیجه‌گیری و پیشنهادها

#### ۱-۵ مقدمه

در بخش انتهایی این گزارش ابتدا در بخش نتیجه‌گیری به یک جمع بندی کلی از اهداف محقق شده در طی این پژوهش خواهیم رسید و سپس در بخش پیشنهادها و کارهای آینده، جهت‌های احتمالی تحقیقات آتی بیان خواهند شد.

#### ۲-۵ نتیجه‌گیری

در این پژوهش چارچوب جدیدی برای دفاع در برابر حملات تخاصمی تحت عنوان ARCANE بر مبنای روش پیشین ACGAN-ADA ارائه شد. این روش روی دو مجموعه داده‌ی مطرح CIFAR10 و Tiny-ImageNet و تحت دو حمله‌ی تخاصمی FGSM و CW آزموده شد. در نتیجه‌ی این آزمایش‌ها نشان داده شد که ARCANE می‌تواند هم در تشخیص و هم در پاکسازی، بهترین نتایج پیشین را با اختلاف زیادی شکست دهد و به طور متوسط در تشخیص و پاکسازی به ترتیب 16.62% و 11.8% حملات بهتر از ACGAN-ADA عمل کرده است.

### ۳-۵ پیشنهادها و کارهای آینده

یکی از نقاط ضعف اصلی ARCANE در حال حاضر، عدم توانایی آن در انجام عملیات تشخیص و پاکسازی حملات به صورت بر خط<sup>۱</sup> است. با توجه به نتایج بدست آمده از اهمیت ویژگی‌های مورد استفاده در نتیجه‌ی تصمیم‌گیرنده‌ی نهایی XGBoost (شکل ۴-۱)، با کنار گذاشتن ویژگی MSE که عمدی زمان پردازش را به خود اختصاص می‌دهد، احتمالاً<sup>۲</sup> می‌توان به نتایجی مشابه در تشخیص ولی به مراتب سریع‌تر دست پیدا کرد. از طرف دیگر، با گسترش روز افزون تحقیقات انجام شده روی مدل‌های انتشاری و روش‌های بهینه‌تر حل معادلات دیفرانسیل معمولی (ODE<sup>۳</sup>) برای نمونه‌گیری از این مدل‌ها در زمان کمتر (مانند [۶۳])، به نظر می‌رسد که عملیات پاکسازی نیز می‌تواند در زمان کوتاه‌تری با هدر رفت حداقل دقت، انجام بپذیرد.

به عنوان یکی از اهداف احتمالی پژوهش‌های آتی، استفاده از معیار شباht LPIPS<sup>۴</sup> [۷۷] به جای معیار MSE پر استفاده‌ی MSE پیشنهاد می‌شود. با وجود آن که محاسبه‌ی این معیار به مراتب هزینه‌بر تر از محاسبه‌ی MSE است، به صورت تجربی نشان داده شده است که LPIPS به تصور یک انسان نزدیک تر است و خصوصاً نسبت به تصاویر مات شده (که همچنان MSE های کوچکی با تصویر مرجع دارند) حساس تر است. علاوه بر آن، این معیار، مانند MSE مشتق پذیر است و بنابراین می‌توان مستقیماً از آن به عنوان تابع هزینه‌ی بهینه سازی ARCANE-GAN در زمان پاکسازی استفاده کرد.

<sup>1</sup>Online

<sup>2</sup>Ordinary Differential Equation

<sup>3</sup>Learned Perceptual Image Patch Similarity

# 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
Area Under Curve .....	سطح زیر نمودار
Attack Surface .....	سطح حمله
Attacker .....	مهاجم
Auto Trader .....	تاجر خودکار
Autoencoder .....	خودرمندگنگار
Auxiliary Classifier .....	دسته‌بند اضافی

## B

Backpropagation .....	پس انتشار
-----------------------	-----------

Binary Search .....	جستجوی دودویی .....
Black Box .....	جعبه سیاه .....
Box Constraints .....	محدودیت های جعبه‌ای .....

**C**

Cascade .....	آبشاری .....
Class Conditional .....	مشروط بر کلاس .....
Classifier Guided Diffusion Model .....	مدل انتشاری هدایت شده با دسته‌بند .....
Classifiers .....	دسته‌بند ها .....
Confidence .....	اطمینان .....
Confidence Reduction .....	تقلیل اطمینان .....
Cross-validation .....	اعتبار سنجی ضربدری .....
Cyber Attacks .....	حملات سایبری .....

