



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 آفای آرین تشکر

تحت عنوان

آرکین: دفاع در برابر حملات تخاصمی با استفاده از مدل های مولد مشروط بر کلاس

در تاریخ ۹۰/۰۴/۱۴۰۳ توسط کمیته تخصصی زیر مورد بررسی و تصویب نهایی قرار گرفت:

دکتر محمد حسین منشی

۱- استاد راهنمای پایان نامه

دکتر علیرضا بصیری

۲- استاد داور

دکتر عبدالرضا میرزا بی

۳- 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	۱-۲-۲-۲ سطح حمله
۱۲	۱-۲-۲-۲ FGSM
۱۳	۱-۲-۲-۲ CW
۱۶	۱-۲-۲-۲ PGD
۱۶	۳-۲ روش‌های دفاع در برابر حملات تخصصی
۱۷	۱-۳-۲ روش‌های پاکسازی حمله
۲۵	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
۵۰	.....	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
۶۰	.....	۳-۴ نتایج آزمون مدل ها
۶۰	.....	۴-۱ سنجش عملکرد مدل های مولد
۶۱	.....	۴-۲ سنجش عملکرد تشخیص حملات
۶۴	.....	۴-۳ سنجش عملکرد پاکسازی حملات
۶۵	.....	۴-۴ بررسی و مقایسه نتایج
۶۶	.....	۵-۴ مطالعات جانی
۶۷	.....	۵-۵-۱ تاثیر نوع دسته بند در دقیقت تشخیص وجود حمله
۷۱	.....	۵-۵-۲ تاثیر ویژگی های استفاده شده در بهبود دقیقت تشخیص

### فصل پنجم: نتیجه گیری و پیشنهادها

۷۳	.....	۱-۵ مقدمه
۷۳	.....	۲-۵ نتیجه گیری
۷۴	.....	۳-۵ پیشنهادها و کارهای آینده
۷۵	.....	واژه نامه انگلیسی به فارسی
۸۱	.....	واژه نامه فارسی به انگلیسی
۸۷	.....	مراجع
۹۳	.....	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
۱۵	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 نظری یک DDPM	۲۷-۲
۴۵	.....	نمونه‌های تولید شده توسط DDPM	۲۸-۲
۴۶	.....	۲۹-۲ نمونه‌های تولید شده توسط Classifier Guided Diffusion Model	
۴۸	.....	۳۰-۲ نمونه‌های تولید شده توسط Classifier-Free Guided Diffusion Model	
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	
۶۳	.....	۱-۴ اهمیت ویژگی‌های استفاده شده در XGBoost	
۶۴	.....	۲-۴ نمونه‌ای از پاکسازی‌های انجام شده توسط ARCANE	
۷۰	.....	۳-۴ تاثیر نوع دسته‌بند در تشخیص وجود حمله در ARCANE-Diff	
۷۰	.....	۴-۴ تاثیر نوع دسته‌بند در تشخیص وجود حمله در ARCANE-GAN	

## فهرست جداول

۱۰	۱-۲ مقایسه حملات جعبه سیاه و جعبه سفید
۱۷	۲-۲ مقایسه حملات CW و PGD, FGSM
۶۱	۱-۴ عملکرد مدل های مولد روی CIFAR10 و Tiny-ImageNet
۶۲	۲-۴ عملکرد تشخیص ARCANE-Diff و ARCANE-GAN طبق معیار pAUC-0.2
۶۴	۳-۴ دقت پاکسازی ARCANE-GAN و ARCANE-Diff
۶۵	۴-۴ مقایسه عملکرد تشخیص ARCANE با سایر روش ها روی مجموعه داده CIFAR10
۶۶	۵-۴ مقایسه عملکرد تشخیص ARCANE با سایر روش ها روی مجموعه داده Tiny-ImageNet
۶۶	۶-۴ مقایسه نتایج پاکسازی حملات روی مجموعه داده CIFAR10 و حمله CW
۶۹	۷-۴ تاثیر نوع دسته بند در دقت نهایی تشخیص ARCANE-Diff
۶۹	۸-۴ تاثیر نوع دسته بند در دقت نهایی تشخیص ARCANE-GAN
۷۲	۹-۴ تاثیر ویژگی های استفاده شده برای تشخیص حملات بر میزان دقت در ARCANE-Diff
۷۲	۱۰-۴ تاثیر ویژگی های استفاده شده برای تشخیص حملات بر میزان دقت در ARCANE-GAN

# 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  
DDPM ..... Denoising Diffusion Probabilistic Model

## F

FGSM ..... Fast Gradient Sign Method  
FID ..... Fréchet Inception Distance

## G

GAN ..... Generative Adversarial Network  
GDA ..... Gaussian Discriminant Analysis  
GDM ..... Guided Diffusion Model  
GI-AT ..... Geometry-aware Instance-reweighted Adversarial Training

GMM	.....	Gaussian Mixture Model
<b>I</b>		
IS	.....	Inception Score
<b>J</b>		
JS	.....	Jensen-Shannon
<b>K</b>		
KL	.....	Kullback-Leibler
<b>L</b>		
LOO	.....	Leave-One-Out
LPIPS	.....	Learned Perceptual Image Patch Similarity
<b>M</b>		
MD	.....	Mahalanobis Detector
MSE	.....	Mean Squared Error
<b>O</b>		
ODE	.....	Ordinary Differential Equation
<b>P</b>		
PGD	.....	Projected Gradient Descent
<b>R</b>		
ROC	.....	Receiver Operating Characteristic
<b>S</b>		
SSIM	.....	Structural Similarity Index Measure
<b>W</b>		
WGAN	.....	Wasserstein Generative Adversarial Network
<b>Y</b>		
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. در این چارچوب که ادامه‌ای بر پژوهش پیشین ACGAN-ADA<sup>۱</sup> به شمار می‌آید، علاوه بر استفاده از روش‌های نوین تشخیص و پاکسازی حملات، از مدل‌های مولد جدیدتری در راستای بهبود کاستی‌های چارچوب پیشین استفاده شده است. به طور خاص در زمینه‌ی تشخیص حملات از <sup>۲</sup> ویژگی اضافه و یک مدل تشخیص دهنده‌ی نهایی جدید بهره برده می‌شود. همچنین در زمینه‌ی پاکسازی حملات، استفاده از مدل‌های انتشاری شرطی به عنوان یک ایده‌ی جدید مورد آزمایش قرار گرفته است. نتایج تجربی نشان می‌دهند که این چارچوب می‌تواند در تشخیص و پاکسازی چنین حملات به ترتیب ۱6.62% و ۱1.8% بهتر از بهترین نتایج قبلی عمل کند.

کلمات کلیدی: ۱ - هوش مصنوعی، ۲ - یادگیری عمیق، ۳ - مدل‌های مولد<sup>۱</sup>، ۴ - دسته بند ها<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

### ۱-۱ اهمیت مسئله

امروزه یادگیری عمیق<sup>۱</sup> به عنوان ابزاری قدرت مند و بهینه برای حل گستره‌ی وسیعی از مسائل پیچیده روزمره شناخته می‌شود که حل آن‌ها با روش‌های یادگیری ماشین سنتی بسیار دشوار و بعضاً غیرممکن بود. در سال‌های اخیر، یادگیری عمیق چنان دستخوش پیشرفت‌های شگرفی شده که اکنون قادر است در گستره‌ی وسیعی از اهداف یادگیری، از عملکرد 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 روش‌هایی 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 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 الگوریتم تاجر خودکار<sup>۴</sup> آمده

<sup>1</sup>Deep Learning

<sup>2</sup>Self-driving Car

<sup>3</sup>Auto Trader



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

است که در طی آن سیستم تاجر خودکار مبتنی بر مدل‌های یادگیری ماشین چار خطا در پیش‌بینی قیمت آینده یک سهم می‌شوند.

دسته خاصی از حملات سایبری<sup>۱</sup> اعمال پذیر روی دسته‌بند های مبتنی بر یادگیری ماشین، حملات تخاصمی هستند که تمرکز اصلی این تحقیق می‌باشدند. این حملات، با افزودن مقدار اندکی نویز که به چشم غیر مسلح قابل مشاهده نیست، باعث تخریب خروجی یک دسته‌بند می‌شوند. در فصل ۲ راجع به این حملات بیشتر توضیح داده خواهد شد.

در این پژوهش چارچوبی تحت عنوان ARCANE<sup>۲</sup> برای دفاع در برابر حملات تخاصمی از دو جنبه‌ی پاکسازی و تشخیص حملات ارائه خواهد شد. این چارچوب نسبت به چارچوب ACGAN-ADA<sup>۳</sup> [۴] که پیش‌تر در همین زمینه ارائه شد از چند لحظه تفاوت دارد:

۱. استفاده از مدل‌های مولد بهبود یافته: در ARCANE از مدل‌های مولد جدیدتر مانند ReACGAN<sup>۴</sup> [۵] و نیز مدل‌های انتشاری (DDPM<sup>۵</sup> [۶]) استفاده می‌شود که می‌تواند برخی از ایرادات 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 جدید 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

<sup>1</sup>Cyber Attacks

<sup>2</sup>Adversarial Robustness using Class-conditional Generative Models

<sup>3</sup>Auxiliary Classifier Generative Adversarial Network - Anomaly Detection of Attacks

<sup>4</sup>Rebooted Auxiliary Classifier Generative Adversarial Network

<sup>5</sup>Denoising Diffusion Probabilistic Model

آستانه‌ی بدست آمده با روش‌هایی مانند جستجوی شبکه‌ای<sup>۱</sup>، در ARCANE از یک مدل XGBoost برای 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 پشت سر بگذارد و به طور متوسط در تشخیص و پاکسازی به ترتیب 16.62% و 11.8% بهتر از ACGAN-ADA عمل کرده است.

## ۲-۱ 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>۲</sup>‌های مناسب مورد بررسی قرار خواهد گرفت. در نهایت در فصل ۵ به جمع‌بندی نهایی و جهت‌های احتمالی تحقیقات آینده پرداخته خواهد شد.

---

<sup>1</sup>Grid Search  
<sup>2</sup>Metric

## ۱ - ۲ مقدمه

در این فصل به بیان مقدماتی در مورد حملات تخاصمی و روش‌های دفاع در برابر آن‌ها پرداخته خواهد شد. سپس مبانی هوش مولد و به خصوص مدل‌های مولد تخاصمی و مدل‌های انتشاری مورد بررسی قرار خواهند گرفت.

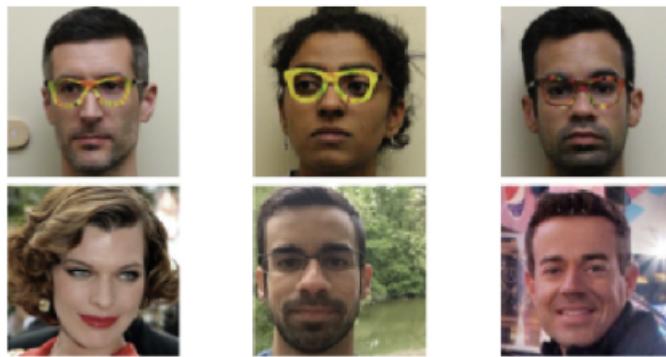
## ۲ - ۲ حملات تخاصمی

حملات تخاصمی به تکنیک‌هایی در حوزه یادگیری ماشین گفته می‌شود که برای هدف خاص فریب دادن مدل‌های مختلف به کمک توسعه و اعمال ورودی‌های آسیب زننده به این مدل‌ها طراحی شده‌اند [۷-۹]. حملات تخاصمی عموماً طی فرایندی تحت عنوان اختلال تخاصمی<sup>۱</sup> ایجاد می‌شوند. این پروسه شامل افزودن مقادیر ناچیزی نویز<sup>۲</sup> به ورودی مدل دسته‌بند قربانی<sup>۳</sup> می‌شود که با وجود نامحسوس بودن به چشم غیرمسلح، باعث اختلال در خروجی دسته‌بند 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



شکل ۲-۱: نمونه‌ی حمله تخاصمی به یک مدل تشخیص چهره [۱۰]. افراد سطر بالا به اشتباہ توسط مدل تشخیص چهره به فرد متناظر در سطر پایین تشخیص داده شده‌اند.

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

**Text:** I love these awful awf ul 80's summer camp movies. The best part about "Party Camp" is the fact that it literally literally has no No plot. The cliches cliches 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 fo0lish sexual innuendo littered throughout. This movie will make you laugh, but never intentionally. I repeat, never.

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

در حوزه تصویر می‌باشد. در این روش‌ها با تغییرات جزئی و نامحسوس روی مقادیر پیکسل<sup>۱</sup>‌های به خصوص، خروجی یک دسته‌بند تصویر تغییر خواهد کرد. با این وجود، از آنجایی که اکثر روش‌های طراحی حملات تخاصمی از ماهیت ورودی مدل به طور مستقیم برای طراحی حمله استفاده نمی‌کنند، این روش‌ها می‌توانند قابل تعیین به تمامی دسته‌بندها، چه در حوزه تصویر و چه خارج از آن، باشند [۱۱، ۸]. در شکل ۲-۲ نمونه‌ای از یک حمله تخاصمی به یک مدل تشخیص چهره را می‌توان مشاهده کرد. با قرار دادن یک عینک حاوی نویز تخاصمی روی صورت افرادی که در سطر بالا قرار دارند، مدل تشخیص چهره، چهره‌ی این افراد را به اشتباه به عنوان افراد نظری در سطر پایین تشخیص داده است. همچنین در شکل ۲-۲ نمونه‌ای از یک حمله تخاصمی به یک مدل دسته‌بند متن برای وظیفه تحلیل احساسات<sup>۲</sup> را می‌توان مشاهده کرد.

به طور رسمی یک حمله تخاصمی را می‌توان طبق تعریف ارائه شده در [۱۲] بررسی کرد. فرض کنیم که مدل دسته‌بند ( $f$ ) را در اختیار داشته باشیم که خروجی آن روی یک نمونه ورودی  $x$  یک توزیع آماری روی تمامی کلاس‌های ممکن باشد. کلاس تشخیص داده شده توسط این مدل در معادله (۱-۲) نمایش داده شده است.

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

<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 سطح 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 را در زمان استفاده از یک مدل یادگیری ماشین که چنین خط لوله‌ای از آن‌ها تشکیل می‌شود، 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.

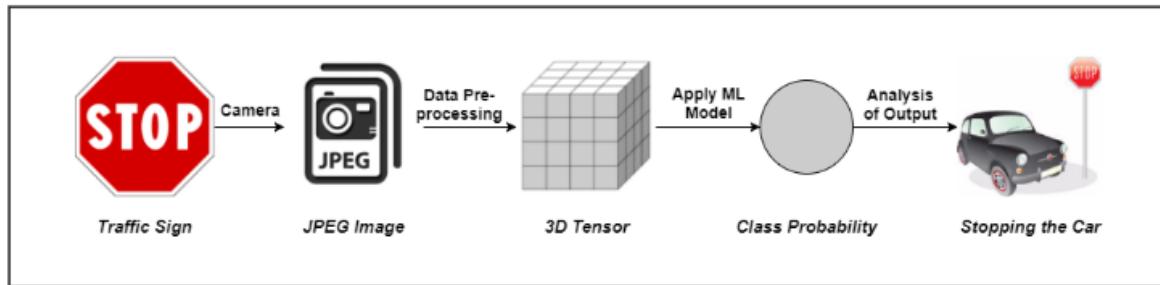
<sup>1</sup> Adversarial Sample

<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 نقاط مختلف 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 گریز 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 دچار تغییرات نامطلوب نماید. واضح است که در این دسته از حملات فرض شده است که مجموعه داده آموزشی مدل قربانی در دسترس مهاجم قرار دارد.

<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 راجع به الگوریتم یادگیری استفاده شده در سیستم مورد حمله و الگوهای احتمالی موجود در مجموعه داده های آموزشی را دریافت کند.

برای تعریف مدل تهدید نیازمند آن هستیم که نوع دسترسی مهاجم به قربانی را نیز در نظر بگیریم [۸، ۷]. 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

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

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

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  $\hat{y} \neq y_{pred}$  بر می‌آید.

۳. **دسته‌بندی اشتباه هدفمند**<sup>۴</sup>: در این حالت، مهاجم تلاش می‌کند خروجی مدل را به کلاس خاص مطلوب  $y_{desired} \neq y_{pred}$  تغییر دهد.

هدف و تمرکز اصلی این پژوهش روی حملات تخاصمی گریزانه‌ی جعبه سفید است که از گرادیان خروجی یک مدل نسبت به ورودی آن مطلع هستند. در این حملات یک ورودی ساختگی در زمان تست به صورت

<sup>1</sup>Confidence Reduction

<sup>2</sup>Confidence

<sup>3</sup>Untargeted Misclassification

<sup>4</sup>Targeted Misclassification

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]:

$$\begin{aligned} & \text{minimize } \mathcal{D}(x, x + \delta) \\ \text{s.t. } & C(x + \delta) = t \neq C(x) \\ & x + \delta \in [0, 1]^n \end{aligned} \quad (4-2)$$

که در آن  $x \in \mathbb{R}^n$  یک نمونه‌ی سالم و  $\mathcal{D}$  یک معیار فاصله مشخص است،  $C$  یک مدل دسته‌بند و  $t \neq C(x)$  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 یک مناسب است طوری که علاوه بر کمینه شدن مقدار  $\mathcal{D}(x, x + \delta)$ ، خروجی دسته‌بند  $C$  روی  $x + \delta$  تغییر کند. اکنون در حملات هدفمند، مقدار  $t$  در معادله (۴-۲) از پیش تعیین می‌شود، در حالی که در حملات غیرهدفمند، مقدار  $t$  می‌تواند هر غیر از  $C(x)$  باشد.

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 ۲-۳ 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 افزوده شود، ورودی تولید شده احتمالاً منجر به زیاد شدن مقدار تابع هزینه نهایی و موفقیت حمله خواهد شد. در [۱۲] نویسندهای از نیز تخاصمی ارائه شده در (۶-۲) استفاده می کنند و نمونه تخاصمی نهایی از معادله (۷-۲) بدست می آید.

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

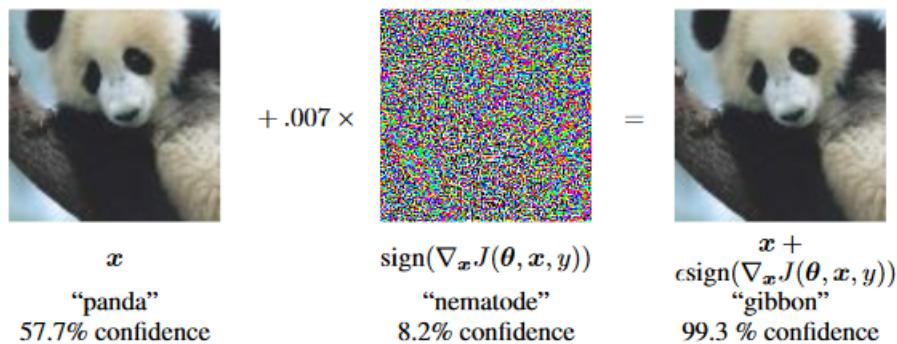
$$\hat{x} = x + \delta \quad (7-2)$$

که در آن مقدار  $\epsilon$  کنترل کننده اندازه نیز 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 برای  $\epsilon$  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 نیز 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.



