

لَهُ مَا
يَرَى



دانشگاه صنعتی اصفهان
دانشکده مهندسی برق و کامپیوتر

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

پایان نامه کارشناسی ارشد مهندسی کامپیوتر - هوش مصنوعی و رباتیکز

آرین 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 رباتیک آقای آرین
تشکر

تحت عنوان

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

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

دکتر محمد حسین منشی

۱- استاد راهنمای پایان نامه

ایکس وای زی

۳- استاد داور

ایکس وای زی

۴- استاد داور

دکتر بهزاد نظری

سرپرست تحصیلات تکمیلی دانشکده

تشکر و قدردانی

قدردان راهنمایی های ارزنده استاد گرانقدر جناب آقای دکتر محمد حسین 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	
	فصل اول : مقدمه
۲	۱-۱ اهمیت مسئله
۳	۱-۲ ساختار گزارش
	فصل دوم : پیشینه پژوهش و مبانی هوش مولد
۴	۱-۲ مقدمه
۴	۲-۲ حملات تخصصی
۶	۱-۲-۲ دسته‌بندی حملات تخصصی
۶	۱-۲-۲-۱ رویه‌ی حمله
۱۰	۱-۲-۲-۲ FGSM
۱۲	۱-۲-۲-۳ CW
۱۴	۱-۲-۲-۴ PGD
۱۶	۳-۲ روش‌های دفاع در برابر حملات تخصصی
۱۶	۱-۳-۲ روش‌های پاکسازی حمله
۲۵	۲-۳-۲ روش‌های تشخیص حمله
۳۴	۴-۲ مختصری در مورد هوش مولد
۳۴	۱-۴-۲ شبکه‌های مولد تخصصی
۳۸	cGAN ۱-۱-۴-۲
۴۰	ACGAN ۲-۱-۴-۲
۴۱	ReACGAN ۳-۱-۴-۲
۴۲	۲-۴-۲ مدل‌های انتشاری
۴۴	۱-۲-۴-۲ ۱-۲ مدل‌های انتشاری هدایت شده (مشروط)

فصل سوم : پیشنهاد روشی نوین برای مقابله با حملات تخاصمی

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

فصل چهارم : شبیه سازی و نتایج ارزیابی

۵۵	۱-۴ مقدمه
۵۵	۲-۴ روش شبیه سازی
۵۷	۳-۴ نتایج آزمون مدل ها
۵۷	۳-۴-۱ سنجش عملکرد مدل های مولد
۵۸	۳-۴-۲ سنجش عملکرد تشخیص حملات
۶۱	۳-۴-۳ سنجش عملکرد پاکسازی حملات
۶۲	۴-۴ بررسی و مقایسه نتایج

فصل پنجم : نتیجه گیری و پیشنهادها

۶۳	۱-۵ نتیجه گیری
۶۳	۲-۵ پیشنهادها
۶۴	واژه نامه انگلیسی به فارسی
۷۰	واژه نامه فارسی به انگلیسی
۷۶	مراجع
۸۱	چکیده انگلیسی

فهرست شکل‌ها

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

۴۲	زنجیره مارکوف نظریک DDPM	۲۷-۲
۴۴	نمونه‌های تولید شده توسط DDPM	۲۸-۲
۴۵	۲۹-۲ نمونه‌های تولید شده توسط Classifier Guided Diffusion Model	
۴۶	۳۰-۲ نمونه‌های تولید شده توسط Classifier-Free Guided Diffusion Model	
۵۱	۱-۳ نحوه تشخیص حمله توسط ARCANE-GAN	
۵۱	۲-۳ نحوه تشخیص حمله توسط ARCANE-Diff	
۵۲	۳-۳ دورنمای ساختار تصمیم‌گیرنده‌ی نهایی مبتنی بر XGBoost	
۵۴	۴-۳ نحوه پاکسازی حمله توسط ARCANE-GAN	
۵۴	۵-۳ نحوه پاکسازی حمله توسط ARCANE-Diff	
۶۰	۱-۴ اهمیت ویژگی‌های استفاده شده در XGBoost	
۶۱	۲-۴ نمونه‌ای از پاکسازی‌های انجام شده توسط ARCANE	

فهرست جداول

۹	۱-۲ مقایسه حملات جعبه سیاه و جعبه سفید
۱۵	۲-۲ مقایسه حملات CW و PGD، FGSM
۵۸	۱-۴ عملکرد مدل های مولد روی CIFAR10 و Tiny-ImageNet
۵۹	۲-۴ عملکرد تشخیص ARCANE-Diff و ARCANE-GAN طبق معیار pAUC-0.2
۶۱	۳-۴ دقت پاکسازی ARCANE-Diff و ARCANE-GAN
۶۲	۴-۴ مقایسه عملکرد پاکسازی ARCANE با سایر روش ها روی مجموعه داده CIFAR10

فهرست اختصارات

C

cGAN	Conditional Generative Adversarial Network
CW	Carlini-Wagner

D

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

F

FGSM	Fast Gradient Sign Method
FID	Fréchet Inception Distance

G

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

I

IS	Inception Score
----------	-----------------

J

JS	Jensen-Shannon
----------	----------------

K

KL Kullback-Leibler

P

PGD Projected Gradient Descent

R

ROC Receiver Operating Characteristic

Y

YOPO You Only Propagate Once

چکیدہ

كلمات کلیدی: ۱- یادگیری ماشین، ۲- یادگیری عمیق، ۳- مدل های مولد^۱، ۴- دسته بند ها^۲، ۵- حملات تخصصی^۳