**D**

Deep Learning .....	یادگیری عمیق .....
Defensive Distillation .....	تقطیر دفاعی .....
Denoising Diffusion Probabilistic Model (DDPM) .....	مدل‌های انتشار احتمالی حذف کننده نویز .....
Detection .....	تشخیص .....
Diffusion Model .....	مدل انتشاری .....
Diffusion Model with Classifier-Free Guidance .....	مدل انتشاری با هدایت بدون دسته‌بند .....
Discriminator .....	ممیز .....
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 .....	فریشه
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 شبکه ای

**H**

Hidden Layers .....	لایه های پنهان
Histogram Matching .....	تطبیق هیستوگرام
Hyperparameter .....	فرآپارامتر

**J**

Joint .....	توأم
-------------	------

**K**

Kernel Density Estimation .....	تخمین چگالی هسته ای
---------------------------------	---------------------

**L**

Latent .....	نهفته
--------------	-------

Learning Rate ..... نرخ یادگیری

Loss ..... ضرر

## M

Mahalanobis Distance ..... فاصله ماھالانوبیس

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 ..... تعادل نش

Noise ..... نویز

Non-equilibrium Thermodynamics ..... ترمودینامیک غیرتعادلی

Norm ..... نرم

## O

Online ..... برخط

## P

Pipeline ..... خط لوله

Pixel ..... پیکسل

Poisoning Attack ..... حمله مسموم‌کننده

Posterior ..... پسین

Prior ..... پیشین

Projected Gradient Descent ..... نزول گرادیان افکننده

Purification ..... پاکسازی

Python ..... پایتون

## R

Random Initialization .....	آغاز تصادفی
Reconstruction Error .....	خطای بازسازی
Reference .....	مرجع
Reformer .....	بهساز
Regularization .....	نظم‌سازی
Reverse Diffusion .....	انتشار معکوس

## S

Self-driving Car .....	ماشین خودران
Sentiment Analysis .....	تحلیل احساسات
Slack Variable .....	متغیر لنگی

## T

Tabular .....	جدولی
Targeted Misclassification .....	دسته‌بندی اشتباه هدفمند
Tensor .....	تنسور
Threat Models .....	مدل‌های تهدید
Tolerance .....	تلورانس
Transposed Convolution .....	کانولوشن ترانهاده

## U

Unsupervised Learning .....	یادگیری بدون نظارت
Untargeted Misclassification .....	دسته‌بندی اشتباه غیر هدفمند

## V

Vanishing Gradient ..... گرادیان محو شونده

Victim ..... قربانی .....

**W**

White Box ..... جعبه سفید .....

**Z**

Zero-sum ..... مجموع-صفر .....

# واژه‌نامه فارسی به انگلیسی

۱

Cascade .....	آبشاری .....
Random Initialization .....	آغاز تصادفی .....
Adversarial Training .....	آموزش تخاصصی .....
Adversarial Perturbation .....	اختلال تخاصصی .....
Confidence .....	اطمینان .....
Cross-validation .....	اعتبار سنجی ضربه‌دری .....
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

Online .....	بر خط .....
Reformer .....	بهساز .....

## پ

Purification .....	پاکسازی .....
Adversarial Purification .....	پاکسازی تخاصمی .....
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 غیرتعادلی .....
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 .....	تنسور .....
Joint .....	توأم .....

## ج

Tabular .....	جدولی .....
Binary Search .....	جستجوی دودویی .....

Grid Search .....	جستجوی شبکه‌ای .....
White Box .....	جبهه سفید .....
Black Box .....	جبهه سیاه .....

**ح**

Adversarial Attacks .....	حملات تخاصمی .....
Cyber Attacks .....	حملات سایبری .....
Exploratory Attack .....	حمله اکتشافی .....
Evasion Attack .....	حمله گریزانه .....
Poisoning Attack .....	حمله مسموم‌کننده .....

**خ**

Pipeline .....	خط لوله .....
Reconstruction Error .....	خطای بازسازی .....
Manifold .....	خمینه .....
Autoencoder .....	خودرمزنگذار .....

**د**

Classifiers .....	دسته بند ها .....
Auxiliary Classifier .....	دسته‌بند اضافی .....
Untargeted Misclassification .....	دسته‌بندی اشتباه غیر هدفمند .....
Targeted Misclassification .....	دسته‌بندی اشتباه هدفمند .....
Divergence .....	دیورژانس .....