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

همانطور که مشخص است، این حمله در یک گام انجام می‌شود و یکبار محاسبه گرادیان تابع هزینه برای تولید حمله کفايت 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

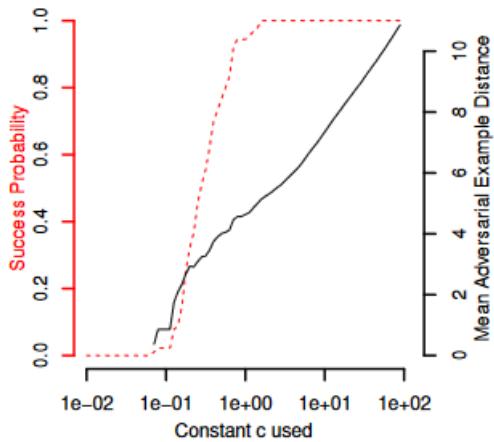
حمله‌ی دیگر مورد استفاده در این پژوهش، حمله‌ی CW است که در [۱۷] ارائه شده است. برای توضیح نحوه‌ی ساختن یک نمونه تخاصمی با این روش به معادله (۴-۲) باز می‌گردیم. در این حمله، از آنجایی که شرط  $C(x + \delta) = t$  بسیار غیرخطی و غیرقابل بهینه‌سازیست، این شرط باید ابتدا به طوری بازنویسی شود که بتوان آن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

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

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



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

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:

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

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 ۲-۵)، می توان مقدار  $c$  بهینه را با استفاده روش جستجوی دودویی<sup>۲</sup> بدست آورد.

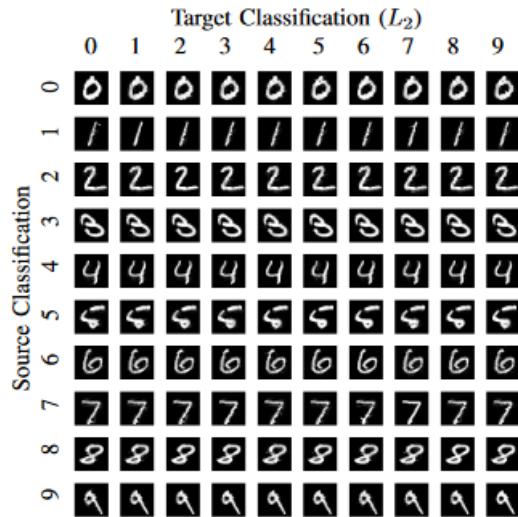
شرطی مانند  $x + \delta \in [0, 1]^n$  به نام محدودیت های جعبه ای<sup>۳</sup> معروف هستند. برای اجرا کردن این شرط در حمله CW از یک تغییر متغیر استفاده می شود. به جای بهینه سازی مستقیم مقدار  $\delta$  در معادله اصلی، متغیر جدید  $w$  معرفی می شود و مقدار

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

<sup>1</sup>Norm

<sup>2</sup>Binary Search

<sup>3</sup>Box Constraints



شکل ۲-۶: نمونه های تخاصمی تولید شده توسط  $L_2$ -CW روی مجموعه داده [۱۹] MNIST.

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

قوی ترین نوع حمله  $L_2$ -CW است که مسئله بهینه سازی آورده شده در رابطه (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 دسته بند قربانی  $F$  هستند (به طوری که  $y = F(x) = \text{softmax}(Z(x))$ ) و پارامتر  $c > 0$  میزان سطح اطمینان حمله را مشخص می کند به طوری که مقادیر بزرگتر  $\alpha$  الگوریتم بهینه ساز را مجبور به یافتن پاسخ هایی می کند که با احتمال قوی تری می توانند خروجی دسته بند را تغییر دهند. به طور معمول از مقدار 14 برای این پارامتر برای تولید حملات قوی استفاده می شود. مسئله بهینه سازی (۱۲-۲) توسط بهینه ساز Adam [۱۹] حل می شود و در نهایت نمونه های تخاصمی به دست خواهند آمد. نمونه هایی از این حمله را می توان در شکل ۶-۲ مشاهده کرد. همانطور که می توان دید، بدون توجه به این که کلاس هدف دارای چه مقداریست تمامی نمونه های تخاصمی متناظر با تصویر سالم به چشم غیر مسلح کاملاً شبیه نمونه اصلی هستند.

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 می‌کنند که تحت عنوان مسئله نقطه‌زنی زیر به عنوان تابع هزینه‌ی مدل قربانی بیان می‌شود:

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

که در آن وظیفه مهاجم حل مسئله بیشینه‌سازی درونی و وظیفه قربانی حل مسئله کمینه‌سازی برونویست. اکنون اگر به فرمول (۶-۲) تشکیل حمله تخاصمی در حمله FGSM بازگردیم، می‌توان گفت که این حمله در واقع می‌تواند به عنوان یک گام از حل مسئله بیشینه‌سازی مهاجم در فرمول بندی (۱۳-۲) در محدوده  $L_\infty$  حول یک نمونه‌ی سالم عمل کند. برای حل این مسئله با استفاده از روش نزول گرادیان<sup>۱</sup>، کافیست همین گام معروفی شده را چندین بار تکرار کنیم و حاصل را روی  $\epsilon$ -کره‌ی حول نمونه‌ی سالم بیافکنیم. این روش بهینه‌سازی با محدودیت که مبنای حمله‌ی PGD است، نزول گرادیان افکنده<sup>۲</sup> نام دارد. به طور دقیق‌تر اگر بخواهیم همچنان مانند حمله FGSM از نرم  $L_\infty$  بهره بگیریم، تشکیل یک حمله تخاصمی در PGD از تکرار گام زیر به دست 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  $x + \mathcal{S}$  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]:

۱. پاکسازی<sup>۴</sup>: در این حالت، هدف پاکسازی ورودی‌های دسته‌بند و یا مقاوم‌سازی دسته‌بند از لحاظ ساختاری

<sup>1</sup>Gradient Descent

<sup>2</sup>Projected Gradient Descent

<sup>3</sup>Learning Rate

<sup>4</sup>Purification

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 مثر	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
بیچیده تر، علاوه بر استفاده از تابع هزینه مخصوص، به پیاده سازی جستجوی دودویی نیز احتیاج است	نسبتاً بیچیده، 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	بیچیدگی پیاده سازی حمله
جمعیه سفید	جمعیه سفید	جمعیه سفید	نوع دسترسی مهاجم

در راستای اصلاح خروجی آن در هنگام بروز حمله است به طوری که به خروجی در زمان‌های دیگر آسیبی وارد نشد.

۲. تشخیص<sup>۱</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 مختصراً راجع به روش‌های پاکسازی و سپس در مورد روش‌های تشخیص بحث خواهد شد.

### ۲-۳-۱ روش‌های پاکسازی حمله

همانطور که پیش‌تر نیز توضیح داده شد، در ادبیات مرتبط با دفاع در برابر حملات تخاصمی، عبارت پاکسازی به دسته خاصی از روش‌های ممکن برای دفاع اطلاق می‌شود که در آن‌ها خروجی دسته‌بندی را در مواجهه با حملات تخاصمی اصلاح می‌کنند بدون آن که به خروجی در ازای ورودی‌های سالم لطمہ‌ای وارد شود. برای سهولت دسته‌بندی، روش‌هایی که به صورت غیرمستقیم موجب تقویت مدل و در نهایت تولید خروجی‌های اصلاح شده می‌شوند را نیز در دسته‌ی روش‌های پاکسازی گنجانده‌ایم. با این مقدمه، روش‌های پاکسازی را می‌توان در ابعاد زیر دسته‌بندی نمود [۱۳، ۹، ۸]:

۱. آموزش تخاصمی<sup>۲</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

<sup>1</sup>Detection

<sup>2</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. در رابطه (۱۳-۲)، مسئله بیشینه‌سازی درونی سعی می‌کند که شبیه فرمول‌بندی ارائه شده در بخش ۲-۱-۲ برای حمله FGSM، قوی‌ترین مقدار<sup>۴</sup> ممکن برای بزرگ کردن مقدار تابع هزینه را پیدا کند. 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 تاب آوری<sup>۵</sup> حمله را در نظر بگیریم، مدل با نمونه‌های تخصصی در یک ۶-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 همگرايی حمله PGD نويز تخصصی بدست آمده برای يك دسته از ورودی‌های سالم، به عنوان نقطه آغازین حمله PGD روی دسته ورودی‌های بعد مورد استفاده قرار می‌گیرد. اين دفاع روی حمله چند - مرحله‌ای PGD [۱۸] مورد آزمایش قرار گرفته و نتایج اميدوار کننده‌ای داشته است. همچنین در YOPO<sup>۵</sup> [۲۱] اين مسئله مورد بررسی قرار گرفته که نياز به يك لايه دفاعي در برابر حملات تخصصي را می‌توان تقریباً فقط به اولین لایه‌ی يك شبکه عميق خلاصه کرد. بنابراین آموزش تخصصی می‌تواند بسیار ارزان‌تر صورت بگیرد. این روش که مستقیماً با مورد مقایسه قرار گرفته است، نشان می‌دهد که می‌تواند در زمانی کمتر دارای Free Adversarial Training 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 GI-AT از این نکته بهره گرفته می‌شود که نمونه داده‌ایی که نزدیک مرزهای تصمیم‌گیری یک مدل قرار دارند، بیشتر می‌توانند در موفقیت یک حمله تخصصی تاثیرگذار باشند. بدین ترتیب این روش با بهره‌گیری از راهکارهای استاندارد آموزش تخصصی، هر یک از نمونه داده‌های آموزشی را بر مبنای این که تولید حمله‌ی تخصصی موفق از روی آن‌ها چقدر دشوار است، وزن‌دهی کرده و در فرایند پس‌انتشار<sup>۱</sup> دخیل می‌کند. نهایتاً در Fast Adversarial Training این نکته به صورت تجربی نشان داده می‌شود که آموزش تخصصی با استفاده از حمله تک-مرحله‌ای FGSM ولی با آغاز تصادفی<sup>۲</sup> (برخلاف [۲۰]) می‌تواند به اندازه آموزش تخصصی با استفاده از حملات قوی‌تر چند-مرحله‌ای (مانند PGD) اثر بخش باشد. در این پژوهش محققین موفق شده‌اند که در کسری از زمان گزارش شده در [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، می‌توانند به راحتی راهکارهای ارائه شده در این کار را دور بزنند، ایده‌های مطرح شده همچنان شایان ذکر به نظر می‌رسند. ایده‌ی اصلی این پژوهش ارائه شده 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.

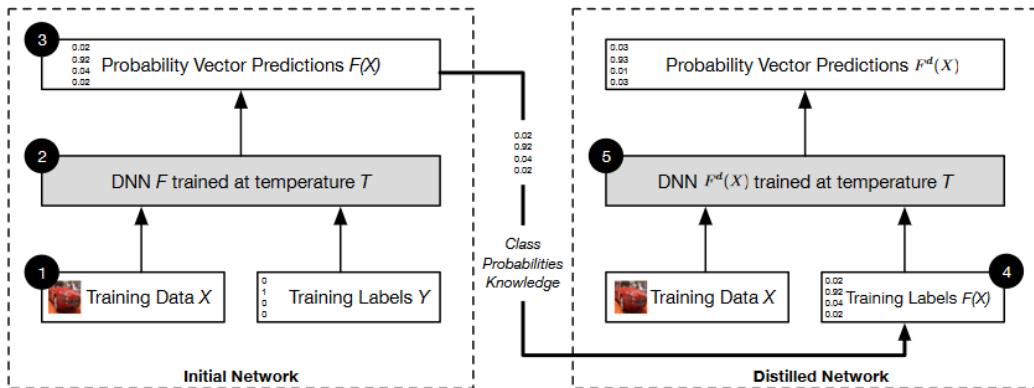
(ج) خودرمزنگار نویزگیر: در این روش، شبیه روش قبلی، یک خودرمزنگار نویز گیر معمولی بدون دانش پیشین از توزیع آماری نویز تخصصی و صرفاً با هدف از بین بردن نویز آموزش داده شده است. سپس در زمان آموزش، هر پیکسل از نمونه‌ی آموزشی با یک نویز گاوی با میانگین ۰ و انحراف معیار متغیر  $\sigma$  ترکیب می‌شود. نشان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



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

خوبی خودگزار قبلی عمل کند.

ایراد اصلی وارد به موارد ۲ ب و ۲ ج این است که با سری کردن خودگزار و مدل قربانی و حمله به مدل سری شده، همچنان می‌توان حمله موفقی داشت. این در حالیست که روش آن<sup>۲۶</sup> نمی‌توانست در برابر حملات جعبه‌سفید قوی‌تر که بعدها ارائه شدند (مانند PGD و CW) مقاومت کند.

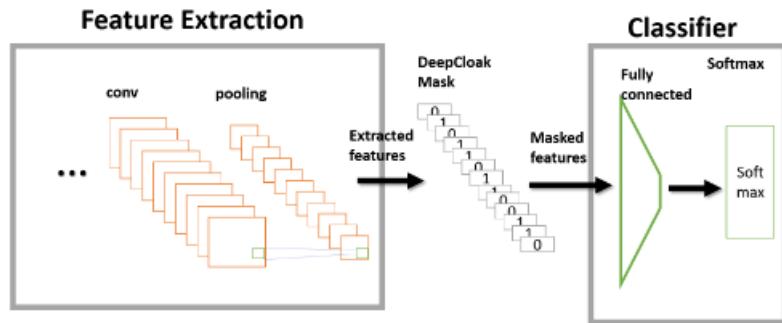
ایده‌ی مهم دیگری که کمی بعدتر ارائه شد، ایده‌ی تقطیر دفاعی<sup>۱</sup> [۲۵] است. همانطور که در شکل ۷-۲ نشان داده شده است، ایده‌ی اصلی در این روش استفاده از پیش‌بینی‌های یک مدل از پیش آموزش داده شده روی مجموعه‌داده‌های سالم، به عنوان برچسب‌های جدید مدل تقطیر شده است. بدین ترتیب، مدل تقطیر شده برخلاف مدل 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 [۲۸-۲۹].

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

<sup>۱</sup>Defensive Distillation



شکل ۲-۸: نحوه عملکرد [۳۰] DeepCloak

بین پیشینی مدل و برچسب واقعی تغییر چندانی نخواهد کرد و بنابراین نمونه‌های تخاصمی که در یک ۴- کره محدود می‌شوند، نخواهند توانست خروجی مدل قربانی را تغییر دهنند.

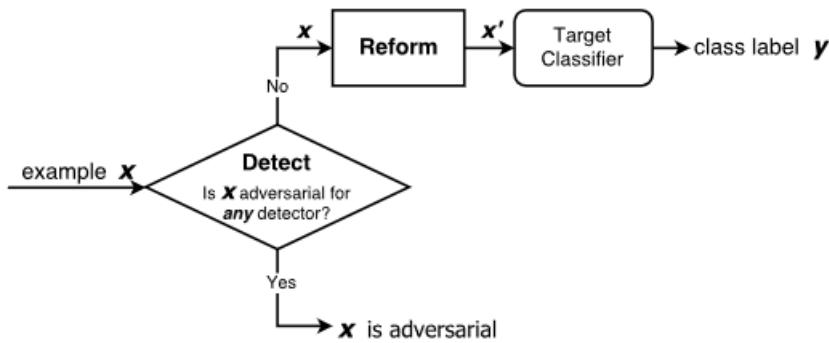
در روش DeepCloak [۳۰] که در شکل ۲-۸ نمایش داده شده، ایده مطرح شده آن است که با قرار دادن یک لایه ماسک‌کننده دقیقاً قبل از لایه خطی که تولید کننده logit های مدل، ویژگی‌های بی‌همیت در خروجی نهایی مدل را از صفر کرده و آن را نسبت به نویز تخاصمی مقاوم‌تر ساخت. آموزش این لایه با دادن ورودی‌های سالم و تخاصمی 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 اشتباہ دسته‌بندها در مواجهه با یک نمونه تخاصمی برشمرده شده است:



[۴۲] نحوه عملکرد MagNet

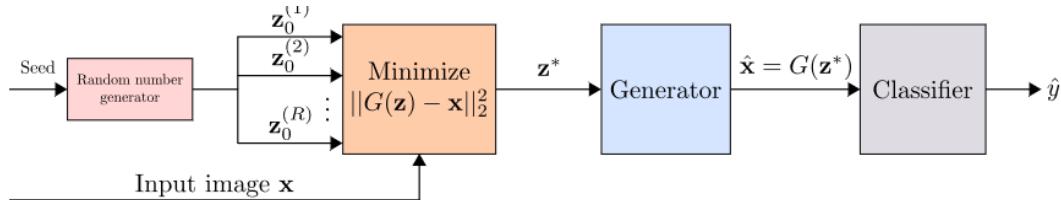
(آ) نمونه تخاصمی از مرزهای خمینه<sup>۱</sup> وظیفه مورد نظر دور است. به عنوان مثال اگر وظیفه تشخیص اعداد دست نویس و مجموعه داده MNIST را در نظر بگیریم، یک نمونه تخاصمی ممکن است تصویری باشد که اصلاً شامل یک عدد دست نویس نیست ولی دسته بند از آنجایی که مجبور به تولید خروجی است، دچار اشتباه خواهد شد.

(ب) نمونه تخاصمی به مرزهای خمینه وظیفه مورد نظر خیلی نزدیک است. در این حالت که اکثر حملات تخاصمی جدید از آن استفاده می‌کنند، اگر دارای یک دسته بند باشیم، دسته بندی که در فضای اطراف یک نمونه تخاصمی قدرت تعمیم 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 می‌شود. برای دفاع در برابر علت اول مطرح شده، MagNet از چندین شبکه تشخیص دهنده استفاده می‌کند برای آن که فاصله یک نمونه زمان تست را با نمونه‌های آموزشی بسنجد. به عبارت دقیق‌تر، یک تشخیص دهنده تابع  $f : \mathcal{X} \rightarrow (0, 1)$  را یاد می‌گیرد که در آن  $\mathcal{X}$  مجموعه تمام نمونه‌های زمان تست است و خروجی این تشخیص دهنده معیاری از فاصله نمونه زمان تست با خمینه نمونه‌های سالم زمان آموزش است. سپس برای برطرف کردن مشکل دوم، این چارچوب از یک اصلاح کننده<sup>۲</sup> برای تصحیح نمونه‌های دور از خمینه نمونه‌های سالم استفاده می‌شود. این شبکه اصلاح کننده توسط یک خود-رمزگذار پیاده‌سازی می‌شود. در صورت تشخیص نمونه تخاصمی توسط حداقل یکی از تشخیص دهنده‌ها، این نمونه به شبکه اصلاح کننده داده می‌شود و خروجی شبکه اصلاح کننده در نهایت به دسته بند می‌رسد (شکل ۹-۲).