1 Generative Models

2 Classifiers

³Adversarial Attacks

فصل اول

مقدمه

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

۱-۱ اهمیت مسئله

امروزه یادگیری عمیق^۱ به عنوان ابزاری قدرتمند و بهینه برای حل گسترده‌ی وسیعی از مسائل پیچیده روزمره شناخته می‌شود که حل آن‌ها با روش‌های یادگیری ماشین سنتی بسیار دشوار و بعض‌اً غیر ممکن بود. در سال‌های اخیر، یادگیری عمیق چنان دستخوش پیشرفت‌های ژگرفی شده که اکنون قادر است در عدیده‌ای از اهداف یادگیری، از کارایی انسان نیز پیشی بگیرد. با توجه به گسترش روز افزون استفاده از هوش مصنوعی - و به خصوص یادگیری عمیق - در مصارف روزانه و صنایع، اهمیت اندیشیدن تسهیلاتی برای مقابله با حملات سایبری احتمالی به چنین سیستم‌هایی نیز به تبع دو چندان شده است. به عنوان مثال، در [۱] نشان داده شده است که می‌توان به یک سیستم ماشین خودران^۲ 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

¹Deep Learning

²Self-driving Car

داده شده است که در آن می‌توان این سیستم را در تشخیص علائم رانندگی دچار خطا کرد. این حمله در شکل ۱-۱ نشان داده شده است. همچنین در [۳] نمونه عملی یک حمله به سه نمونه الگوریتم تاجر خودکار^۱ حمله شده است که در طی آن سیستم تاجر خودکار مبتنی بر مدل‌های یادگیری ماشین دچار خطا در پیش‌بینی قیمت آینده یک سهم می‌شوند.

دسته خاصی از حملات سایبری^۲ اعمال پذیر روی دسته‌بند‌های مبتنی بر یادگیری ماشین، حملات تخاصمی هستند که تمرکز اصلی این تحقیق می‌باشد. در فصل ۲ به تفصیل راجع به این حملات توضیح داده خواهد شد.

۲-۱ ساختار گزارش

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



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

¹ Auto Trader

²Cyber Attacks

³Metric

فصل دوم

پیشینه پژوهش و مبانی هوش مولد

۱ - ۲ مقدمه

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

۲ - ۲ حملات تخاصمی

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

¹ Adversarial Perturbation

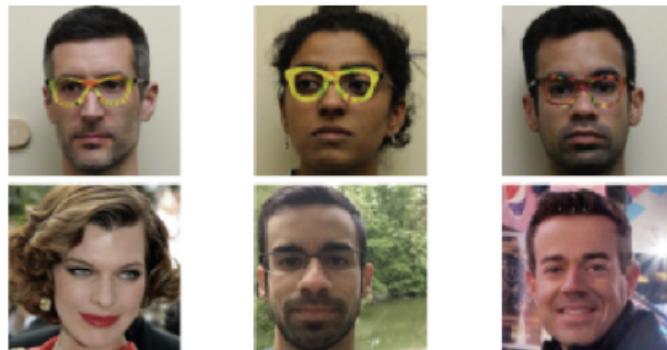
² Noise

³ Victim

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

به طور رسمی یک حمله تخصصی را می‌توان طبق تعریف ارائه شده در [۹] 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 ۲-۲ نمایش داده شده است.

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



شکل ۲-۱: نمونه‌ی حمله تخصصی به یک مدل تشخیص چهره [۸]. افراد سطر بالا به اشتباه 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 **eliehes clichs** here are limitless: the nerds vs. the jocks, the secret camera in the girls locker room, the hikers happening upon a nudist colony, the contest at the conclusion, the secretly horny camp administrators, and the **embarrassingly embarrassing1 foolish fo0l1sh** sexual innuendo littered throughout. This movie will make you laugh, but never intentionally. I repeat, never.

شکل ۲-۲: یک مدل دسته‌بند متن مبتنی بر CNN که با تغییرات جزئی در متن نظر ارسال شده، با وجود حفظ محتوای کلی نظر، دچار اشتباه شده است [۸].

¹Pixel

²Sentiment Analysis

اکنون یک نمونه تخاصمی^۱ به ورودی دستکاری شده‌ی \hat{x} گفته می‌شود که طبق معادلات ۲-۲ و ۳-۲ از افروden مقدار ناچیزی نویز^۵ 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^۲ ارزیابی می‌شود [۴، ۵]. پیش از پرداختن به دسته‌بندی حملات تخاصمی، ابتدا کمی راجع به مدل‌های تهدید^۳ در حوزه یادگیری ماشین با توجه به قدرت و دسترسی مهاجم صحبت خواهد شد.

۱-۱-۲-۲ رویه‌ی حمله

رویه‌ی حمله^۴ عبارتیست که به تمام روش‌های ممکن موجود برای حمله به یک سیستم اطلاق می‌شود. یک سیستم تصمیم‌گیرنده مبتنی بر یادگیری ماشین را می‌توان عملاً به عنوان یک خط لوله^۵ برای پردازش داده‌های ورودی متصور شد. بدین ترتیب، دنباله ای از عملیات ساده روی داده‌های ورودی در زمان استفاده از یک مدل یادگیری ماشین را 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 دنباله ای را می‌توان در شکل ۳-۲ مشاهده کرد.