**ز**

Markov Chain .....	زنگیره مارکوف .....
--------------------	---------------------

**س**

Attack Surface .....	سطح حمله .....
----------------------	----------------

Area Under Curve .....	سطح زیر نمودار.....
Metric .....	سنجه.....
ش	
Generative Adversarial Network (GAN) .....	شبکه مولد تخاصمی .....
ض	
Loss .....	ضرر.....
ف	
Mahalanobis Distance .....	فاصله ماهالانوبیس .....
Hyperparameter .....	فرآپارامتر .....
Fréchet .....	فریشه .....
Mode Collapse .....	فروپاشی مُد .....
Feature Space .....	فضای ویژگی .....
ق	
Victim .....	قربانی .....
ک	
Fully Connected .....	کاملاً متصل .....
Transposed Convolution .....	کانولوشن ترانهاده .....
Min-max .....	کمینه-بیشینه .....
گ	
Vanishing Gradient .....	گرادیان محو شونده .....
ل	
Hidden Layers .....	لایه‌های پنهان .....

## 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 .....	مدل‌های مولد .....
Denoising Diffusion Probabilistic Model (DDPM) .....	مدل‌های انتشار احتمالی حذف کننده نویز .....
Threat Models .....	مدل‌های تهدید .....
Reference .....	مرجع .....
Class Conditional .....	مشروط بر کلاس .....
Discriminator .....	ممیز .....
Regularization .....	نظم‌سازی .....
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 .....

هوش مولد ..... Generative AI

## ۵

یادگیری بدون نظارت ..... Unsupervised Learning

یادگیری عمیق ..... Deep Learning



## 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] Wang, H., Miller, D. J., and Kesidis, G., “Anomaly Detection of Adversarial Examples using Class-conditional Generative Adversarial Networks”, May 2022.
- [5] Kang, M., Shim, W., Cho, M., and Park, J., “Rebooting ACGAN: Auxiliary Classifier GANs with Stable Training”, Nov. 2021.
- [6] Ho, J., Jain, A., and Abbeel, P., “Denoising Diffusion Probabilistic Models”, Dec. 2020.
- [7] Chakraborty, A., Alam, M., Dey, V., Chattopadhyay, A., and Mukhopadhyay, D., “Adversarial Attacks and Defences: A Survey”, Sept. 2018.
- [8] Costa, J. C., Roxo, T., Proen  a, H., and In  cio, P. R. M., “How Deep Learning Sees the World: A Survey on Adversarial Attacks & Defenses”, May 2023.
- [9] 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.
- [10] Ren, K., Zheng, T., Qin, Z., and Liu, X., “Adversarial Attacks and Defenses in Deep Learning”, *Engineering*, Vol. 6, pp. 346–360, Mar. 2020.
- [11] Sun, L., Tan, M., and Zhou, Z., “A survey of practical adversarial example attacks”, *Cybersecurity*, Vol. 1, p. 9, Dec. 2018.
- [12] Goodfellow, I. J., Shlens, J., and Szegedy, C., “Explaining and Harnessing Adversarial Examples”, Mar. 2015.

- [13] 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.
- [14] 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.
- [15] Zhou, M., Gao, X., Wu, J., Liu, K., Sun, H., and Li, L., “Investigating White-Box Attacks for On-Device Models”, Mar. 2024.
- [16] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R., “Intriguing properties of neural networks”, Feb. 2014.
- [17] Carlini, N. and Wagner, D., “Towards Evaluating the Robustness of Neural Networks”, Mar. 2017.
- [18] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A., “Towards Deep Learning Models Resistant to Adversarial Attacks”, Sept. 2019.
- [19] Kingma, D. P. and Ba, J., “Adam: A Method for Stochastic Optimization”, Jan. 2017.
- [20] 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.
- [21] Zhang, D., Zhang, T., Lu, Y., Zhu, Z., and Dong, B., “You Only Propagate Once: Accelerating Adversarial Training via Maximal Principle”, Nov. 2019.
- [22] Zhang, J., Zhu, J., Niu, G., Han, B., Sugiyama, M., and Kankanhalli, M., “Geometry-aware Instance-reweighted Adversarial Training”, May 2021.
- [23] Wong, E., Rice, L., and Kolter, J. Z., “Fast is better than free: Revisiting adversarial training”, Jan. 2020.
- [24] Gu, S. and Rigazio, L., “Towards Deep Neural Network Architectures Robust to Adversarial Examples”, Apr. 2015.
- [25] Papernot, N., McDaniel, P., Wu, X., Jha, S., and Swami, A., “Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks”, Mar. 2016.
- [26] Zi, B., Zhao, S., Ma, X., and Jiang, Y.-G., “Revisiting Adversarial Robustness Distillation: Robust Soft Labels Make Student Better”, Aug. 2021.
- [27] Cui, J., Liu, S., Wang, L., and Jia, J., “Learnable Boundary Guided Adversarial Training”, Aug. 2021.
- [28] Chen, E.-C. and Lee, C.-R., “LTD: Low Temperature Distillation for Robust Adversarial Training”, June 2023.
- [29] 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.
- [30] Gao, J., Wang, B., Lin, Z., Xu, W., and Qi, Y., “DeepCloak: Masking Deep Neural Network Models for Robustness Against Adversarial Samples”, Apr. 2017.
- [31] Liu, X., Cheng, M., Zhang, H., and Hsieh, C.-J., “Towards Robust Neural Networks via Random Self-ensemble”, July 2018.
- [32] 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.
- [33] Samangouei, P., Kabkab, M., and Chellappa, R., “Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models”, May 2018.
- [34] 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.