یکی از مهم‌ترین ایده‌های مطرح شده در زمینه استفاده از شبکه‌های جانبی چارچوب Defense-GAN

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



شکل ۲-۱۰: دورنمای عملکرد چارچوب [۳۴] Defense-GAN

است. در این پژوهش برای پاکسازی یک نمونه تخصصی از یک شبکه مولد تخصصی<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 تخصصی 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 تخصصی کمک خواهد کرد. یکی از نقاط قوت Defense-GAN ماهیت غیرخطی آن به دلیل وجود یک حلقه طبیعی‌سازی نزول گرادیان در فرایند پیاده‌سازی مکانیزم دفاع می‌باشد. این امر، Defense-GAN را نسبت به حملات جعبه‌سفید مقاوم می‌سازد. همچنین، این چارچوب هیچ پیش‌فرضی راجع به نوع حمله تخصصی 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 تخصصی برای پاکسازی نویز تخصصی استفاده شده است. این چارچوب که در شکل ۱۱-۲ نمایش داده شده است، از یک خودرمزنگذار به عنوان مولد و یک شبکه ممیز تشکیل شده است. هدف نهایی این چارچوب آن است که مولد  $G$  طوری آموزش ببیند که بتوان نویزهای ناچیز 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

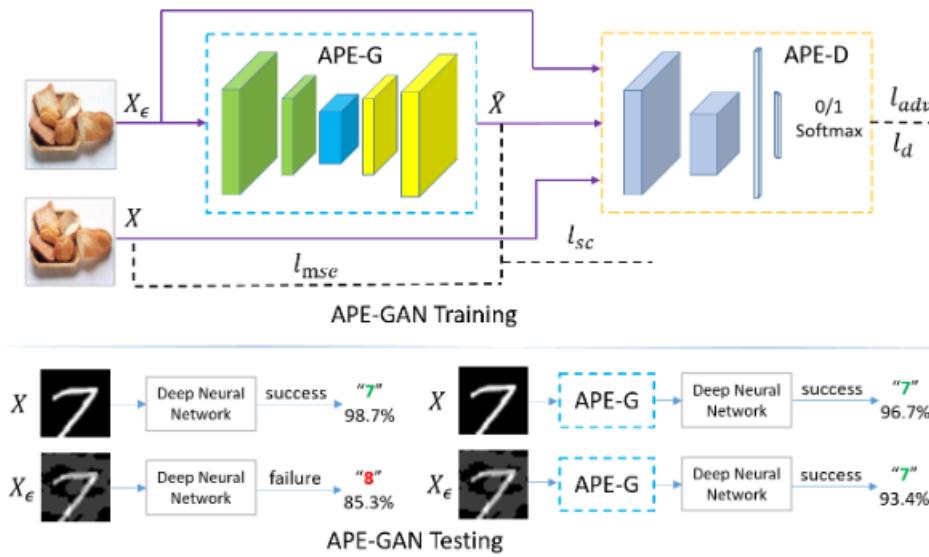
<sup>1</sup>Generative Adversarial Network (GAN)

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

<sup>3</sup>برای توضیحات مقدماتی راجع به GAN ها، می‌توانید به بخش ۱-۴-۲ مراجعه کنید

<sup>4</sup>Latent

<sup>5</sup>Adversarial Perturbation Elimination Generative Adversarial Network



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

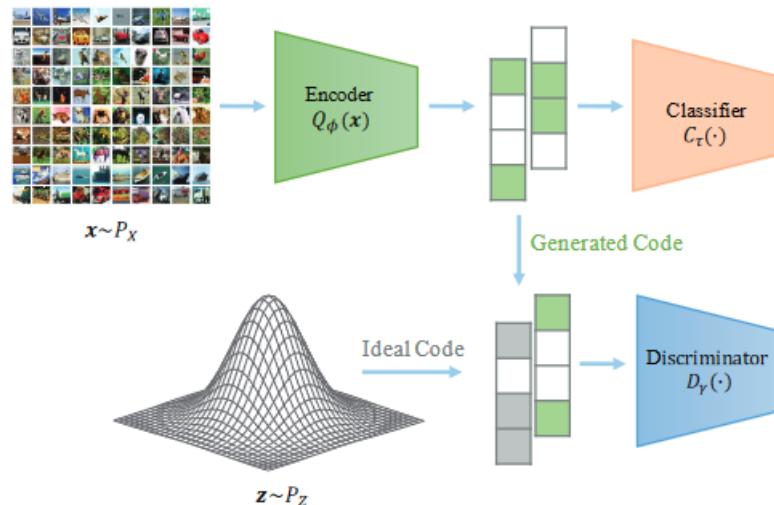
از یک ساختار تخاصمی و برقاری یک بازی خصم‌مانه بین این  $G$  و یک مدل ممیز  $D$  استفاده می‌شود. وظیفه  $D$  در این وضعیت تشخیص دادن نمونه‌های پاکسازی شده توسط  $G$  از نمونه‌های سالم متناظر است. در نهایت این مولد قادر به تولید نمونه‌های پاکسازی شده‌ای خواهد بود که توسط ممیز غیر قابل تشخیص هستند و در 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 تخاصمی 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 تغییر کنیم، دسته‌بند عملکرد بهتری در برابر نمونه‌های تخاصمی خواهد داشت. همان‌طور که در شکل ۱۲-۲ نشان داده شده است، هر دسته‌بند را می‌توان به صورت دو زیر شبکه متصور شد: یک Encoder که وظیفه آن استخراج ویژگی از ورودی دسته‌بند 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>۲</sup> مقایسه می‌شوند و توسط یک شبکه جانبی ممیز<sup>۳</sup>

<sup>1</sup>Embedding Regularized Classifier

<sup>2</sup>Prior

<sup>3</sup>Discriminator



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

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>۱</sup> برای پاکسازی نویز تخاصمی تزریق شده به یک تصویر استفاده می‌شود<sup>۲</sup>. نحوه عملکرد این چارچوب در شکل ۱۳-۲ نمایش داده شده است. مدل‌های انتشاری از دو فرایند انتشار پیش‌رو<sup>۳</sup> و انتشار معکوس<sup>۴</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.

## ۲-۳-۲ روش‌های تشخیص حمله

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

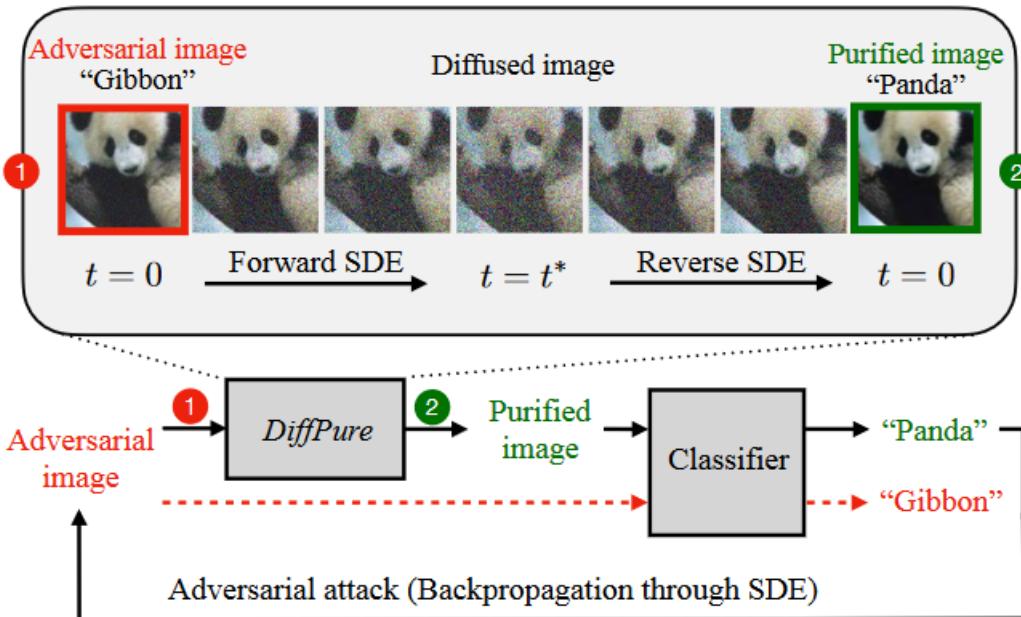
۱. تشخیص دهنده‌های مبتنی بر ورودی یا خروجی‌های مدل قربانی: ساده‌ترین روش تشخیص نمونه‌های

<sup>۱</sup>Diffusion Model

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

<sup>۳</sup>Forward Diffusion

<sup>۴</sup>Reverse Diffusion

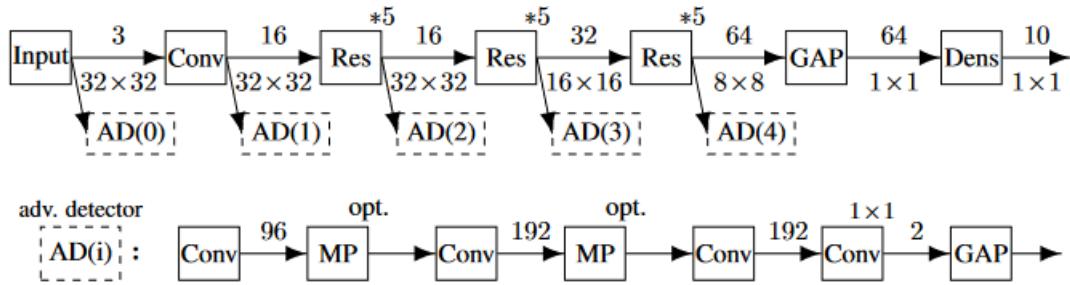


شکل ۲-۱۳: نحوه عملکرد [۳۶] DiffPure

تخاصمی، پیاده‌سازی یک دسته‌بند دو-کلاسه برای تمیز دادن نمونه‌های سالم از نمونه‌های تخاصمی است. 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 تعمیم‌پذیری کمی دارد و خصوصاً نسبت به نوع حمله مورد استفاده شده بسیار حساس است. در این پژوهش از حملات FGSM و JSMA [۳۸] برای ارزیابی مدل استفاده شده است. نتایج تجربی نشان می‌دهند که تشخیص دهنده‌ها به نوع حمله‌ای که برای آن آموزش دیده اند حساس هستند و بنابراین قابلیت تعمیم‌پذیری اندکی نسبت به انواع مختلف حملات تخاصمی دارند. به عبارت دقیق‌تر، تشخیص دهنده‌هایی که روی حملات FGSM آموزش داده شده‌اند، نتوانستند در این پژوهش در برابر حملات JSMA به خوبی دفاع کنند و بلعکس، تشخیص دهنده‌های JSMA عملکرد ضعیفی در برابر حملات FGSM داشتند.

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 پیشین هم به زیرشبکه تشخیص دهنده و هم به ادامه دسته‌بند، منتقل خواهند شد. این معماری در شکل ۲-۱۴ نشان داده شده است. برای آموزش این شبکه، ابتدا دسته‌بند اصلی روی نمونه‌های سالم آموزش داده خواهد شد. سپس، زیرشبکه‌های تشخیص دهنده در لایه‌های میانی به دسته‌بند اضافه می‌شوند

<sup>۱</sup>Hidden Layers



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

و نمونه‌های تخاصمی به تعداد برابر نمونه‌های آموزشی، تشکیل خواهند شد. در نهایت وزن‌های دسته‌بند اصلی ثابت<sup>۱</sup> شده و فقط زیر شبکه‌های تشخیص دهنده روی نمونه‌های سالم و تخاصمی به عنوان یک دسته‌بند دو-کلاسه، آموزش داده خواهند شد. در [۴۰] ایده‌ی مشابه ولی با ارتباط بین زیر شبکه‌های دسته‌بند با الهام گرفتن از ایده‌ی Boosting برای دسته‌بندی‌های آبشراری<sup>۲</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>۳</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 مجموعه داده 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>۴</sup> مطرح می‌کنند. بدین ترتیب برای هر یک از کلاس‌های مجموعه داده 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 مشاهده شده در یک لایه‌ی پنهان کاملاً متصل برای ورودی  $x$  به شرط پارامترهای مدل  $\theta$  و کلاس  $c$  به صورت ترکیب وزن‌داری از توزیع‌های گاوسی تخمین زده می‌شود و سپس با یافتن یک حد آستانه  $TH_c$  برای هر کلاس، می‌توان از

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

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

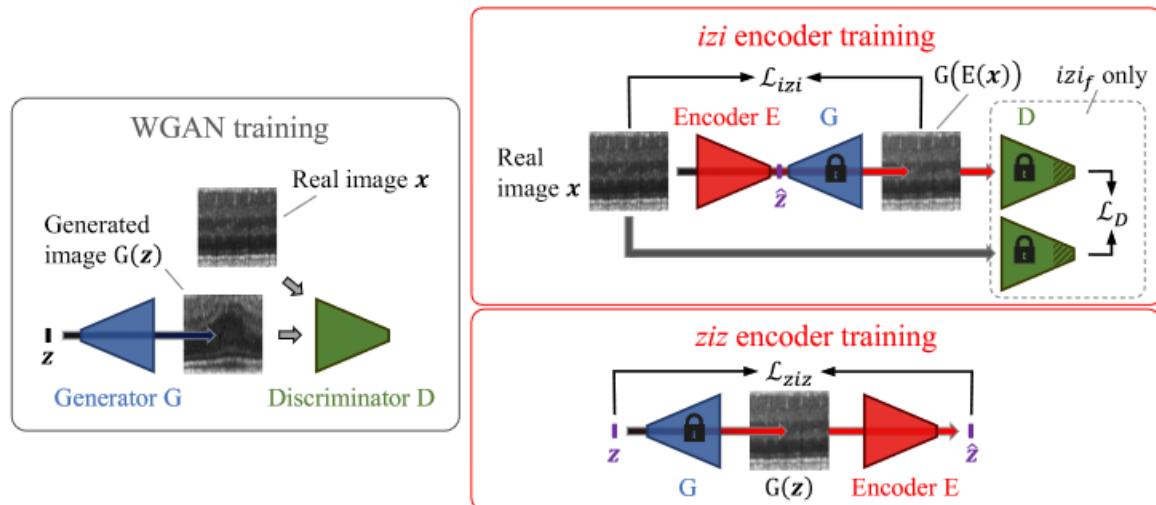
در [۴۳] ML-LOO مفهومی تحت عنوان Feature Attribution برای تشخیص حملات تخاصمی مطرح

<sup>1</sup>Freeze

<sup>2</sup>Cascade

<sup>3</sup>Fully Connected

<sup>4</sup>Gaussian Mixture Model



[۴۴] دورنمای نحوه آموزش f-AnoGAN

می شود. این مفهوم که برای هر ورودی  $\Phi(x) : \mathbb{R}^d \rightarrow \mathbb{R}^d$  با  $x \in \mathbb{R}^d$  نمایش داده است، معیاری از میزان تاثیر هر ویژگی از ورودی (برای تصاویر، هر پیکسل از تصویر) در خروجی دسته‌بندی نهایی است. برای اندازه‌گیری  $\Phi$  از روش LOO<sup>۱</sup> استفاده می‌شود، بدین ترتیب که به ازای هر ویژگی از ورودی اختلاف احتمال خروجی محتمل‌ترین کلاس در حالت عادی و در زمانی که آن ویژگی با یک مقدار مرجع (مثلاً صفر) جایگزین شده است، محاسبه می‌شود. به عبارت دقیق‌تر:

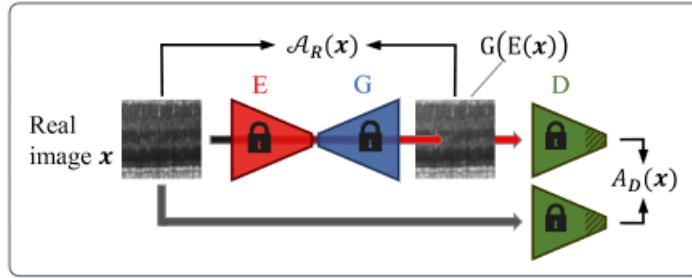
$$\Phi(x)_i := f(x)_c - f(x_{(i)})_c, \quad \text{s.t. } c = \arg \max_{j \in C} f(x)_j.$$

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

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

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

<sup>۱</sup>Leave-One-Out



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

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 نظیر یک دیگر باشند و بنابراین

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

ولی اگر در نمونه  $x$  آنومالی وجود داشته باشد، از آنجایی که  $E$  فقط 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  $D_{interm}(\cdot)$  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  $A(x)$  را بدست خواهد داد که از آن و با کمک یک حد آستانه تنظیم شده می‌توان برای تشخیص آنومالی استفاده کرد.

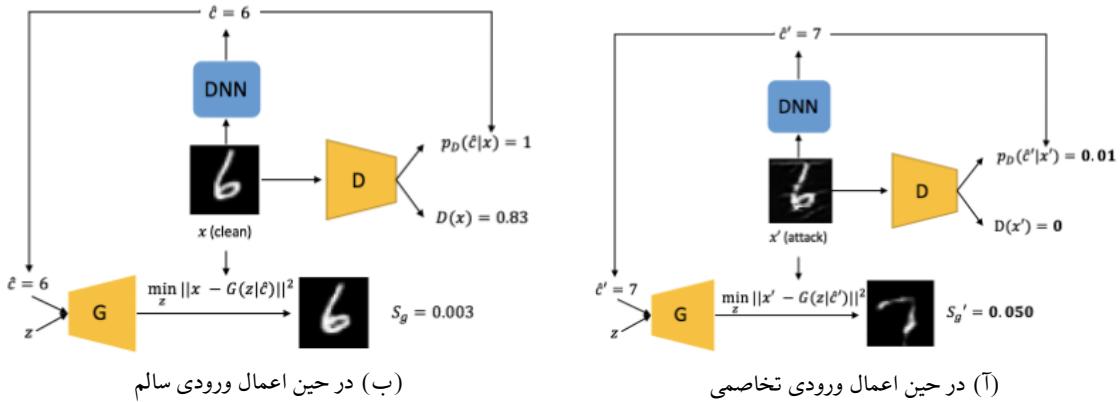
در نهایت روش ارائه شده در [۴] را مورد بررسی قرار خواهیم داد<sup>۳</sup>. در این روش - که نزدیک ترین کار انجام شده به کار ماست - بار دیگر از شبکه‌های مولد تخصصی، ولی این بار مشروط بر کلاس<sup>۴</sup> کمک گرفته شده است. در ACGAN-ADA<sup>۵</sup> ابتدا یک شبکه ACGAN ایجاد می‌شود. سپس از تمام بخش‌های این شبکه برای ایجاد سنجه‌هایی برای تشخیص حمله استفاده می‌شود.