در این خط لوله، سیستم ابتدا با استفاده از دوربین‌های نصب شده در اقصی نقاط خودرو، تصاویری را به عنوان ورودی دریافت می‌کند. این تصاویر طی عملیات پیش‌پردازش مناسب به فرمت قابل استفاده برای مدل

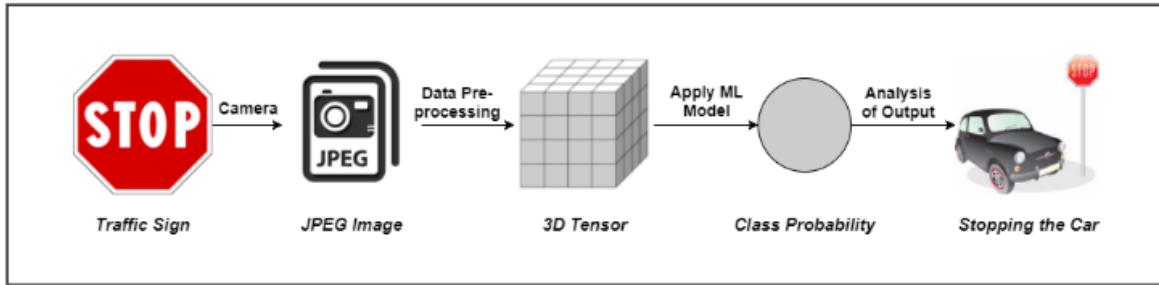
¹ Adversarial Sample

² Attacker

³ Threat Models

⁴ Attack Surface

⁵ Pipeline



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

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

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

۱. حمله گریزانه^۲: معمول‌ترین دسته‌ی حملات خصم‌مانه. در این حالت مهاجم در صدد گریز از یک سیستم امنیتی به وسیله‌ی اعمال تغییرات در نمونه‌های ورودی در زمان تست، بر می آید. در این حالت هیچ پیش‌فرضی در رابطه با دسترسی مهاجم به داده‌های آموزشی مدل، وجود ندارد.

۲. حمله مسموم‌کننده^۳: در این دسته از حملات که در زمان آموزش مدل قابل پیاده سازی هستند، مهاجم تلاش می کند که داده‌های آموزشی مدل را پیش از فرایند یادگیری، مسموم کند. به عبارت دقیق‌تر، مهاجم با افزودن نمونه‌های آموزشی جدیدی که با دقت و با هدف خاصی برای انحراف مدل آموزش دیده شده‌ی نهایی طراحی شده‌اند، به مجموعه داده‌های آموزشی، سعی می کند که کل فرایند آموزش مدل را مختل و یا خروجی آن را پس از اتمام آموزش دچار تغییرات نامطلوب نماید. واضح است که در این دسته از حملات فرض شده است که مجموعه داده آموزشی مدل قربانی در دسترس مهاجم قرار دارد.

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

¹Tensor

²Evasion Attack

³Poisoning Attack

⁴Exploratory Attack

⁵Black Box

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

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

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

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

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

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

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

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

¹White Box

²Feature Space

³Confidence Reduction

⁴Confidence

دیگر:

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

۲. دسته‌بندی اشتباه غیر هدفمند^۱: در این حالت، مهاجم در صدد تغییر خروجی مدل به هر کلاس $\hat{y} \neq y_{pred}$ برمی‌آید.

۳. دسته‌بندی اشتباه هدفمند^۲: در این حالت، مهاجم تلاش می‌کند خروجی مدل را به کلاس خاص مطلوب $y_{desired} \neq \hat{y}$ تغییر دهد.

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

جدول ۲ - مقایسه حملات جعبه سیاه و جعبه سفید

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

¹Untargeted Misclassification

²Targeted Misclassification

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

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

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

۲-۱-۲-۲ حمله FGSM

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

$$\nabla_x J(f(x; \theta), y) \tag{۵-۲}$$

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

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

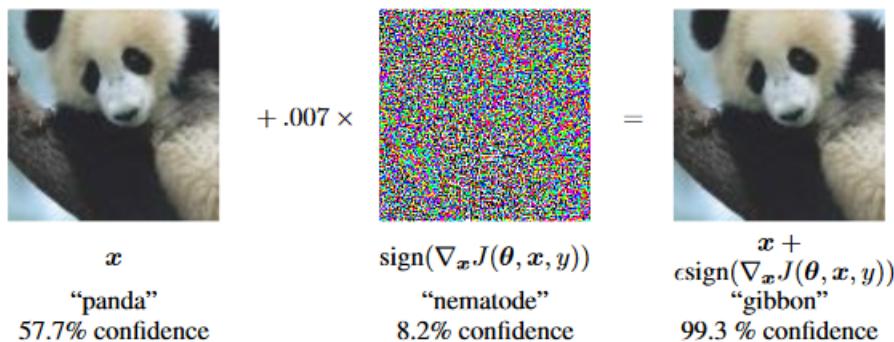
$$\hat{x} = x + \delta \quad (7-2)$$

که در آن مقدار ϵ که کنترل کننده اندازه نویز تخصصی بوده و بسته به مجموعه داده مورد استفاده و ضریب اطمینان لازم برای موفقیت حمله قابل تنظیم است. نویسندها در این پژوهش مقدار ۰.۰۰۷ را پیشنهاد کرده اند. به طور خلاصه در این حمله مقدار هر پیکسل از یک تصویر ورودی x به اندازه ϵ 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 غیرهادفمند است. اگر بخواهیم با استفاده از همین روش یک حمله هدفمند را با برچسب مطلوب $y_{desired}$ اجرا کنیم، کافیست که مقدار δ به شکل معادله ۸-۲ تغییر داده شود:

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

شکل ۴-۲ یک نمونه از حمله FGSM آورده شده است. قربانی این حمله یک شبکه GoogLeNet است و ورودی یک تصویر "پاندا" است که مدل آموزش دیده شده می تواند با سطح اطمینان ۵۷.۷٪ برچسب این تصویر را به درستی تشخیص دهد. اکنون با افزودن ۰.۰۰۷ از نویز تخصصی تولید شده - که خود با سطح اطمینان ۸.۷٪



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

توسط مدل به عنوان "کرم‌لوله‌ای" دسته‌بندی شده. تصویر نهایی که همچنان به چشم غیرمسلح مانند تصویر اولیه است، با سطح اطمینان ۹۹.۳٪ توسط مدل به کلاس "میمون دست‌دراز" تعلق گرفته است.