- [35] 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.
- [36] Nie, W., Guo, B., Huang, Y., Xiao, C., Vahdat, A., and Anandkumar, A., “Diffusion Models for Adversarial Purification”, May 2023.
- [37] Gong, Z., Wang, W., and Ku, W.-S., “Adversarial and Clean Data Are Not Twins”, Apr. 2017.
- [38] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A., “The Limitations of Deep Learning in Adversarial Settings”, Nov. 2015.
- [39] Metzen, J. H., Genewein, T., Fischer, V., and Bischoff, B., “On Detecting Adversarial Perturbations”, Feb. 2017.
- [40] Li, X. and Li, F., “Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics”, Oct. 2017.
- [41] Saberian, M. J. and Vasconcelos, N., “Boosting Classifier Cascades”, *Advances in Neural Information Processing Systems*, Vol. 23, pp. 2047–2055, 2010.
- [42] 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.
- [43] Yang, P., Chen, J., Hsieh, C.-J., Wang, J.-L., and Jordan, M. I., “ML-LOO: Detecting Adversarial Examples with Feature Attribution”, June 2019.
- [44] 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.
- [45] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X., “Improved Techniques for Training GANs”, June 2016.
- [46] Odena, A., Olah, C., and Shlens, J., “Conditional Image Synthesis With Auxiliary Classifier GANs”, July 2017.
- [47] Feinman, R., Curtin, R. R., Shintre, S., and Gardner, A. B., “Detecting Adversarial Samples from Artifacts”, Nov. 2017.
- [48] 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.
- [49] Kullback, S. and Leibler, R. A., “On Information and Sufficiency”, *The Annals of Mathematical Statistics*, Vol. 22, pp. 79–86, Mar. 1951.
- [50] Lee, K., Lee, K., Lee, H., and Shin, J., “A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks”, Oct. 2018.
- [51] Roth, K., Kilcher, Y., and Hofmann, T., “The Odds are Odd: A Statistical Test for Detecting Adversarial Examples”, May 2019.
- [52] Tian, J., Zhou, J., Li, Y., and Duan, J., “Detecting Adversarial Examples from Sensitivity Inconsistency of Spatial-Transform Domain”, Mar. 2021.
- [53] 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.
- [54] 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.
- [55] Saxena, D. and Cao, J., “Generative Adversarial Networks (GANs): Challenges, Solutions, and Future Directions”, *ACM Computing Surveys*, Vol. 54, pp. 1–42, Apr. 2022.

- [56] 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.
- [57] Mounjid, O. and Guo, X., “Convergence of GANs Training: A Game and Stochastic Control Methodology”, Dec. 2021.
- [58] Mirza, M. and Osindero, S., “Conditional Generative Adversarial Nets”, Nov. 2014.
- [59] Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S., “Deep Unsupervised Learning using Nonequilibrium Thermodynamics”, Nov. 2015.
- [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 Borovcak, 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.
- [76] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A., “Image-to-Image Translation with Conditional Adversarial Networks”, Nov. 2018.
- [77] Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O., “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric”, Apr. 2018.

# ARCANE: 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](mailto:manshaei@ece.iut.ac.ir))

## Abstract

The ever-increasing use of artificial intelligence and specifically deep learning decision makers in our everyday lives necessitates a need to combat potential cyber-attacks deployed against such systems. An important category of such cyber-attacks, dubbed "Adversarial Attacks", alter the outputs of a deep learning-based decision maker by making near-invisible perturbations to its inputs. In this study a framework called ARCANE using class-conditional generative models is presented that aims to provide a novel means of defense against adversarial attacks. Experimental results show that ARCANE is able to outperform previous SOTA frameworks by a margin of 16.62% and 11.8% on adversarial detection and purification tasks, respectively.

**Key Words:** Artificial Intelligence, Deep Learning, Generative Models, Adversarial Robustness, Adversarial Attacks



**Isfahan University of Technology**

Department of Electrical and Computer Engineering

# **ARCANE: 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, 2024

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

Dr. Behzad Nazari, Department Graduate Coordinator