<sup>1</sup> Reconstruction Error

<sup>2</sup> Feature Matching

<sup>3</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>4</sup> Class Conditional



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

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 ورودی است. نحوه عملکرد این روش در شکل ۲-۱۷ نشان داده شده است. در حین اعمال ورودی سالم (شکل ۲-۱۷-۲) مقدار  $D(x)$  و  $p_D(\hat{c}|x)$  بالا بوده در حالی که مقدار  $S_g$  ناچیز است. در مقابل، در حین اعمال ورودی تخاصمی (شکل ۲-۱۷-۲آ) دقیقاً بر عکس این حالت اتفاق خواهد افتاد. در ادامه این پژوهش ترکیب‌های مختلفی از این سنجه‌ها را مورد استفاده قرار داده است و بهترین حالت ممکن گزارش شده، حالت D-AD است که فقط از سنجه‌های مربوط به زیرشبکه ممیز ( $S_C$  و  $S_R$ ) استفاده می‌کند.

<sup>1</sup>Posterior

<sup>2</sup>Auxiliary Classifier

۲. تشخیص دهنده‌های مبتنی بر ویژگی‌های خاص نمونه‌های تخاصمی: دیدگاه دیگری که می‌توان برای تشخیص نمونه‌های تخاصمی در نظر گرفت 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-Detection [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  $(x)$   $p_X$  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 دفاع داده می‌شوند و پس از مشخص شدن برچسب پیش‌بینی شده توسط این مدل (چه درست، چه غلط بر اثر حمله تخاصمی)، این ورودی به KDE متناظر فرستاده خواهد شد و در نهایت از یک مدل Logistic Regression برای تشخیص نمونه‌های تخاصمی از روی امتیاز چگالی آن نمونه خاص توسط KDE متناظر، استفاده می‌شود.

ایده‌ی دیگر مطرح شده در این زمینه، ADA [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 این نمونه‌های تخاصمی از کلاس نمونه سالم متناظر شان نیز چندان دور نخواهند بود. در همین راستا، در این پژوهش با استفاده از مدل‌های تخمین چگالی، فضای تولید شده توسط لایه‌های پنهان دسته‌بند مورد دفاع را به صورت ریاضی مدل می‌کنند. سپس از دیورژانس<sup>۲</sup> KL<sup>۳</sup> بین معیارهای تخمین زده شده

<sup>۱</sup>Kernel Density Estimation

<sup>۲</sup>Divergence

<sup>۳</sup>Kullback-Leibler

توسط خود دسته‌بند و مدل‌های تخمین چگالی و یک حد آستانه مناسب، برای تشخیص حملات استفاده می‌شود.

در [۵۰] روش دیگری تحت عنوان MD<sup>۱</sup> 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  $(\cdot)^f$  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 گاووسی چند متغیره پیروی می‌کند، می‌توان یک دسته‌بند مولد تعریف کرد. به عبارت دقیق‌تر، می‌توان  $C$  توزیع گاووسی 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  $C$  است. علت منطقی بودن این فرض آن است که می‌توان Gaussian Discriminant Analysis (GDA) با یک دسته‌بند softmax نشان داد چنین دسته‌بند مولدی تحت معادل است. اکنون برای تشکیل معیاری از سطح اطمینان نسبت به یک نمونه از فاصله ماهalanobis<sup>۲</sup> بین نمونه  $x$  و نزدیک ترین گاووسی مشروط بر کلاس موجود از بین  $C$  گاووسی 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.

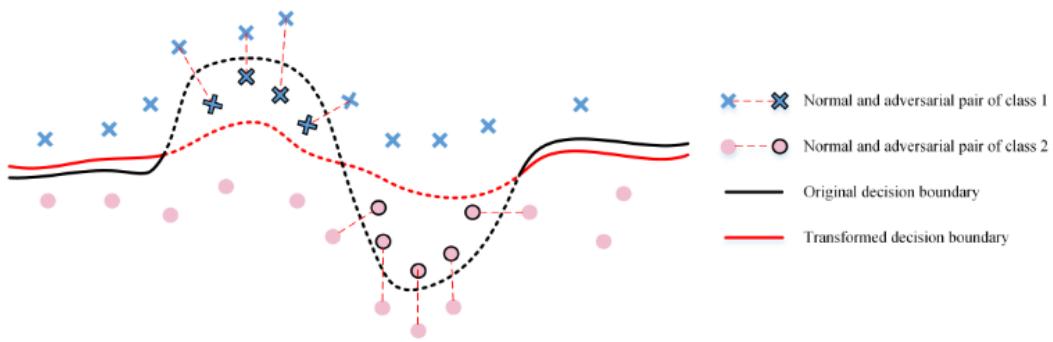
در [۵۱] روش آماری دیگری برای تشخیص نمونه‌های تخاصمی ارائه شده است. اگر logit های پیش از اعمال softmax در یک دسته‌بند با بردار  $\vec{f}_y(x)$  نشان داده شوند،  $f_y(x)$  درایه‌ی  $y$ -ام این بردار خواهد بود که به عبارتی دیگر، log-odds برای کلاس  $y$  نیز پنداشته می‌شود. همچنین فرض کنیم که مقادیر جفتی log-odds 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 های جفتی چگونه با افروzen مقدار کمی نویز  $\eta$  به ورودی  $x$  هنگامی که  $y_{true} = y$  اگر برچسب‌های واقعی در اختیار باشند

<sup>1</sup>Mahalanobis Detector

<sup>2</sup>Mahalanobis Distance

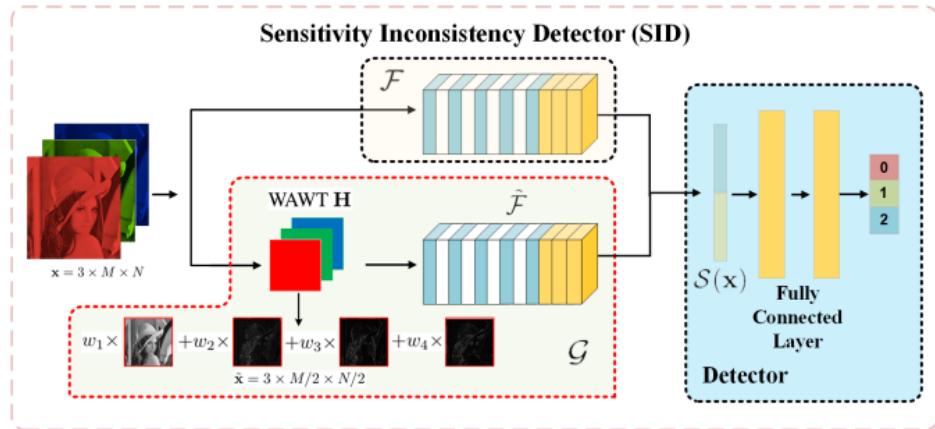


شکل ۱۸-۲: تشکیل نمونه‌های تخاصمی در اطراف خمیدگی‌های مرز تصمیم یک دسته‌بند [۵۲]

و هنگامی که  $y = F(x) = \arg \max_y f_y(x)$  در زمان تست، تغییر می‌کنند. 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 افزوده  $\eta$  مقاوم نیست، بنابراین اگر نمونه تخاصمی  $\delta = \hat{x} - x$  در اختیار باشد که شده مقدار کمی از تاثیر نویز تخاصمی  $\delta$  را از بین برده و مقدار log-odds جفتی بین کلاس به اشتباه تشخیص داده شده توسط دسته‌بند و کلاس اصلی را طبق معادله (۱۵-۲) 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 جایه‌جا کردن نمونه‌های سالم به سمت دیگر مرز تصمیم در جاها بیی 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 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

<sup>۱</sup>Sensitivity Inconsistency Detector



شکل ۱۹-۲: عملکرد SID در زمان تست [۵۲]

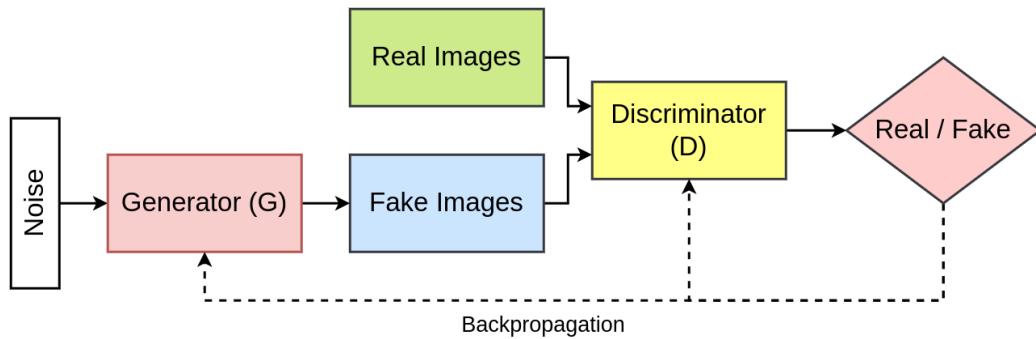
ایجاد شده توسط  $\mathcal{F}$  و  $\mathcal{G}$  را حداقل می‌کند:

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

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 معادله‌ی (۱۶-۲) را برآورده کند، در این پژوهش از تبدیل Wavelet با میانگین وزن دار<sup>۲</sup> استفاده می‌شود. در نهایت، مطابق شکل ۱۹-۲ خروجی‌های دو دسته‌بند به یک شبکه تشخیص‌دهنده‌ی نهایی تحویل داده می‌شوند و انتظار می‌رود که بر اساس اختلاف مرز تصمیم‌های ایجاد شده توسط دو دسته‌بند، شبکه تشخیص‌دهنده بتواند سالم و یا تخاصمی بودن یک نمونه‌ی ورودی را تشخیص دهد.

<sup>1</sup> Slack Variable

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



شکل ۲۰ - ۲: دورنمای شبکه‌های مولد تخصصی

#### 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 و غیره است. در هوش مولد از محتوای موجود کنونی و برای آموزش مدل هایی استفاده می شود که توانایی خلق محتوای جدید را داشته باشند بدون آن که لزوماً نمونه های آموزشی را دقیقاً در خروجی تکرار کنند. در حقیقت مدل های مولد تخمینی پارامتری از توزیع احتمالی پیچیده محتوایی که قرار است تولید کنند را یاد میگیرند و در نهایت روشی برای نمونه گیری از این تخمین پارامتری به ما ارائه خواهد کرد.

در ادامه این بخش به بررسی دو مورد از اصلی ترین مدل های مولد در حوضه های 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 نظریه بازی ها در هوش مصنوعی، معرفی شدند. بنای اصلی این شبکه ها به عنوان چارچوبی جدید برای یادگیری بدون ناظارت<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، ایده ای کلی GAN ها برقراری یک بازی مجموع - صفر<sup>۳</sup> بین دو شبکه به نام های مولد<sup>۴</sup> و ممیز است. هدف شبکه های مولد در حالت ایده آل تولید تصاویری از توزیع تصاویر حقیقی موجود است. برآورده کردن این هدف به کمک بازی ایجاد شده میان مولد و ممیز صورت می گیرد.

<sup>1</sup>Generative AI

<sup>2</sup>Unsupervised Learning

<sup>3</sup>Zero-sum

<sup>4</sup>Generator

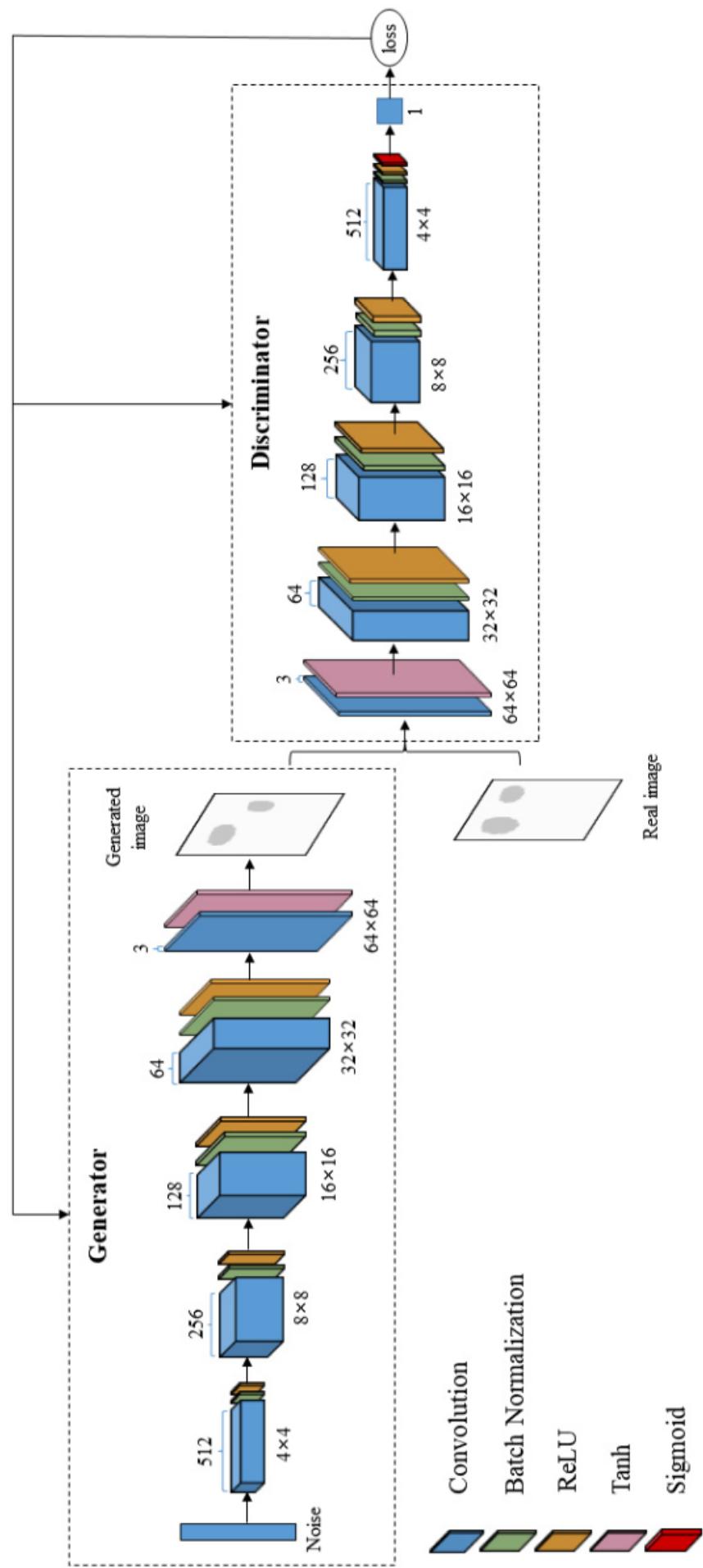
مولد تلاش می‌کند با تولید نمونه‌هایی که به نمونه‌های واقعی شبیه هستند، ممیز را فریب دهد. از طرف دیگر، ممیز در تلاش است که با تقویت خودش در برابر فریب خوردن از مولد مصون بماند و همچنان بتواند نمونه‌های ساختگی و واقعی را از هم تفکیک کند. به طور دقیق‌تر، مولد  $G$  و ممیز  $D$  بازی کمینه–بیشینه زیر را 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 بندی مناسب برای بازی، نقطه تعادل بهینه بازی در جایی اتفاق خواهد افتاد که توزیع آماری نمونه‌های تولید شده توسط مولد دقیقاً با توزیع نمونه‌های واقعی برابر باشد (قضیه ۱ در [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 شبکه 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 شگرفی 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



(ب) نمونه‌های تولید شده روی مجموعه داده CIFAR

(آ) نمونه‌های تولید شده روی مجموعه داده MNIST

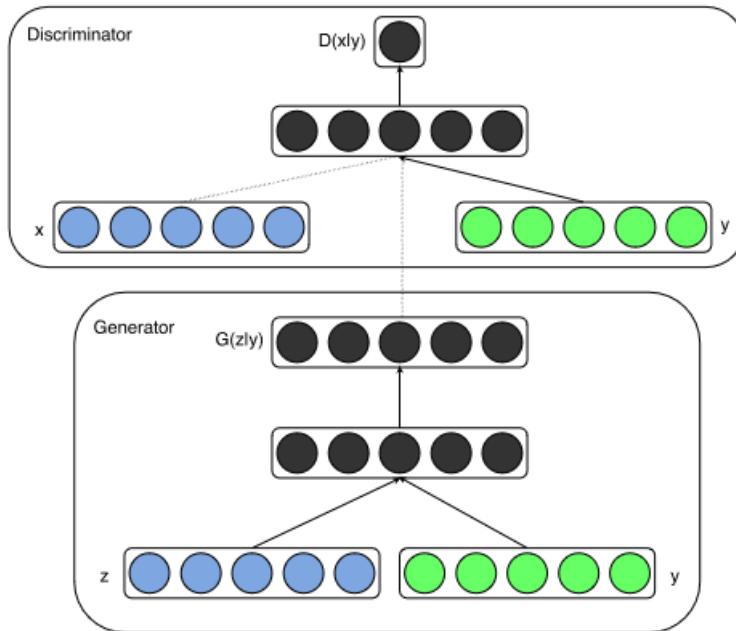
شکل ۲-۲: نمونه‌های تولید شده توسط یک GAN معمولی [۵۳]

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 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 نمونه 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 نش<sup>۲</sup> در GAN‌ها به دلیل ساختار بازی کمینه-بیشینه برقرار شده بین دو شبکه و پیچیدگی حل یک مسئله بهینه‌سازی نقطه زینی، مطلوب آرمانی اما بسیار دشوار است. 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 بعضیاً 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 مولد به مراتب ساده‌تر است، ممیز به سرعت از مولد در بازی پیشی می‌گیرد و می‌تواند به راحتی نمونه‌های

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



شکل ۲-۲: معماری cGAN [۵۸]

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 cGAN 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 پژوهش‌های احتمالی آینده معرفی شده بود و ظرف مدت چند ماه، اولین تحقیق در این زمینه به ثمر رسید و در [۵۸] محققین معماری cGAN<sup>۲</sup> را معرفی کردند. ایده‌ی تزریق یک شرط به ورودی‌های مولد و ممیز در این پژوهش به صورت بسیار ابتدایی و بدیهی مطرح شده است و به عنوان عامل مشروط کننده از برچسب کلاس

<sup>1</sup>Vanishing Gradient

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



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].