همانطور که مشخص است، این حمله در یک گام انجام می‌شود و یکبار محاسبه گرایان تابع هزینه برای تولید حمله کفایت می‌کند و بنابراین حمله FGSM بسیار سریع قابل پیاده‌سازی است. با وجود این که امروزه حملات جعبه‌سفید به مراتب قوی‌تری برای سنجش مدل‌های دسته‌بند موجود است، FGSM همچنان به عنوان یک روش آسان، قابل فهم و سریع برای آزمایش‌های اولیه مورد استفاده قرار می‌گیرد.

۳-۱-۲-۲ حمله CW

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

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

از فرمول‌بندی معادل ۱۰-۲ استفاده می‌شود:

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

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

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

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

^۱Norm

خواهد آمد:

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

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

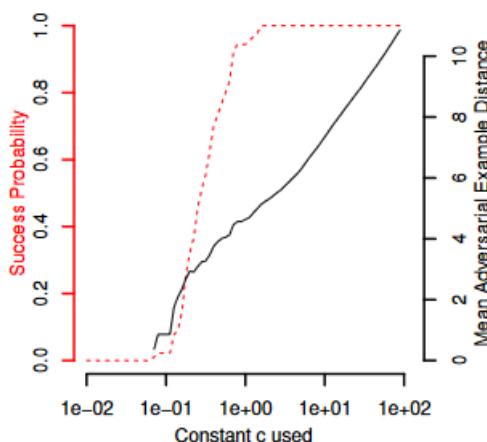
این حمله با نرم‌های L_0 ، L_2 و L_∞ قابل پیاده سازیست. از آنجایی که احتمال موفقیت حمله و نیز اندازه نرم نمونه تخاصمی بر حسب مقدار ثابت c استفاده شده، توابعی اکیداً نزولی هستند (طبق شکل ۵-۲)، می‌توان مقدار c بهینه را با استفاده روش جستجوی دودویی^۱ بدست آورد. شروطی مانند $x + \delta \in [0, 1]^n$ به نام محدودیت‌های جعبه‌ای^۲ معروف هستند. برای اجرا کردن این شرط، در حمله CW از یک تغییر متغیر استفاده می‌شود. به جای بهینه‌سازی مستقیم مقدار δ در معادله اصلی، متغیر جدید w معرفی می‌شود و مقدار

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

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

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

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



شکل ۲-۵: نمودار حساسیت موفقیت حمله و اندازه نیز تخاصمی بر حسب مقدار ثابت c [۱۴]

¹Binary Search

²Box Constraints

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

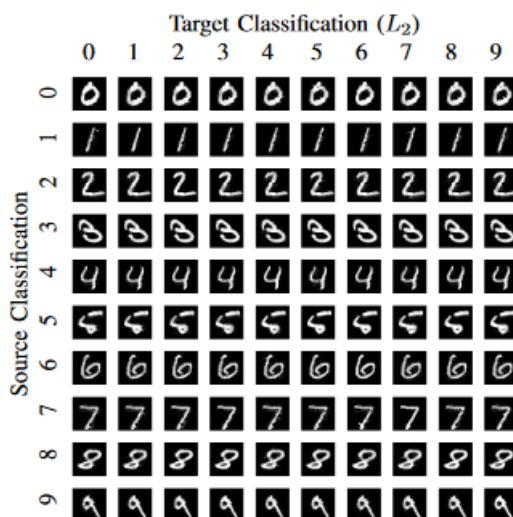
$$f(x) = \max(\max\{Z(x)_i : i \neq t\} - Z(x)_t, -\kappa)$$

و $Z(\cdot)$ خروجی های logit مدل دسته بند قربانی F هستند (به طوری که $F(x) = \text{softmax}(Z(x)) = y$) و پارامتر $\kappa > 0$ میزان سطح اطمینان حمله را مشخص می کند به طوری که مقادیر بزرگتر κ 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 [۱۶] حل می شود و در نهایت نمونه های تخاصمی به دست خواهد 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.

۴-۱-۲-۲ PGD حمله

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

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



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

که در آن وظیفه مهاجم حل مسئله بیشینه‌سازی درونی و وظیفه قربانی حل مسئله کمینه‌سازی بروونیست. اکنون اگر 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 فرمول‌بندی ۱۳-۲ در محدوده‌ی L_∞ حول یک نمونه‌ی سالم عمل کند. اکنون اگر بخواهیم این مسئله را با استفاده از نزول گرادیان^۱ حل کنیم، کافیست همین گام معرفی شده را چندین بار تکرار کنیم و حاصل را روی ϵ -کرهٔ حول نمونه‌ی سالم بیافکنیم. این روش بهینه‌سازی با محدودیت که مبنای حمله‌ی PGD است، نزول گرادیان افکنده^۲ نام دارد. به طور دقیق‌تر اگر بخواهیم همچنان مانند حمله FGSM از نرم L_∞ بهره بگیریم، تشکیل یک حمله تخاصمی در PGD از تکرار گام زیر به‌دست خواهد آمد:

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

که در آن $\Pi_{x+\mathcal{S}}$ نشان دهنده عملیات افکنش روی ϵ -کره‌ی S + x بوده و این‌بار α نمایانگر نرخ یادگیری^۳ نزول گرادیان افکنده است. نکته قابل توجه در این رابطه آن است که عملیات افکنش حول ϵ -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.