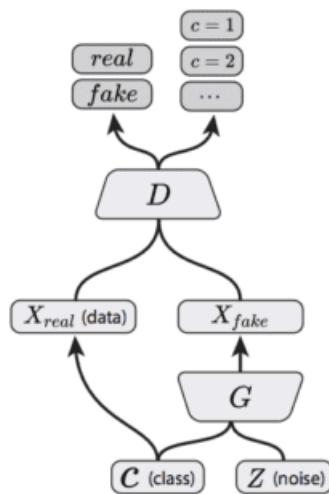
نمونه‌ها استفاده می‌شود. مطابق شکل ۲۳-۲ برای تزریق این شرط در هر دو شبکه‌ی مولد و ممیز یک لایه‌ی embedding قابل یادگیری معرفی می‌شود که برچسب را به فضای چند بعدی دلخواه ببرد. پس از دریافت embedding متناظر با برچسب ورودی، این بردار به بردار ورودی چسبانده<sup>۱</sup> می‌شود و لایه‌های بعدی مانند یک GAN معمولی عمل 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.

---

<sup>1</sup>Concatenate



شکل ۲۵-۲: معماری ACGAN

ACGAN ۲-۱-۴-۲

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 هدایت 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.  $P(C|x)$ .

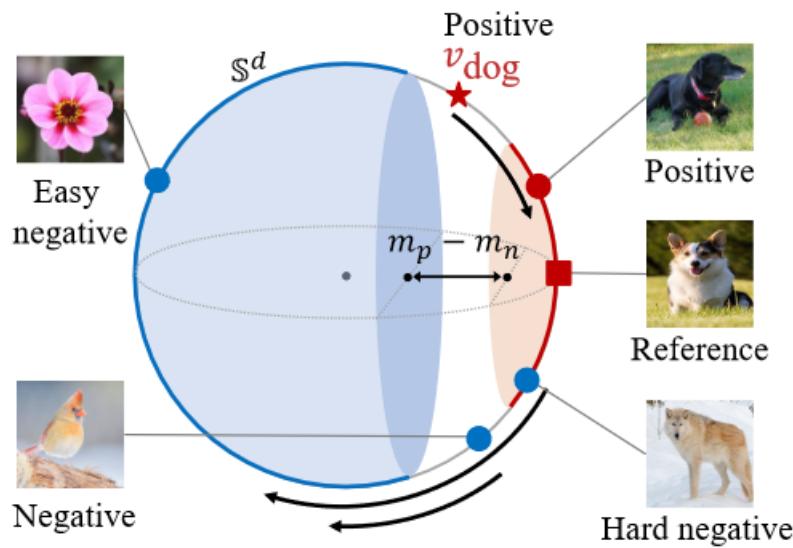
به تبع این موضوع، تابع هزینه‌ی آموزش یک ACGAN از دو بخش تشکیل می‌شود:

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

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 ۲۶-۲: فضای embedding تصاویر در یک ReACGAN

بدین ترتیب ممیز 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 هنگامی است که تعداد کلاس‌های مجموعه داده‌ها زیاد می‌شود.

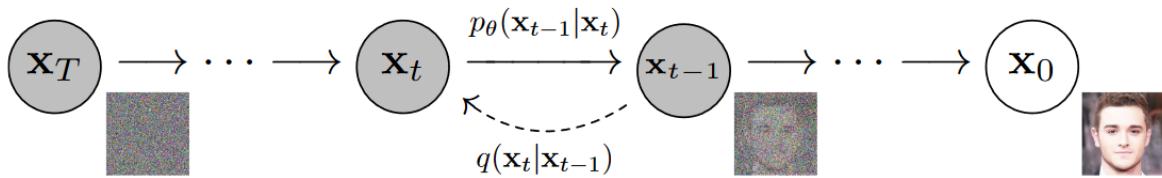
#### ReACGAN ۳-۱-۴-۲

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 متعدد، 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 مستتر در خود نمونه‌های آموزش با در نظر گرفت تفاوت‌ها و شباهت‌های آن‌ها نسبت به فاصله‌ی آن‌ها در فضای embedding است. همانطور که در شکل

<sup>1</sup>Data-to-Data Cross-Entropy



شکل ۲-۲۷: زنجیره مارکوف نظیر یک DDPM [۵۶]

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

## ۲-۴-۲ مدل های انتشاری

دسته ای دیگری از مدل های مولد که اخیراً معرفی شده اند و به سرعت در حال گسترش هستند، مدل های انتشاری هستند. ایده ای مدل های انتشاری از مبحثی در فیزیک تحت عنوان ترمودینامیک غیرتعادلی<sup>۲</sup> الگو برداری شده است. با وجود این که اولین پژوهش در رابطه با مدل های انتشاری در سال ۱۵۰۰ و تقریباً به صورت همزمان با GAN منتشر شد [۵۹]، این مدل ها اولین بار در ۲۰۲۰ پس از انتشار مقاله معروف [۵۶] تحت عنوان مدل های انتشار احتمالی حذف کننده نویز<sup>۳</sup> شهرت جهانی پیدا کردند. مدل های انتشاری یک زنجیره مارکوف<sup>۴</sup> از گام های انتشار را تعریف می کنند که به تدریج مقداری نویز تصادفی به یک نمونه می افزایند و سپس یاد می گیرند که عکس این فرایند انتشار را انجام دهند تا بتوانند با شروع از نویز به یک نمونه ساختگی که به توزیع نمونه های واقعی نزدیک است، برستند (شکل ۲-۲۷).

<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  $\{\beta_t \in (0, 1)\}_{t=1}^T$  کنترل می‌شود. اگر توزیع احتمالی فرایند انتشار پیش‌رو را با  $q$  نشان دهیم، طبق تعریف 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  $(x_t | x_{t-1})$  از  $q$  نمونه برداری کرد، می‌توان با شروع از نمونه‌ای از یک توزیع گاویسی ایزوتروپیک و اعمال  $T$  گام زمانی معکوس، به نمونه‌ای از توزیع واقعی  $(x) \sim q$  رسید. می‌توان نشان داد که اگر  $\beta_t$  به اندازه کافی کوچک باشد، انتشار معکوس  $(x_{t-1} | x_t) \sim q$  نیز تقریباً یک توزیع گاویسی است. ولی از آنجایی که محاسبه‌ی  $(x_{t-1} | x_t) \sim q$  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.



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

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 مُعمولاً 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 گاوی 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 بر باشد. 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 راجع به مدل‌های انتشاری مشروط صحبت خواهیم کرد.

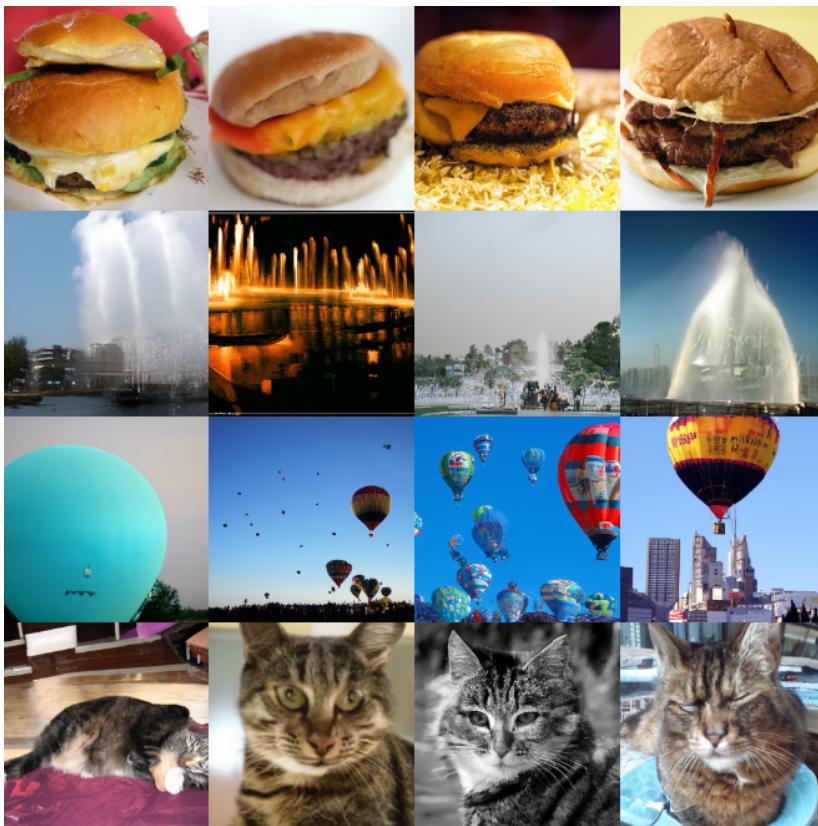
#### ۱-۲-۴-۲ مدل‌های انتشاری هدایت شده (مشروط)

مانند GAN‌ها تولید مشروط محتوا بر مبنای شروطی مانند برچسب کلاس و یا متن توصیف کننده‌ی 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  $f_\phi(y|x_t)$

<sup>1</sup>Maximum Likelihood Estimation

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



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

روی تصاویر نویزی  $x_t$  آموزش داده شده و سپس از الگاریتم 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 ضریبی 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 مقدار 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

<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  $y$  خواهد 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  $y$ ، دور انداخته می‌شود تا مدل توزیع غیرشرطی  $(x|y)$  را نیز به طور ضمنی یاد بگیرد. در نهایت برای تولید یک نمونه مانند حالت هدایت شده با دسته‌بند، ترکیبی از توزیع‌های شرطی و غیرشرطی 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 متناظر با شرط داده شده در زمان آموزش را یاد گرفته‌اند 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 قوی تر از مدل‌های انتشاری هدایت شده با دسته‌بند عمل می‌کنند. در شکل ۲-۳۰ تعدادی از نمونه‌های تولید شده توسط این مدل مشاهده می‌شود.

در این فصل ابتدا در بخش ۲-۲ به بیان تعریف رسمی حملات تخاصمی و انواع مختلف آن‌ها پرداختیم. سپس در بخش ۳-۲، یک دسته‌بندی روی روش‌های دفاع در برابر حملات تخاصمی از دو جنبه‌ی تشخیص و پاکسازی، ارائه شد. نهایتاً در بخش ۴-۲ در مورد هوش مولد و در نوع مهم از مدل‌های مولد (یعنی شبکه‌های مولد تخاصمی و مدل‌های انتشاری) بحث شد. در فصل بعد، پس از بیان دقیق صورت مسئله‌ی مورد بررسی، روش دفاع نوین پیشنهادی ما بر مبنای روش پیشین ACGAN-ADA ارائه خواهد شد.



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 عنوانتابع  $f$  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 یک نمونه‌ی تخاصمی با ابعاد برابر تصویر ورودی تولید می‌کند که خروجی دسته‌بند را دستخوش تغییر می‌کند در حالی که تغییرات نمونه‌ی تخاصمی نسبت به نمونه‌ی اصلی ناچیز است. به عبارت دقیق‌تر، اگر  $x$  یک نمونه‌ی سالم باشد، هدف یک حمله تخاصمی را می‌توان طبق معادله‌ی (۱-۳) بیان کرد:

$$\arg \max_c f_\theta(x) \neq \arg \max_c f_\theta(A(f_\theta, x)), \quad \|x - A(f_\theta, x)\| \leq \epsilon \quad (1-3)$$

حال هدف ما پیاده‌سازی یک مکانیزم دفاعی با دو هدف تشخیص و پاکسازی - 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، بینانگذاری می شود. 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 می پردازیم.

#### ۱-۳-۳ استفاده از مولد های قوی تر

یکی از ایرادات مطرح شده توسط خود محققین ACGAN-ADA، کمبودهای ذاتی معماری ACGAN در مدل سازی آماری داده های 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 مولد های بهتر با قابلیت تولید نمونه های متعدد 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 می باشد. تاکید اصلی ما در این پژوهش بر استفاده از مدل های شرطی از نظریه ای ثابت شده در [۵۷] نشأت می گیرد که ادعا می کند مدل های مشروط از آنجایی که در ازای هر شرط باید احتمالاً توزیع احتمالی ساده تر را یاد بگیرند، از فرایند آموزش پایدارتری نسبت به مدل های بدون شرط برخوردار خواهند بود و به نقطه ای بهینه ای نزدیک تری به بهینه ای سراسری در فضای پارامترها، همگرا خواهند شد. این امر به مدل کمک می کند که علاوه بر یادگرفتن مفاهیم مرتبط با تنوع نمونه های تولید شده، نمونه های نزدیک تری به توزیع واقعی نمونه ها تولید کند.

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

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

$$S_R = D(x)$$

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

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

<sup>1</sup> Adversarial Robustness using Class-conditionAl geNerative modEls

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

۱.  $S_R$ : بزرگ بودن این مقدار می‌تواند به منزله‌ی بزرگ بودن میزان نویز تخاصمی تزریق شده به نمونه‌ی ورودی و مواجهه با یک نمونه‌ی تخاصمی ضعیف از لحاظ بصری تلقی شود. بنابراین انتظار می‌رود که برای نمونه‌های سالم مقدار  $S_R$  کوچک باشد.
۲.  $S_C$ : متناظر با احتمالیست که دسته‌بند جانبی ACGAN به کلاس تشخیص داده شده توسط دسته‌بند مورد حمله، نسبت داده است. اگر این احتمال نسبت به احتمال نظری دسته‌بند قربانی نباشد، احتمالاً با یک نمونه‌ی تخاصمی مواجه هستیم.
۳.  $S_g$ : اگر مولد بتواند به شرط کلاس تشخیص داده شده توسط دسته‌بند قربانی، نمونه‌ای تولید کند که نرم اختلاف آن با نمونه‌ی ورودی کوچک باشد بنابراین احتمالاً نمونه‌ی ورودی در توزیع آماری شرطی یادگیری شده توسط مولد قرار دارد و سالم است. 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 از سه معیار دیگر نیز برای تشخیص حمله بهره برده می‌شود:

۴.  $p_{victim}(\hat{c}|x)$ : سطح اطمینان خود دسته‌بند قربانی از تصمیمی که در رابطه با یک نمونه اتخاذ کرده است می‌تواند خود به عنوان معیاری برای تشخیص حملات استفاده شود. به عنوان مثال قطعیت بسیار بالا برای یک نمونه (مثلاً احتمال ۰.۹۹۸ در یک دسته‌بندی ۱۰ کلاسه) و یا قطعیت نسبتاً پایین (مثلاً ۰.۳ در همان دسته‌بندی ۱۰ کلاسه) می‌تواند نمایانگر یک حمله‌ی تخاصمی باشد.

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

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

<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 محاسبه  $S_g$  استفاده شد که نتیجه‌ی بهتری نسبت به MSE به تنهایی نداشت.

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

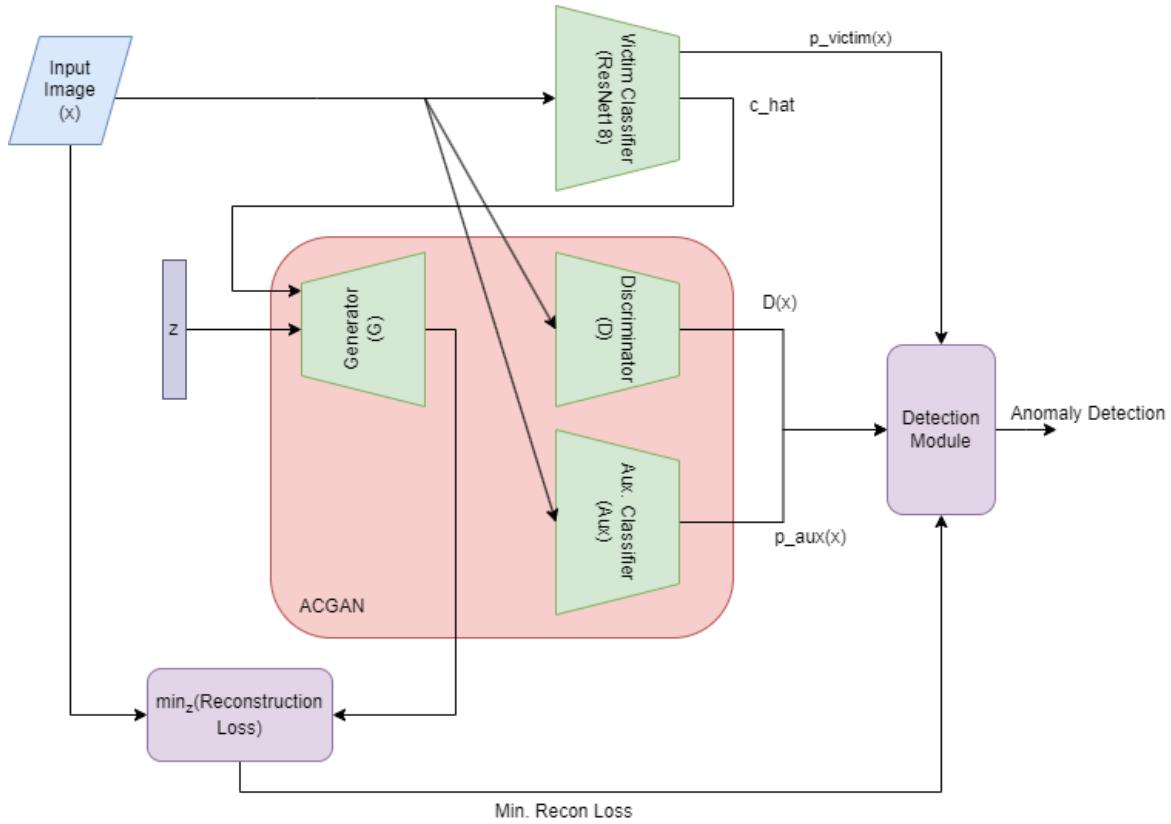
۶. احتمال توأم<sup>۱</sup> تقریبی دو معیار معرفی شده پیشین(با فرض ضعیف استقلال این دو توزیع) می‌تواند به عنوان معیار جدیدی برای تشخیص استفاده شود. از آنجایی که تصمیم اتخاذ شده بر اساس  $S_C$  و  $S_R$ ، تفسیر مشابهی در هر دو معیار دارد، این احتمال توأم می‌تواند معیاری بهتری نسبت به هر کدام از آن‌ها به تنها‌یی ارائه دهد.

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

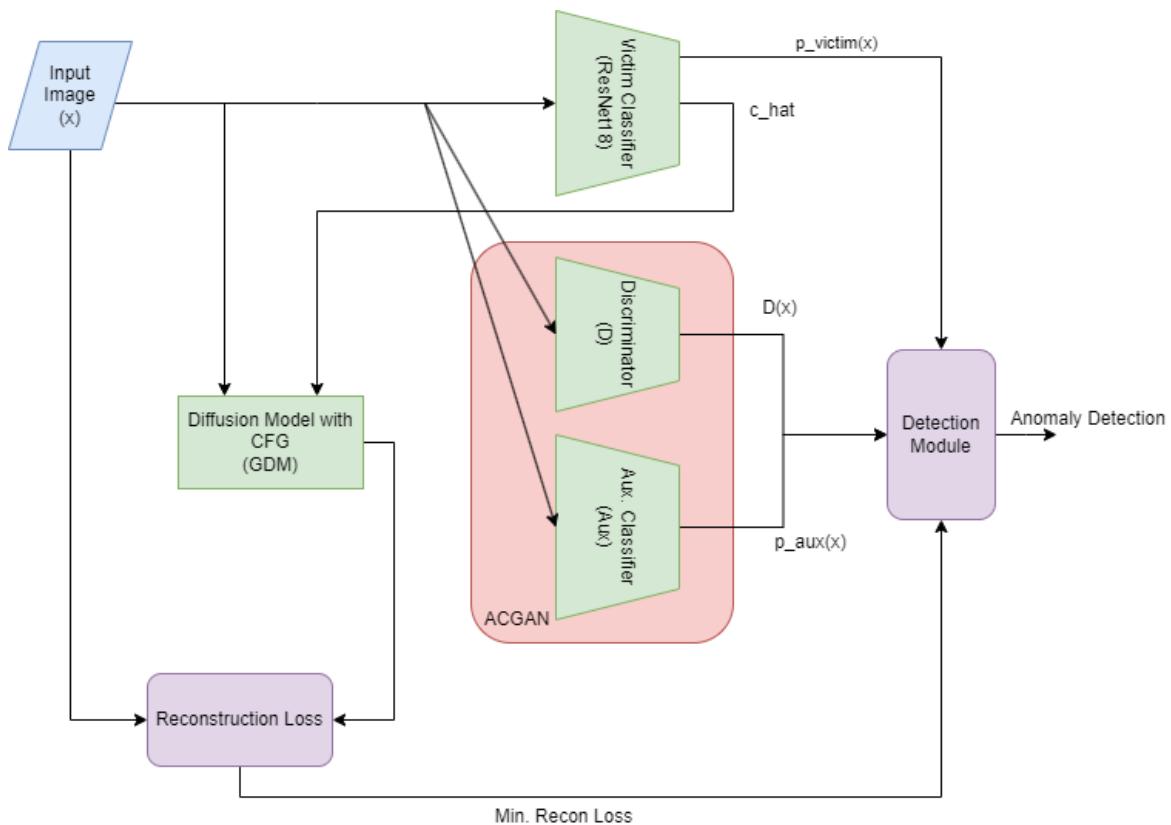
---

<sup>1</sup>Joint

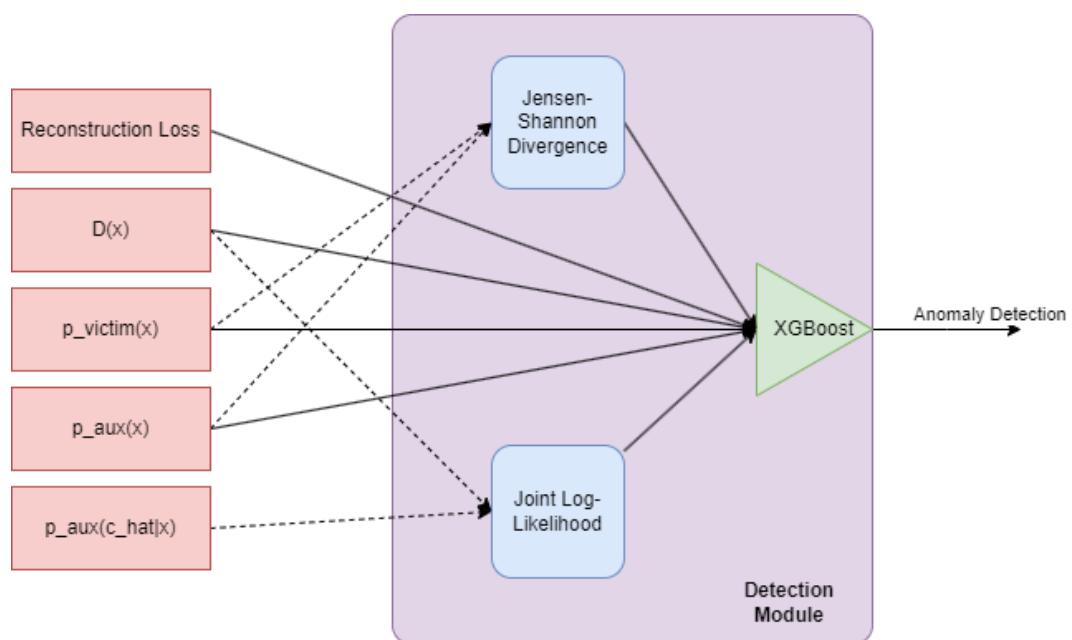
<sup>2</sup>Tabular



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 از 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 صحیح را تولید خواهد کرد.

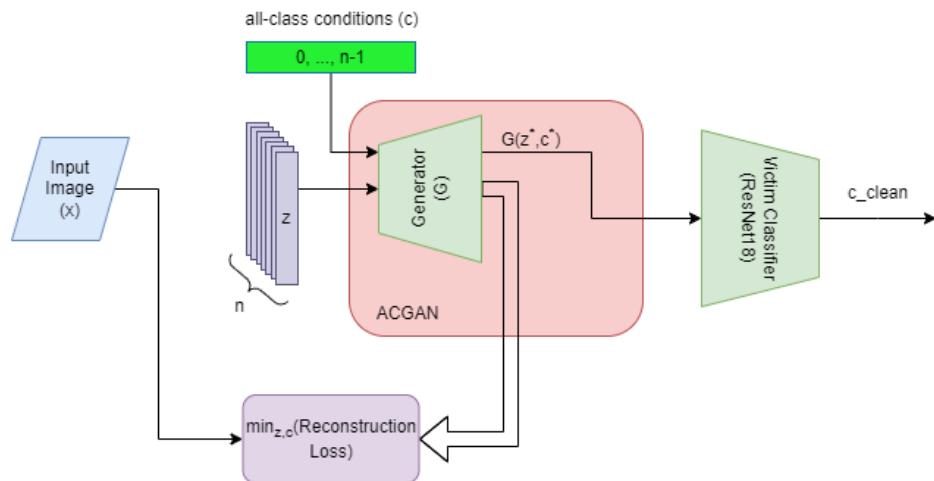
برای تولید تصاویر با استفاده از مولد ReACGAN از ایده‌ای شبیه Defense-GAN [۳۳] ولی به صورت شرطی بهره برد می‌شود. ابتدا یک بردار پنهان  $\mathbf{z}$  با ابعاد مورد انتظار مولد تولید می‌شود. سپس  $n$  (تعداد کلاس‌ها) کپی از این بردار به صورت همزمان ولی با کلاس‌های متناظر مختلف به مولد داده می‌شوند. سپس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

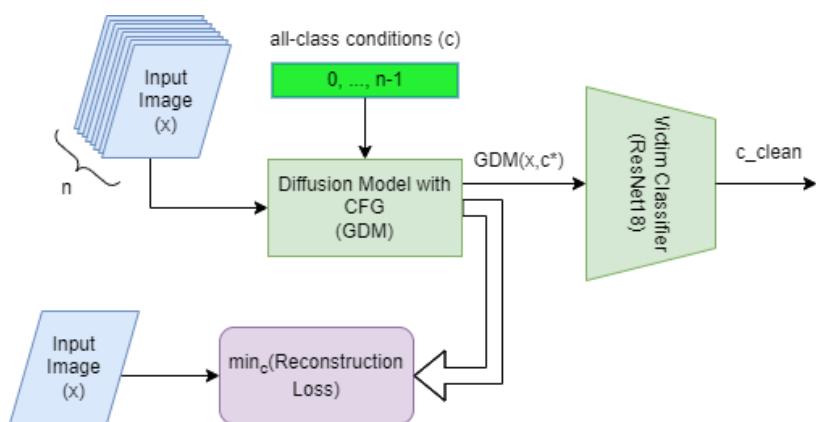
برای حصول اطمینان از تولید تصاویر با کیفیت از لحاظ ترکیب رنگ، فقط در مرحله‌ی پاکسازی، بهترین خروجی‌های هر دو مولد از یک فیلتر تطبیق هیستوگرام<sup>۱</sup> نیز عبور خواهند کرد تا مطمئن شویم ترکیب رنگ خروجی نهایی تا حد ممکن به تصویر ورودی نزدیک است. این امر باعث بهبود دقت پاکسازی می‌شود. چارچوب کلی پاکسازی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

در این فصل به بیان صورت دقیق مسئله و روش نوین ارائه شده برای حل آن پرداخته شد. در فصل بعد جزئیات پیاده‌سازی روش معرفی شده به همراه نتایج ارزیابی عملکرد آن مورد بررسی قرار خواهد گرفت.

<sup>1</sup>Histogram Matching



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



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

## ۱ - ۴ مقدمه

### فصل چهارم

#### شبیه سازی و نتایج ارزیابی

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 مقایسه‌ای بین روش کنونی و روش‌های ارائه شده پیشین مورد بحث قرار خواهد گرفت. در نهایت به ارائه نتایج چند مطالعه‌ی جانبی حول میزان تاثیر ویژگی‌ها و نیز دسته‌بند استفاده شده به عنوان تشخیص دهنده‌ی نهایی خواهیم پرداخت.

## ۲ - ۴ روش شبیه سازی

تمامی مدل‌ها و آزمایش‌های صورت گرفته روی بستر کتابخانه یادگیری عمیق معروف PyTorch<sup>۱</sup> و PyTorch با استفاده از زبان برنامه‌نویسی پایتون<sup>۲</sup> پیاده سازی شده اند. برای آموزش مدل ReACGAN از مخزن-StudioGAN<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.

<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 آموزش و آزمون مدل‌های مولد از مجموعه داده‌های CIFAR10 [۷۱] و Tiny-ImageNet [۷۲] استفاده شده است. در ادامه مشخصات کلی این دو مجموعه داده آورده شده است:

:CIFAR10 •

- شامل 60000 تصویر رنگی  $32 \times 32$
- 50000 نمونه برای آموزش و 10000 برای آزمون
- دارای 10 کلاس از اشیا و حیوانات و مناظر طبیعی
- از هر کلاس 6000 نمونه در اختیار است (5000 نمونه در مجموعه آموزشی و 1000 نمونه در مجموعه آزمون).

:Tiny-ImageNet •

- برگرفته شده از مجموعه داده معروف ImageNet [۷۳].
- شامل 110000 تصویر رنگی  $64 \times 64$  است که از کوچک کردن تصاویر وضوح بالای اصلی ImageNet بدست آمده اند.
- دارای 200 کلاس (و 550 تصویر از هر کلاس).
- به ازای هر کلاس، 500 نمونه آموزشی و 50 نمونه اعتبارسنجی در اختیار است.

برای محاسبه سنجه‌ها از دو کتابخانه Scikit-learn<sup>۱</sup> [۷۴] و TorchMetrics<sup>۲</sup> [۷۵] استفاده شده است. پیاده سازی مدل دسته‌بند نهایی برای تشخیص حمله با استفاده از کتابخانه معروف XGBoost [۶۸]<sup>۳</sup> انجام شده است. نهایتاً برای پیاده سازی حملات تخصصی از کتابخانه Torchattacks<sup>۴</sup> با کمی تغییرات، بهره برداری شده است.

در مقاله‌ی ACGAN-ADA<sup>۴</sup> [۴] برای آزمون مدل‌ها ابتدا یک دسته‌بند ResNet18 به عنوان قربانی در نظر گرفته شده است. سپس تمام نمونه‌هایی که از مجموعه داده‌های تست متناظر با هر دیتاست (CIFAR10 و Tiny-ImageNet) که توسط دسته‌بند قربانی به درستی دسته‌بندی شده اند، به عنوان مجموعه داده‌های تست نهایی مدد نظر قرار گرفته است. با توجه به زمان بر بودن محاسبات و نیز در دسترس نبودن وزن‌های مدل قربانی آموزش داده شده در ACGAN-ADA به ناچار برای آزمون مدل‌ها از تعداد مشخصی نمونه که در ادامه توضیح

<sup>1</sup><https://scikit-learn.org/stable/index.html>

<sup>2</sup><https://lightning.ai/docs/torchmetrics/stable//index.html>

<sup>3</sup><https://xgboost.readthedocs.io/>

<sup>4</sup><https://github.com/Harry24k/adversarial-attacks-pytorch>

داده خواهد شد، استفاده شده است. برای اطمینان از قابل تکرار بودن نتایج، دانه تصادفی<sup>۱</sup> استفاده شده برای نمونه برداری در ابتدای آزمایشات، تنظیم 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 تخصصی ایجاد شده است. به عنوان مثال، برای مجموعه داده 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 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 تخصصی) برای هر حمله گزارش شده است. همچنین، برای آموزش تشخیص دهنده‌ی نهایی XGBoost از ۱۰۰۰ نمونه‌ی سالم و تعداد مساوی نمونه‌ی تخصصی از هر نوع حمله استفاده شده است. برای حصول اطمینان از عادلانه بودن نتایج، مطمئن شده ایم که مجموعه داده 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>۲</sup> لایه با جستجوی شبکه‌ای روی فرآپارامترهای XGBoost انجام شده است و سپس مدل‌های دارای بهترین AUC انتخاب شده اند. به طور خلاصه، برای آموزش تشخیص دهنده‌ی نهایی به ازای هر جفت (نوع حمله، مجموعه داده) از مجموعاً ۲۰۰۰ نمونه‌ی آموزشی که شامل ۶ ویژگی و ۱ برچسب هستند استفاده شده است. در بخش ۵-۴ مطالعاتی در زمینه‌ی علت استفاده از دسته‌بند XGBoost و نیز تاثیر ویژگی‌های استفاده شده در دقت تشخیص دهنده‌ی نهایی انجام شده است. همچنین، در تمامی حملات، قربانی مورد حمله یک مدل دسته‌بند ResNet-18 بوده که روی مجموعه داده متناظر آموزش دیده است.

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>۳</sup> های استفاده شده برای هر مدل، می‌توانید به مخزن ARCANE مراجعه کنید.

<sup>1</sup>Random Seed

<sup>2</sup>Cross-validation

<sup>3</sup>Hyperparameter

<sup>4</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 [۷۶] Fréchet Inception Distance (FID) [۴۵] برای سنجش کیفیت تصاویر تولید شده، استفاده شده است. در ادامه به طور مختصر هر یک را مورد بررسی قرار خواهیم داد.

• **IS:** برای محاسبهٔ این معیار، که یکی از اولین معیارهای قدرتمند معرفی شده برای سنجش عملکرد مدل‌های مولد است، 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 مورد 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 فرشه<sup>۲</sup> بین توزیع بردار ویژگی‌های استخراج شده توسط دسته‌بند InceptionV3 از مجموعه‌داده‌ی واقعی و نمونه‌های ساختگی تولید شده توسط یک مدل مولد را محاسبه می‌کند. از آنجایی

<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

که این یک معیار فاصله است، بنابراین مقادیر کمتر برای آن نمایانگر نزدیک تر بودن توزیع مجموعه داده 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 ای فرشه فرم بسته‌ی (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 سنجه‌ی 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 قاعده‌ی 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، سنجه‌ی FID برای برقراری مقایسه‌ای بین مجموعه داده های واقعی و نمونه های ساختگی به کار می رود در حالی که سنجه‌ی IS یک آنالیز آماری صرف فقط روی نمونه های یک مجموعه از داده هاست (معمولًا نمونه های ساختگی) [۶۵].

در جدول ۱-۴ عملکرد تولید تصاویر مدل های Guided Diffusion Model (GDM) و ReACGAN 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 تسلیک 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-۰.۰۲) استفاده شده است. علت ذکر شده برای این انتخاب در مقاله‌ی ACGAN-ADA جزئی ۰.۲ (۰.۲۰ pAUC) اهمیت بیشتر تشخیص های صورت گرفته در نواحی ای از نمودار است که نرخ مثبت کاذب<sup>۲</sup> کم می باشد، چرا

<sup>۱</sup> Area Under Curve

<sup>۲</sup> False Positive

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

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

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

که نرخ مثبت کاذب کم در عمل برای یک تشخیص دهنده بسیار مطلوب است و نواحی دیگر نمودار ROC در مقام یک سیستم دفاعی، از اهمیت ناچیزی برخوردارند [۴]. به عبارت ریاضی اگر AUC به صورت زیر تعریف شده باشد:

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

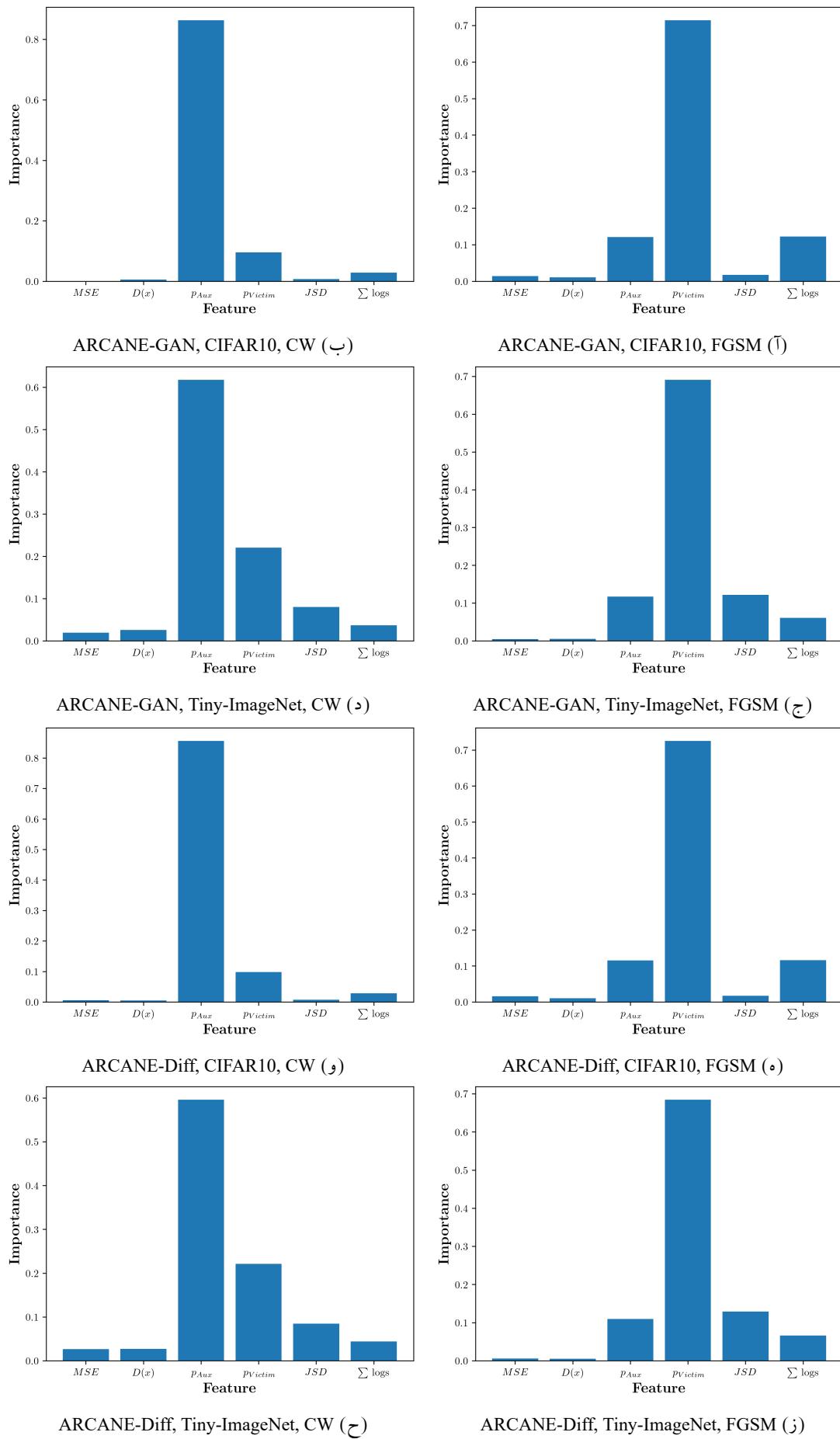
آنگاه  $pAUC-U$  به صورت زیر تعریف خواهد شد:

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

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

نتایج عملکرد تشخیص ARCANE-Diff و ARCANE-GAN در جدول ۴-۲ آورده شده اند.