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 همنچنان موثر	قوت حمله
کند تر، زیرا علاوه بر چندین گام نزول گرادیان، نیاز به انجام جستجوی دودویی برای بهینه‌سازی مقدار یکی از پارامترها است.	کند، چون به چند گام از نزول گرادیان افکنده احتیاج است	بسیار سریع، چراکه حمله در یک گام صورت می‌پذیرد	زمان اجرای حمله
پیچیده تر، علاوه بر استفاده از تابع هزینه مخصوص، به پیاده سازی جستجوی دودویی نیز احتیاج است	نسبتاً پیچیده، نیاز به اعمال چندین گام نزول گرادیان افکنده است که کمی دشوارتر از نزول گرادیان معمولیست	ساده، فقط نیاز به یکبار محاسبه گرادیان تابع هزینه مدل نسبت به ورودی اعمال شده به مدل است	پیچیدگی پیاده سازی حمله
جهبه سفید	جهبه سفید	جهبه سفید	نوع دسترسی مهاجم

¹Gradient Descent

²Projected Gradient Descent

³Learning Rate

۳-۲ روش‌های دفاع در برابر حملات تخاصمی

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

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

۲. تشخیص^۲: 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.

۱-۳-۲ روش‌های پاکسازی حمله

در ادبیات مرتبط با دفاع در برابر حملات تخاصمی، عبارت پاکسازی به دسته خاصی از روش‌های ممکن برای دفاع اطلاق می‌شود. در این پژوهش، برای سهولت دسته‌بندی، کمی از تعریف رسمی پاکسازی تخاصمی^۳ دور شده‌ایم و آن را مطابق تعریف ارائه شده در بخش ۳-۲ در نظر می‌گیریم. بنابراین پاکسازی تخاصمی خود در این دسته قرار خواهد گرفت. با این مقدمه، روش‌های پاکسازی را می‌توان در ابعاد زیر دسته‌بندی نمود [۱۰، ۶، ۵]:

۱. آموزش تخاصمی^۴: یکی از ابتدایی ترین روش‌های دفاع که می‌تواند قابل اعمال روی هر شبکه عصبی مورد حمله باشد. در [۱۳] پیشنهاد شده است که انجام فرایند آموزش روی ترکیبی از نمونه‌های سالم و نمونه‌های تخاصمی می‌تواند باعث منظم‌سازی^۵ شبکه عصبی آموزش دیده و پرورش توانایی برای مقابله در برابر حملات تخاصمی باشد. در [۹] 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 [۱۵] روشی نوین برای آموزش تخاصمی ارائه شده است که در

¹Purification

²Detection

³Adversarial Purification

⁴Adversarial Training

⁵Regularization

آن تابع ضرر^۱ به فرم یک مسئله بهینه‌سازی کمینه-بیشینه^۲ بیان شده است (رابطه ۱۳-۲) در این رابطه، مسئله بیشینه‌سازی درونی سعی می‌کند که شبیه فرمول‌بندی ارائه شده در بخش ۲-۱-۲-۲ برای حمله FGSM، قوی‌ترین مقدار^۳ی ممکن برای بزرگ کردن مقدار تابع هزینه پیدا شود. این در حالیست که در مسئله کمینه‌سازی برونوی هدف کمینه کردن مقدار این تابع ضرر تخاصمی با تنظیم کردن پارامترهای مدل (θ) است. بدین ترتیب، اگر مقدار تلورانس^۴ حمله را در نظر بگیریم، مدل با نمونه‌های تخاصمی در یک^۵-کره حول هر نمونه‌ی سالم آموزش داده خواهد شد.

مهم‌ترین ایراد آموزش تخاصمی این است که تولید نمونه‌های تخاصمی قوی در زمان آموزش -خصوصاً روی مجموعه داده‌های بزرگ مانند ImageNet - می‌تواند بسیار زمان بر باشد و بدین ترتیب اکثر روش‌های آموزش تخاصمی از حملات تک-مرحله‌ای مانند FGSM برای افزایش مجموعه داده‌های آموزشی استفاده می‌کنند. برای حل این مسئله روش Free Adversarial Training [۱۷] ارائه شد که از اطلاعات گرادیان مدل در زمان آموزش برای تولید حملات تخاصمی (PGD) استفاده می‌کند و بنابراین نیازمند تولید مجدد حمله نیست. همچنین در این روش، برای کاهش زمان همگرایی حمله PGD نویز تخاصمی بدست آمده برای یک دسته از ورودی‌های سالم، به عنوان نقطه آغازین حمله PGD روی دسته ورودی‌های بعد مورد استفاده قرار می‌گیرد. این دفاع روی حمله چند-مرحله‌ای PGD [۱۵] مورد آزمایش قرار گرفته و نتایج امیدوار کننده‌ای داشته است. همچنین در (YOPO) You Only Propagate Once [۱۸] این مسئله مورد بررسی قرار گرفته که نیاز به یک لایه دفاعی در برابر حملات تخاصمی را می‌توان تقریباً فقط به اولین لایه‌ی یک شبکه عمیق خلاصه کرد. بنابراین آموزش تخاصمی می‌تواند بسیار ارزان‌تر صورت بگیرد. این روش که مستقیماً با Free Adversarial Training مورد مقایسه قرار گرفته است، نشان می‌دهد که می‌تواند در زمانی کمتر دارای عملکرد مشابه باشد. از دیگر ایده‌های جالب توجه مطرح شده در این زمینه می‌توان به (GI-AT) Geometry-aware Instance-reweighted Adversarial Training [۱۹] و

Adversarial Training [۲۰] اشاره کرد. در GI-AT از این نکته بهره گرفته می‌شود که نمونه داده‌هایی که نزدیک مرزهای تصمیم‌گیری یک مدل قرار دارند، بیشتر می‌توانند در موفقیت یک حمله تخاصمی تاثیرگذار باشند. بدین ترتیب این روش با بهره‌گیری از راهکارهای استاندارد آموزش تخاصمی، هر یک از نمونه داده‌های آموزشی را بر مبنای این که تولید حمله‌ی تخاصمی موفق از روی آن‌ها چقدر دشوار است، وزن‌دهی کرده و در فرایند پساننتشار^۴ دخیل می‌کند. نهایتاً در Fast Adversarial Training این

¹Loss²Min-max³Tolerance⁴Backpropagation

نکته به صورت تجربی نشان داده می شود که آموزش تخاصمی با استفاده از حمله تک-مرحله‌ای FGSM ولی با آغاز تصادفی^۱ (بر خلاف [۱۷]) می تواند به اندازه آموزش تخاصمی با استفاده از حملات قوی تر چند-مرحله‌ای (مانند PGD) اثر بخش باشد. در این پژوهش محققین موفق شده‌اند که در کسری از زمان گزارش شده در [۱۷]^۲ به نتیجه مشابه دست بیابند.