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



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

جدول ۴-۳: 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



(ب) پاکسازی توسط ARCANE-GAN

(آ) پاکسازی توسط ARCANE-Diff

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

### ۴-۳-۳ سنجش عملکرد پاکسازی حملات

برای سنجش عملکرد پاکسازی از دقت پاکسازی استفاده شده است. در دو حالت عملیات پاکسازی موفقیت

آمیز تلقی می‌شود:

۱. نمونه سالم بوده و برچسب 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 متناظرشان 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 ۴-۲ 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 "قریباغه" آمده است.

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

#### ۴-۴ برسی و مقایسه نتایج

در ابتدا به بررسی عملکرد تشخیص حملات توسط ARCANE خواهیم پرداخت. نتایج 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 حمله‌ی FGSM، روشنگاری ARCANE-GAN با ۶.27٪ و ۲.58٪ بهبود برای حملات به ترتیب CW و ACGAN-ADA را روی مجموعه داده CIFAR10 شکست داده است.

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

CW	FGSM	
0.0576	0.0566	[۴۴] f-AnoGAN
0.0533	0.1642	[۴۶] KD
0.1042	0.1783	[۵۰] MD
0.0910	0.0436	[۵۱] ODDS
0.1489	0.1388	[۵۲] SID
0.1593	0.1782	[۴۸] ADA
0.1881	0.1819	<sup>۱</sup> [۴۹] ACGAN-ADA
<u>0.1999</u>	<u>0.1866</u>	ARCANE-GAN
<u>0.1999</u>	<u>0.1856</u>	ARCANE-Diff

در ادامه مطابق آزمایش‌های انجام شده در [۴] برای آزمون نتیجه‌ی تشخیص روی Tiny-ImageNet فقط روش‌های ARCANE، ACGAN-ADA، ADA، MD، KD، f-AnoGAN مقایسه شده‌اند. نتایج این مقایسه در جدول ۴-۵ 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 داده ۲۰۰ کلاس‌هی-Tiny-ImageNet مشهود تر است. در این مجموعه داده، ARCANE توانسته است روش ACGAN-ADA را با ۳۲.۷۵٪ بهبود به ترتیب روی حملات CW و FGSM شکست دهد.

در ادامه به مقایسه نتایج پاکسازی حملات خواهیم پرداخت. مطابق [۴]، مقایسه‌ای بین نتایج پاکسازی سه چارچوب ARCANE، Pix2Pix و ACGAN-ADA روی مجموعه داده CIFAR10 و تحت

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

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

CW	FGSM	
0.0571	0.0655	[۴۴] f-AnoGAN
0.0542	0.1168	[۴۷] KD
0.0918	0.1104	[۵۰] MD
0.1312	0.1385	[۴۸] ADA
0.1532	0.1496	[۴] ACGAN-ADA
<b>0.1913</b>	<u>0.1985</u>	ARCANE-GAN
<u>0.1912</u>	<b>0.1986</b>	ARCANE-Diff

جدول ۴-۶: مقایسه نتایج پاکسازی حملات روی مجموعه داده‌ی CIFAR10 و حمله‌ی CW

Purification Accuracy	
0.3274	[۳۳] Defense-GAN
0.8423	[۴] ACGAN-ADA
0.8632	ˇ[۷۷] pix2pix
<u>0.875</u>	ARCANE-GAN
<b>0.965</b>	ARCANE-Diff

حمله‌ی CW در جدول ۴-۶ 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 ۱۱.۷۹ درصدی نسبت به pix2pix منجر به دقت پاکسازی بهتری می‌شود. علت عملکرد بهتر ARCANE-Diff نسبت به ARCANE-GAN را می‌توان احتمالاً روش متفاوت ARCANE-Diff و نیز توانایی ذاتی مدل‌های انتشاری در از بین بردن نویز به واسطه طراحی و نحوه‌ی آموزش آن‌ها در نظر گرفت. همچنین به نظر می‌رسد که رابطه‌ی 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.

#### ۴-۵ مطالعات جانبی

در این بخش قصد داریم تاثیر نوع مدل دسته‌بند استفاده شده به عنوان تشخیص دهنده‌ی نهایی و نیز تاثیر هر کدام از ویژگی‌های مطرح شده برای کمک به تشخیص نهایی حملات را مورد بررسی قرار دهیم. ابتدا به بررسی تاثیر نوع دسته‌بند خواهیم پرداخت و سپس تاثیر ویژگی‌های جدید افزوده شده به عنوان معیارهای تصمیم‌گیری را بر تفکیک، بررسی خواهیم کرد.

<sup>۱</sup> به دلیل پر هزینه بودن انجام عملیات پاکسازی روی مجموعه داده‌ی 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 ۴-۴ مراجعه کنید.

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

## ۱-۵-۴ تاثیر نوع دسته‌بند در دقت تشخیص وجود حمله

برای بررسی تاثیر نوع دسته‌بند استفاده شده به عنوان تشخیص دهنده‌ی نهایی، مجموعاً ۱۱ دسته‌بند مطرح در حیطه‌ی یادگیری ماشین کلاسیک با فرآپارامترهای پیشنهادی از پیش تعیین شده برای آنها، روی ۱۶۰۰ نمونه‌ی آموزشی آموزش دیدند و روی ۴۰۰ نمونه‌ی اعتبارسنجی مورد مقایسه قرار گرفتند که در هر بار تکرار آزمایش برای تمام دسته‌بندها یکسان هستند ولی بین تکرارهای متوالی، مجدداً نمونه برداری شده اند. برای اطمینان از این که نتایج بدست آمده قابل اطمینان هستند، این آزمایش ۲۰ بار برای هر دسته‌بند تکرار شده و نتیجه‌ی نهایی به عنوان میانگین و انحراف معیار دقت هر دسته‌بند در مجموع این ۲۰ تکرار گزارش می‌شود.

دسته‌بندهای استفاده شده برای این آزمایش‌ها شامل موارد زیر هستند:

- **k-نزدیک ترین همسایه<sup>۱</sup>** [۷۸]: یکی از ساده‌ترین الگوریتم‌های یادگیریست که نمونه‌های زمان تست را در بر اساس برچسب اکثریت k نزدیک‌ترین نمونه‌ی همسایه در فضای ویژگی، دسته‌بندی می‌کند.
- **ماشین بردار پشتیبان خطی<sup>۲</sup>** [۷۹]: ماشین بردار پشتیبان خطی، تلاش می‌کند که مرز تصمیم بهینه بین دو کلاس از نمونه‌ها را با بیشینه کردن حاشیه‌ی بین این مرز تصمیم و هر یک از کلاس‌ها بدست بیاورد.
- **ماشین بردار پشتیبان با تابع پایه شعاعی<sup>۳</sup>**: یک ماشین بردار پشتیبان که از یک کرنل غیر خطی برای حل مسائل دسته‌بندی با مرزهای تصمیم پیچیده تر استفاده می‌کند.
- **دسته‌بند فرایند گاوی<sup>۴</sup>** [۸۰]: یک روش احتمالی و غیرپارامتریک کمتر شناخته شده که از فرایندهای گاوی برای تخمین توزیع احتمالی کلاس‌های موجود استفاده می‌کند.
- **XGBoost**: یکی از انواع الگوریتم‌های boosting که از درخت‌های تصمیم کم عمق به عنوان دسته‌بند پایه استفاده می‌کند. مانند دیگر روش‌های boosting در XGBoost نیز به ترتیب درختچه‌هایی به دسته‌بند نهایی اضافه می‌شوند تا خطاهای دسته‌بندی‌های پیشین را بپوشانند و در نهایت به دقت بالایی دست پیدا شود.
- **AdaBoost** [۸۱]: یکی از اولین الگوریتم‌های boosting که از سری کردن چندین دسته‌بند ضعیف که هر یک خطاهای مجموعه‌ی دسته‌بندهای پیش از خودشان را مرتفع می‌کنند تشکیل شده است. در این روش، به نمونه‌های به اشتباه دسته‌بندی شده وزن بیشتری در حین آموزش تعلق می‌گیرد که دسته‌بندهای بعدی بتوانند نمونه‌های دشوارتر را به درستی دسته‌بندی کنند.

<sup>1</sup>kNN

<sup>2</sup>Linear SVM

<sup>3</sup>RBF SVM

<sup>4</sup>Gaussian Process Classifier

- درخت تصمیم<sup>۱</sup> [۸۲]: یک الگوریتم ساده و قابل تفسیر که نمونه‌های ورودی را بر اساس مقادیر ویژگی‌ها به زیرمجموعه‌هایی با آنتروپی کمتر افزای می‌کند تا ساختاری درخت-شکل بدست آید که طی کردن مسیر از ریشه تا هر برگ، دسته‌ی هر نمونه را مشخص می‌کند.
  - جنگل تصادفی<sup>۲</sup> [۸۳]: یک روش یادگیری گروهی<sup>۳</sup> که از چندین درخت تصمیم با فراپارامترهای مختلف که هر کدام دسترسی به زیر مجموعه‌ای از تمام ویژگی‌های موجود را دارند، تشکیل می‌شود و نهایتاً پیش بینی‌های این درخت‌ها به نحوی با هم ترکیب می‌شوند تا برچسب نهایی یک نمونه‌ی ورودی مشخص شود.
  - شبکه عصبی<sup>۴</sup> [۸۴]: شبکه‌های عصبی، مدل‌های محاسباتی الهام گرفته شده از ذهن انسان هستند که از چندین لایه به هم متصل از نورون‌ها تشکیل می‌شوند. این شبکه‌های با استفاده از الگوریتم پساننتشار و نزول گرادیان قادر به یادگیری الگوهای پیچیده در نمونه‌های موجود هستند.
  - بیز ساده<sup>۵</sup> [۸۵]: یکی از ساده‌ترین روش‌های آماری دسته‌بندی که بر اساس قضیه‌ی معروف بیز بنیان‌گذاری شده است که فرض می‌کند بین ویژگی‌های مشاهدات موجود، استقلال احتمالاتی برقرار است.
  - تحلیل تفکیک مرتبه دو<sup>۶</sup> [۸۶]: یک دسته‌بند مولد که توزیع احتمالی مشروط بر کلاس داده‌ها را با استفاده از یک گاووسی چندمتغیره با ماتریس کواریانس جداگانه برای هر کلاس، تخمین می‌زند و سپس با در اختیار داشتن این تخمین، در زمان تست، برچسب نمونه‌های جدید را پیش بینی می‌کند.
- نتایج بدست آمده از آزمایشات برای ARCANE-Diff و ARCANE-GAN به ترتیب در شکل‌های ۳-۴ و ۴-۴ قابل مشاهده هستند. همچنین نتایج دقیق عددی برای این آزمایشات به ترتیب در جداول ۷-۴ و ۸-۴ قرار دارند. تمامی اعداد این جداول به چهارمین رقم اعشار با معنی رند شده‌اند.

<sup>1</sup>Decision Tree<sup>2</sup>Random Forest<sup>3</sup>Ensemble<sup>4</sup>Neural Network<sup>5</sup>Naive Bayes<sup>6</sup>Quadratic Discriminant Analysis

جدول ۴-۷: تاثیر نوع دسته‌بند در دقت نهایی تشخیص ARCANE-Diff

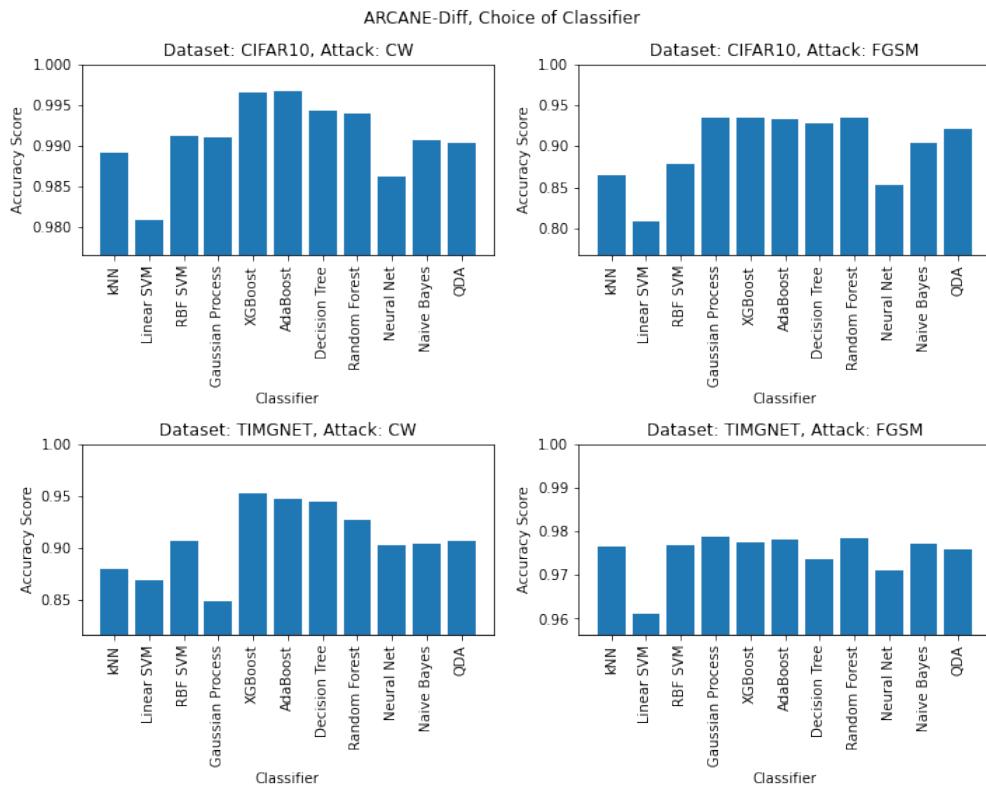
Tiny-ImageNet		CIFAR10		
FGSM	CW	FGSM	CW	
97.6375 ± 0.7136	88.025 ± 1.6084	86.4375 ± 1.7498	98.9125 ± 0.4491	<b>kNN</b> <b>Linear SVM</b> <b>RBF SVM</b> <sup>1</sup> <b>GP</b> <b>XGBoost</b> <b>AdaBoost</b> <b>Decision Tree</b> <sup>2</sup> <b>RF</b> <b>Neural Net</b> <b>Naive Bayes</b> <b>QDA</b>
96.125 ± 0.8043	86.9375 ± 2.0247	80.775 ± 1.3984	98.0875 ± 0.5549	
97.675 ± 0.554	90.7125 ± 1.8274	87.8625 ± 1.5562	99.1125 ± 0.3577	
<b>97.8625</b> ± 0.5888	84.8375 ± 15.113	<u>93.5</u> ± 1.0665	99.1 ± 0.3824	
97.7625 ± 0.5447	<b>95.275</b> ± 0.7022	<b>93.5625</b> ± 1.272	<u>99.65</u> ± 0.3905	
<u>97.825</u> ± 0.6713	<b>94.7375</b> ± 0.673	93.4 ± 1.1948	<b>99.675</b> ± 0.2385	
97.3625 ± 0.6914	94.5 ± 0.8101	92.775 ± 1.3012	99.4375 ± 0.3863	
97.8375 ± 0.6087	92.65 ± 1.6132	93.425 ± 1.1782	99.4 ± 0.4569	
97.1125 ± 0.5447	90.3125 ± 1.8026	85.325 ± 1.4472	98.625 ± 0.471	
97.7125 ± 0.6532	90.4 ± 1.5898	90.4 ± 1.171	99.0625 ± 0.4248	
97.6 ± 0.691	90.6125 ± 1.5702	92.15 ± 1.1657	99.025 ± 0.3437	

جدول ۴-۸: تاثیر نوع دسته‌بند در دقت نهایی تشخیص ARCANE-GAN

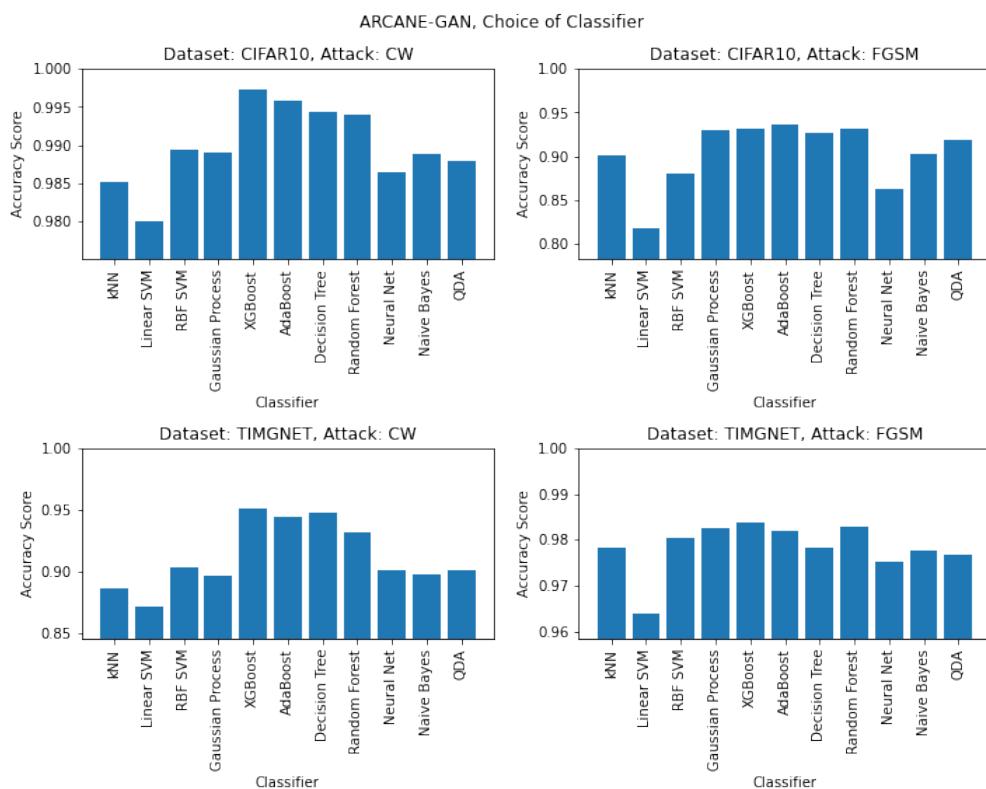
Tiny-ImageNet		CIFAR10		
FGSM	CW	FGSM	CW	
97.8375 ± 0.7386	88.6 ± 0.9467	90.0625 ± 1.0544	98.5125 ± 0.4435	<b>kNN</b> <b>Linear SVM</b> <b>RBF SVM</b> <b>GP</b> <b>XGBoost</b> <b>AdaBoost</b> <b>Decision Tree</b> <b>RF</b> <b>Neural Net</b> <b>Naive Bayes</b> <b>QDA</b>
96.375 ± 0.9503	87.1625 ± 1.1626	81.825 ± 1.5127	98.0 ± 0.6423	
98.05 ± 0.7525	90.35 ± 1.047	88.0875 ± 1.1626	98.9375 ± 0.5639	
98.2625 ± 0.7684	89.7 ± 4.6581	93.075 ± 1.1014	98.9 ± 0.443	
<b>98.3625</b> ± 0.7643	<b>95.1</b> ± 1.0012	<u>93.225</u> ± 0.8976	<b>99.725</b> ± 0.2487	
98.1875 ± 0.8905	<u>94.3875</u> ± 0.8309	<b>93.6</b> ± 0.9434	<u>99.5875</u> ± 0.309	
97.8375 ± 0.7553	94.75 ± 1.0983	92.7 ± 1.1141	99.4375 ± 0.4021	
<u>98.275</u> ± 0.6842	93.15 ± 1.5195	93.125 ± 0.731	99.4 ± 0.4359	
97.5125 ± 0.882	90.1125 ± 1.0322	86.2375 ± 1.226	98.6375 ± 0.5835	
97.775 ± 0.6796	89.8125 ± 0.8057	90.225 ± 1.1065	98.8875 ± 0.503	
97.6625 ± 0.5824	90.1625 ± 0.8226	91.8625 ± 1.2183	98.7875 ± 0.4891	

همانطور که از جداول بر می‌آید، دسته‌بندهای AdaBoost و XGBoost در هر دو چارچوب ARCANE-Diff بهترین عملکرد را از خود نشان داده‌اند. به دلیل کوچک‌تر بودن نسبی انحراف معیار<sup>۳</sup> ARCANE-GAN مربوط به نتایج XGBoost و نیز دقت عملکرد بالاتر آن در اکثر موارد نسبت به AdaBoost در این پژوهش از XGBoost به عنوان تشخیص دهنده‌ی نهایی استفاده شده است.

<sup>1</sup>Gaussian Process<sup>2</sup>Random Forest<sup>3</sup>انحراف معیار کوچک‌تر نوید یک دسته‌بند مقاوم‌تر در برابر بیش‌پرازش به نمونه‌های آموزشی را می‌دهد.



شکل ۴-۳: تاثیر نوع دسته‌بند در تشخیص وجود حمله در ARCANE-Diff



شکل ۴-۴: تاثیر نوع دسته‌بند در تشخیص وجود حمله در GAN

## ۴-۵-۲ تاثیر ویژگی‌های استفاده شده در بهبود دقت تشخیص

برای بررسی تاثیر ویژگی‌های استفاده شده در بهبود دقت تشخیص، با تثبیت XGBoost به عنوان تشخیص دهنده‌ی نهایی، ابتدا از ۳ ویژگی اولیه اصلی شروع کرده و سپس به تدریج ویژگی‌های جدید را می‌افزاییم و دقت تشخیص حملات را در هر مرحله می‌سنجیم. این آزمایشات برای ARCANE-Diff و ARCANE-GAN و برای هر جفت (مجموعه داده، نوع حمله) به تفکیک صورت گرفته‌اند. یادآوری می‌کنیم که در بخش ۳-۳-۲ ویژگی‌های جدید اضافه شده به موارد موجود در ACGAN-ADA معرفی شدند. این ویژگی‌ها عبارتند از:

$$\min_z \|x - G(z|\hat{c})\|^2 . \quad ۱$$

$$D(x) . \quad ۲$$

$$p_D(\hat{c}|x) . \quad ۳$$

$$p_{victim}(\hat{c}|x) . \quad ۴$$

$$\text{JSD}(p_D(x)||p_{victim}(x)) . \quad ۵$$

$$\log(p_D(\hat{c}|x)) + \log(D(x)) . \quad ۶$$

در جداول ۴-۹ و ۴-۱۰ نتایج آزمایشات انجام شده برای بررسی تاثیر ویژگی‌های ۴ الی ۶ در لیست ۴-۵-۴، بررسی شده‌اند. در ستون اول هر دو جدول، دقت تشخیص دهنده‌ی نهایی بدون استفاده از ویژگی‌های جدید آورده شده است. سپس در ستون‌های بعدی تاثیر افزودن ویژگی با شماره مطابق لیست ۴-۵-۴ به ویژگی‌های موجود نسبت به حالت قبلی نشان داده شده است و تمامی اعداد به چهارمین رقم اعشار با معنی گرد شده‌اند. همانطور که مشاهده می‌شود تاثیر ویژگی ۴ به طور متوسط از با اختلاف در تمامی حالات از دو ویژگی دیگر بیشتر است. ویژگی‌های ۵ و ۶ بسته به نوع حمله و مجموعه داده مورد بررسی تاثیرات مختلف اگرچه اندک را روی دقت نهایی می‌گذارند. برای ARCANE-Diff تاثیر ویژگی ۵ در حمله CW و مجموعه داده Tiny-ImageNet مثبت بوده است و نیز ویژگی ۶ در حمله CW تاثیر مثبت اندکی روی دقت نهایی داشته است. در ARCANE-GAN ویژگی ۵ دارای تاثیر اندک منفی روی دقت نهایی و ویژگی ۶ تقریباً بی تاثیر بوده است. نتایج کامل به تفکیک جفت‌های (مجموعه داده، نوع حمله) در مخزن پروژه<sup>۱</sup> قابل دسترسی هستند. یافته‌های این آزمایش، نتایج نمودارهای شکل ۴-۱ را تایید می‌کنند.

---

<sup>۱</sup><https://github.com/Adversarian/arcane>

جدول ۴ - تاثیر ویژگی‌های استفاده شده برای تشخیص حملات بر میزان دقت در ARCANE-Diff

$6 \leftarrow 5$	$5 \leftarrow 4$	$4 \leftarrow 3$	3	Global Avg.
<b>-0.1875(97.375)</b>	<b>+0.0(97.5625)</b>	<b>+7.25(97.5625)</b>	<b>90.3125</b>	<b>Avg. by Attack</b>
<b>+0.125(98.125)</b>	<b>+0.125(98.0)</b>	<b>+4.875(97.875)</b>	<b>93.0</b>	CW
<b>-0.5(96.625)</b>	<b>-0.125(97.125)</b>	<b>+9.625(97.25)</b>	<b>87.625</b>	FGSM
<b>Avg. by Dataset</b>				
<b>-0.375(97.5)</b>	<b>-0.25(97.875)</b>	<b>+7.0(98.125)</b>	<b>91.125</b>	CIFAR10
<b>-0.0(97.25)</b>	<b>+0.25(97.25)</b>	<b>+7.5(97.0)</b>	<b>89.5</b>	Tiny-ImageNet

جدول ۵ - تاثیر ویژگی‌های استفاده شده برای تشخیص حملات بر میزان دقت در ARCANE-GAN

$6 \leftarrow 5$	$5 \leftarrow 4$	$4 \leftarrow 3$	3	Global Avg.
<b>-0.062(96.25)</b>	<b>-0.1875(96.3125)</b>	<b>+6.8125(96.5)</b>	<b>89.6875</b>	<b>Avg. by Attack</b>
<b>+0.0(96.5)</b>	<b>-0.25(96.5)</b>	<b>+3.375(96.75)</b>	<b>93.375</b>	CW
<b>-0.125(96.0)</b>	<b>-0.125(96.125)</b>	<b>+10.25(96.25)</b>	<b>86.0</b>	FGSM
<b>Avg. by Dataset</b>				
<b>-0.125(96.375)</b>	<b>-0.25(96.5)</b>	<b>+6.625(96.75)</b>	<b>90.125</b>	CIFAR10
<b>+0.0(96.125)</b>	<b>-0.125(96.125)</b>	<b>+7.0(96.25)</b>	<b>89.25</b>	Tiny-ImageNet

در این فصل، نتایج آزمایش‌های انجام شده برای بررسی عملکرد چارچوب ARCANE مورد بررسی قرار گرفتند. در فصل بعد به طور مختصر به جمع‌بندی روش ارائه شده و بیان راستاهای پژوهش‌های آینده خواهیم پرداخت.

## 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 تصمیم گیرنده‌ی 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> [۸۷] به جای معیار MSE پر استفاده‌ی MSE پیشنهاد می‌شود. با وجود آن که محاسبه‌ی این معیار به مراتب هزینه‌بر تر از محاسبه‌ی 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 MSE‌های 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 Sample .....	نمونه تخاصمی
Adversarial Training .....	آموزش تخاصمی
Area Under Curve .....	سطح زیر نمودار
Attack Surface .....	سطح حمله
Attacker .....	مهاجم
Auto Trader .....	تاجر خودکار
Autoencoder .....	خودرمزنگذار
Auxiliary Classifier .....	دسته‌بند اضافی

## B

Backpropagation .....	پس انتشار
Binary Search .....	جستجوی دودویی

Black Box ..... جعبه سیاه .....

Box Constraints ..... محدودیت های جعبه ای

## C

Cascade ..... آبشاری .....

Class Conditional ..... مشروط بر کلاس .....

Classifier Guided Diffusion Model ..... مدل انتشاری هدایت شده با دسته بند .....

Classifiers ..... دسته بند ها .....

Confidence ..... اطمینان .....

Confidence Reduction ..... تقلیل اطمینان .....

Cross-validation ..... اعتبار سنجی ضربه دری .....

Cyber Attacks ..... حملات سایبری .....

## D

Decision Tree ..... درخت تصمیم .....

Deep Learning ..... یادگیری عمیق .....

Defensive Distillation ..... تقطیر دفاعی .....

Denoising Diffusion Probabilistic Model (DDPM) ..... مدل های انتشار احتمالی حذف کننده نویز .....

Detection ..... تشخیص .....

Diffusion Model ..... مدل انتشاری .....

Diffusion Model with Classifier-Free Guidance ..... مدل انتشاری با هدایت بدون دسته بند .....

Discriminator ..... ممیز .....

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

Ensemble ..... گروهی .....

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
Freeze .....	ثابت
Fully Connected .....	کاملاً متصل

## G

Gaussian Process Classifier .....	دسته بند فرایند گاووسی
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 .....	فراپارامتر

## J

Joint .....	توأم
-------------	------

**K**

Kernel Density Estimation .....	تخمین چگالی هسته‌ای
kNN .....	-نزدیک ترین همسایه k

**L**

Latent .....	نهفته
Learning Rate .....	نرخ یادگیری
Linear SVM .....	ماشین بردار پشتیبان خطی
Loss .....	ضرر

**M**

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

**N**

Naive Bayes .....	بیز ساده
Nash Equilibrium .....	تعادل نش
Neural Network .....	شبکه عصبی
Noise .....	نویز
Non-equilibrium Thermodynamics .....	ترمودینامیک غیرتعادلی
Norm .....	نرم

**O**

Online .....	بر خط
--------------	-------

**P**

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

**Q**

Quadratic Discriminant Analysis .....	تحلیل تفکیک مرتبه دو
---------------------------------------	----------------------

**R**

Random Forest .....	جنگل تصادفی
Random Initialization .....	آغاز تصادفی
Random Seed .....	دانه تصادفی
RBF SVM .....	ماشین بردار پشتیبان باتابع پایه شعاعی
Reconstruction Error .....	خطای بازسازی
Reference .....	مرجع
Reformer .....	اصلاح کننده
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 .....	تاب آوری .....
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 .....
Reformer .....	اصلاح کننده .....
Confidence .....	اطمینان .....
Cross-validation .....	اعتبار سنجی ضربدری .....
$\text{Distortion}(x, \hat{x}) = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2$ where $x, \hat{x} \in \mathbb{R}^n$ .....	اعوجاج .....
Forward Diffusion .....	انتشار پیش رو .....
Reverse Diffusion .....	انتشار معکوس .....

ب

Online .....	بر خط .....
Naive Bayes .....	بیز ساده .....

## پ

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

## ت

Tolerance .....	تاب آوری .....
Auto Trader .....	تاجر خودکار .....
Sentiment Analysis .....	تحلیل احساسات .....
Quadratic Discriminant Analysis .....	تحلیل تفکیک مرتبه دو .....
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 .....
Tensor .....	تنسور .....
Joint .....	توأم .....

## ث

Freeze .....	ثابت .....
--------------	------------

## ج

Tabular .....	جدولی .....
Binary Search .....	جستجوی دودویی .....
Grid Search .....	جستجوی شبکه‌ای .....
White Box .....	جعبه سفید .....
Black Box .....	جعبه سیاه .....
Random Forest .....	جنگل تصادفی .....

## 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 .....	خمنه .....
Autoencoder .....	خودرمزنگذار .....

## د

Random Seed .....	دانه تصادفی .....
Decision Tree .....	درخت تصمیم .....
Classifiers .....	دسته بند ها .....
Auxiliary Classifier .....	دسته‌بند اضافی .....
Gaussian Process Classifier .....	دسته‌بند فرایند گاوسی .....
Untargeted Misclassification .....	دسته‌بندی اشتباه غیر هدفمند .....

Targeted Misclassification .....	دسته‌بندی اشتباه هدفمند .....
Divergence .....	دیوژانس .....

**ذ**

Markov Chain .....	زنجیره مارکوف .....
--------------------	---------------------

**س**

Attack Surface .....	سطح حمله .....
Area Under Curve .....	سطح زیر نمودار .....
Metric .....	سنجه .....

**ش**

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

**ض**

Loss .....	ضرر .....
------------	-----------

**ف**

Mahalanobis Distance .....	فاصله ماهالانوبیس .....
Hyperparameter .....	فرآپارامتر .....
Fréchet .....	فریش .....
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 .....
Ensemble .....	گروهی .....

**ل**

Hidden Layers .....	لایه‌های پنهان .....
---------------------	----------------------

**م**

RBF SVM .....	ماشین بردار پشتیبان باتابع پایه شعاعی .....
Linear SVM .....	ماشین بردار پشتیبان خطی .....
Self-driving Car .....	ماشین خودران .....
Slack Variable .....	متغیر لنگی .....
False Positive .....	ثبت کاذب .....
Zero-sum .....	مجموع-صفر .....
Box Constraints .....	محدودیت‌های 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

## ه

Generative AI ..... هوش مولد

## ی

Unsupervised Learning ..... یادگیری بدون نظارت

Deep Learning ..... یادگیری عمیق

## K

kNN ..... -نزدیک ترین همسایه k

ΛΥ

## 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., Hondu, 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] Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database”, in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, (Miami, FL), pp. 248–255, IEEE, June 2009.
- [74] 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.
- [75] Nicki Skafte Detlefsen, Jiri Borovec, 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.
- [76] 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.

- [77] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A., “Image-to-Image Translation with Conditional Adversarial Networks”, Nov. 2018.
- [78] Cover, T. and Hart, P., “Nearest neighbor pattern classification”, *IEEE Transactions on Information Theory*, Vol. 13, pp. 21–27, Jan. 1967.
- [79] Vapnik, V. N., “The Support Vector method”, in *Artificial Neural Networks — ICANN'97* (Goos, G., Hartmanis, J., Van Leeuwen, J., Gerstner, W., Germond, A., Hasler, M., and Nicoud, J.-D., eds. ), Vol. 1327, pp. 261–271, Berlin, Heidelberg: Springer Berlin Heidelberg, 1997.
- [80] Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning, Cambridge, Mass: MIT Press, 2006.
- [81] Freund, Y. and Schapire, R. E., “A desicion-theoretic generalization of on-line learning and an application to boosting”, in *Computational Learning Theory* (Goos, G., Hartmanis, J., Leeuwen, J., Carbonell, J. G., Siekmann, J., and Vitányi, P., eds. ), Vol. 904, pp. 23–37, Berlin, Heidelberg: Springer Berlin Heidelberg, 1995.
- [82] Von Winterfeldt, D. and Edwards, W. *Decision Analysis and Behavioral Research*. Cambridge [Cambridgeshire] ; New York: Cambridge University Press, 1986.
- [83] Tin Kam Ho, “Random decision forests”, in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, Vol. 1, (Montreal, Que., Canada), pp. 278–282, IEEE Comput. Soc. Press, 1995.
- [84] Rosenblatt, F., “The perceptron: A probabilistic model for information storage and organization in the brain.”, *Psychological Review*, Vol. 65, No. 6, pp. 386–408, 1958.
- [85] Rish, I., “An empirical study of the naïve bayes classifier”, *IJCAI 2001 Work Empir Methods Artif Intell*, Vol. 3, Jan. 2001.
- [86] Ghojogh, B. and Crowley, M., “Linear and Quadratic Discriminant Analysis: Tutorial”, June 2019.
- [87] 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 Jun 29, 2024

1. Mohammad Hossein Manshaei, Prof. (Supervisor)
2. Alireza Basiri, Assistant Prof. (Examiner)
3. Abdolreza Mirzaei, Associate Prof. (Examiner)

Dr. Behzad Nazari, Department Graduate Coordinator