۲. تغییرات در مدل قربانی و فرایند آموزش: پژوهش [۲۱] یکی از اولین کارهای انجام شده در این زمینه که با فاصله بسیار اندکی از کشف حملات تخاصمی و آسیب‌پذیری شبکه‌های عصبی به این حملات، صورت گرفت. با وجود این که حملات تخاصمی ایجاد شده پس از این مقاله، می توانند به راحتی راهکارهای ارائه شده در این کار را دور بزنند، ایده‌های مطرح شده همچنان شایان ذکر به نظر می‌رسند. ایده‌ی اصلی این پژوهش ارائه روش‌های آموزش جدید به گونه‌ایست که نمونه‌های تخاصمی تولید شده توسط حمله L-BFGS [۱۳] دارای اعوجاج^۳ بیشتری نسبت به نمونه‌های متناظر سالم دارند و دیگر قابل صرف نظر نیستند. در این پژوهش سه روش پیش پردازش برای مقابله با حمله L-BFGS مورد بررسی قرار گرفته‌اند:

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

(ب) خودرنگذار^۴: در این روش یک خودرنگذار با هدف بازسازی نمونه‌های سالم از روی نمونه‌های تخاصمی، آموزش داده شده است که می‌تواند از حمله جلوگیری کند.

(ج) خودرنگذار نویزگیر: در این روش، شبیه روش قبلی، یک خودرنگذار نویزگیر معمولی بدون داشتن پیشین از توزیع آماری نویز تخاصمی 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 σ ترکیب می‌شود. نشان داده شده است که با قرار دادن $0.1 = \sigma$ این خودرنگذار می‌تواند به خوبی خودرنگذار قبلی عمل کند.

ایراد اصلی وارد به موارد ۲ ب و ۲ ج این است که با سری کردن خودرنگذار و مدل قربانی و حمله به مدل سری شده، همچنان می‌توان حمله موفقی داشت. این در حالیست که روش ۲ آ 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^۴ [۲۲] است. همانطور که در شکل ۷-۲

^۱ Random Initialization

^۲ Distortion(x, \hat{x}) = $\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2$ where $x, \hat{x} \in \mathbb{R}^n$

^۳ Autoencoder

^۴ Defensive Distillation

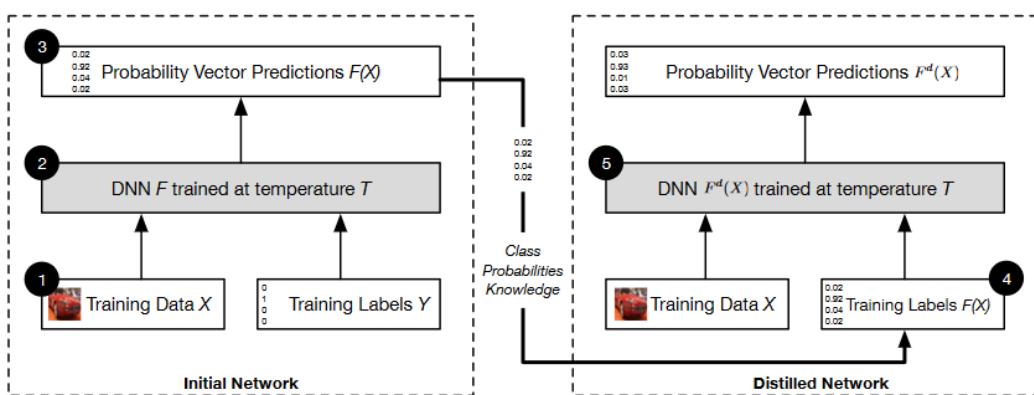
نشان داده شده است، ایده‌ی اصلی در این روش استفاده از پیش‌بینی‌های یک مدل از پیش‌آموزش داده شده روی مجموعه‌داده‌های سالم، به عنوان برچسب‌های جدید مدل تقطیر شده است. بدین ترتیب، مدل تقطیر شده برخلاف مدل اصلی دارای برچسب‌هایی با مقادیر پیوسته هستند که این امر آن‌ها را نسبت به حملات تخاصمی مقاوم‌تر می‌سازد. این روش در برابر حمله CW نتوان 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 [۲۳-۲۵].

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

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

که در آن y و \hat{y} به ترتیب برچسب حقیقی و پیش‌بینی مدل و λ پارامتر تنظیم کننده میزان جریمه منظمسازی می‌باشد. هدف این روش (که با افزودن جمله منظمساز شامل $\nabla_x H(y, \hat{y})$ محقق می‌شود) این است که از موضوع اطمینان حاصل شود که در صورت ایجاد تغییرات اندک در یک نمونه ورودی، دیورژانس KL بین پیش‌بینی مدل و برچسب واقعی تغییر چندانی نخواهد کرد و بنابراین نمونه‌های تخاصمی که در یک ۶-کره محدود می‌شوند، نخواهند توانست خروجی مدل قربانی را تغییر دهند.

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



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

صورت می‌گیرد. قوت این روش در برابر حمله FGSM به صورت تجربی نشان داده شده است.

در ایده‌ای مشابه پژوهش قبلی و ترکیب آن با [۲۱] ، در [۲۸] Random Self-Ensemble یک لایه نویز برای افروzen مقدار ناچیزی نویز ایزوتروپیک گاوی به ورودی‌های مدل و نیز خروجی‌های لایه‌های پنهان درون مدل قرار داده می‌شود:

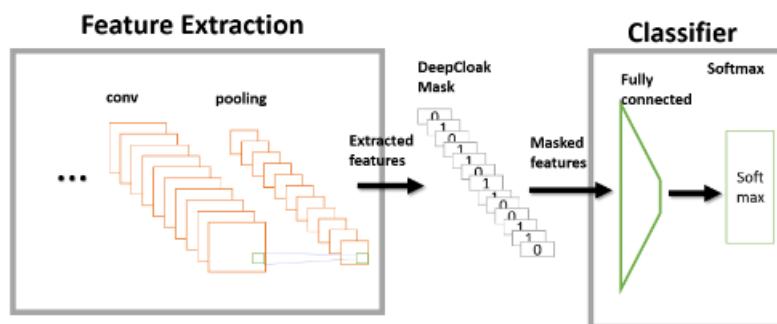
$$\text{NoiseLayer}(x) \rightarrow x + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2 I).$$

مقادیر بزرگتر σ در این لایه مقاومت در برابر حملات تخاصمی را در ازای افت دقت دسته‌بندی، افزایش می‌دهند. برای کاهش تاثیر منفی σ از این لایه هم در زمان آموزش (با استفاده از تکنیک reparameterization) و هم در زمان تست، استفاده می‌شود. سپس، با تغییر مقدار σ - و به تبع آن، مقدار نویز افزوده شده به خروجی لایه‌ها، ϵ - 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 [۲۹] دو علت اصلی برای اشتباه دسته‌بندها در مواجهه با یک نمونه تخاصمی برشمرده شده است:

(آ) نمونه تخاصمی از مزهای خمینه^۱ وظیفه مورد نظر دور است. به عنوان مثال اگر وظیفه تشخیص اعداد دست نویس و مجموعه‌داده MNIST را در نظر بگیریم، یک نمونه تخاصمی ممکن است تصویری باشد که اصلاً شامل یک عدد دست‌نویس نیست ولی دسته‌بند از آنجایی که مجبور به تولید خروجی است، دچار اشتباه خواهد شد.

(ب) نمونه تخاصمی به مزهای خمینه وظیفه مورد نظر خیلی نزدیک است. در این حالت که اکثر



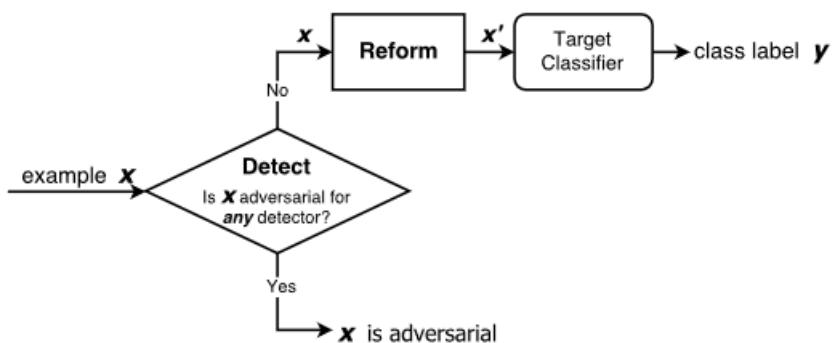
شکل ۲-۸: نحوه عملکرد [۲۷] DeepCloak

^۱Manifold

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

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

یکی از مهم‌ترین ایده‌های مطرح شده در زمینه استفاده از شبکه‌های جانبی چارچوب Defense-GAN^۲ است. در این پژوهش برای پاکسازی یک نمونه تخصصی از یک شبکه مولد تخصصی^۳ بهره برده می‌شود. در این چارچوب که در شکل ۱۰-۲ نمایش داده شده است، ابتدا یک WGAN روی مجموعه‌ای از داده‌های سالم آموزش داده می‌شود^۴. 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



[۲۹] MagNet نحوی عملکرد

^۱Reformer

^۲Generative Adversarial Network (GAN)

^۳برای توضیحات مقدماتی راجع به GAN ها، می‌توانید به بخش ۱-۴-۲ مراجعه کنید

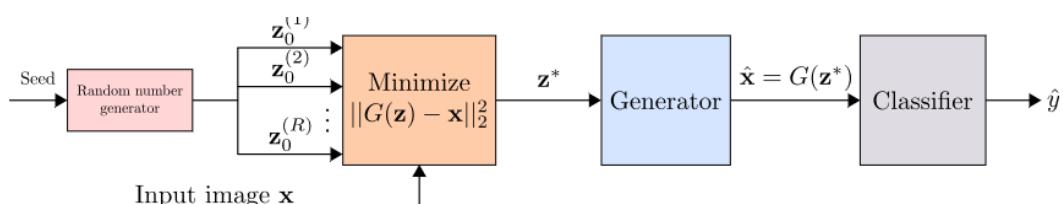
مولد، افکنده می‌شود. به عبارت دقیق‌تر، ابتدا R نمونه بردار نهفته^۱ به صورت تصادفی تولید می‌شوند.

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

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



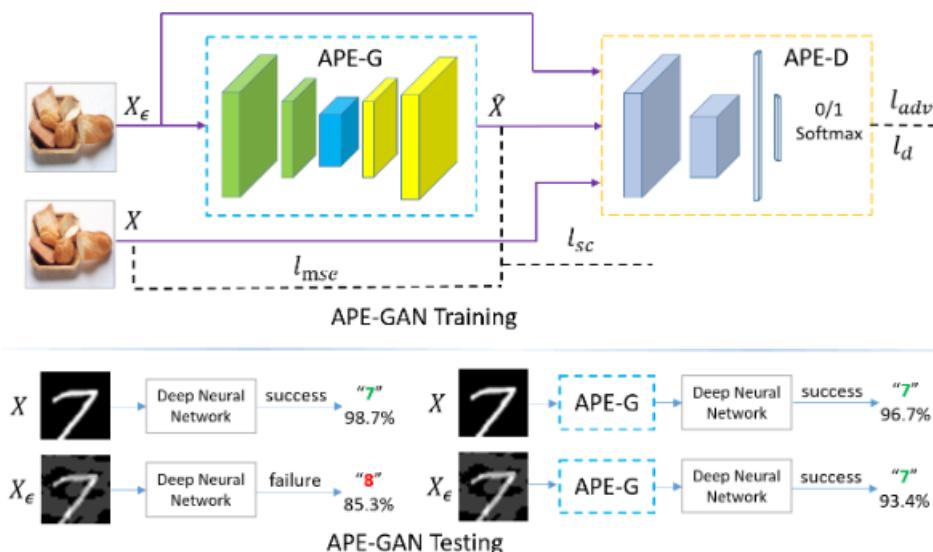
شکل ۱۰-۲ : دورنمای عملکرد چارچوب [۳۰] Defense-GAN

¹Latent

²Adversarial Perturbation Elimination Generative Adversarial Network

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

در [۳۲] استفاده از یک دسته‌بند با Embedding Regularized Classifier (Embedding منظم شده) 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 ترغیب 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 زیرشبکه متصور شد: یک Encoder که وظیفه آن استخراج ویژگی از ورودی دسته‌بند است و یک هد دسته‌بند برای تولید خروجی کلاس نهایی. اکنون در این چارچوب توزیع بردارهای نهفته تولید شده توسط شبکه Encoder با یک توزیع پیشین^۱ مقایسه می‌شوند و توسط یک شبکه جانبی ممیز^۲ 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.

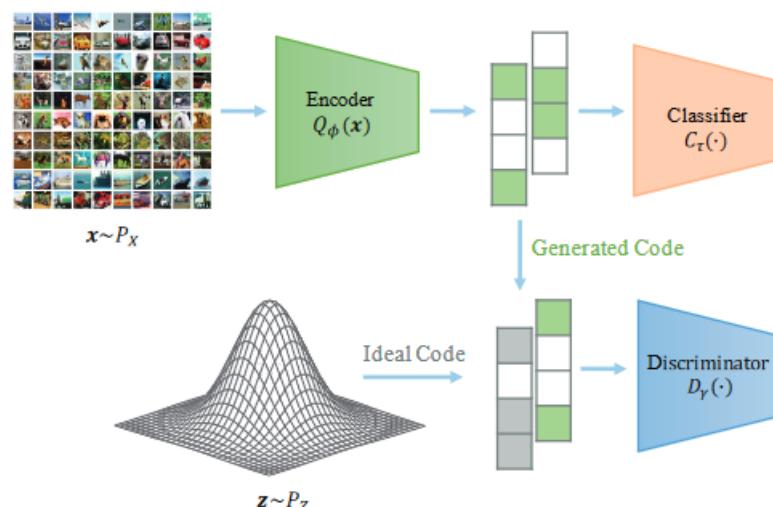


شکل ۱۱-۲: دورنمای APE-GAN در زمان آموزش و تست [۳۱]

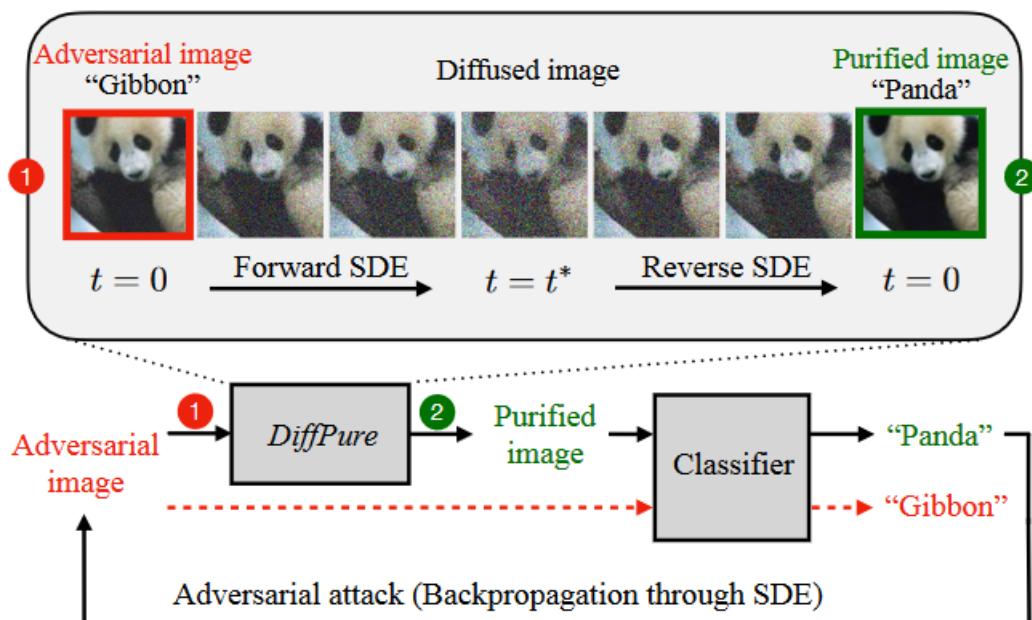
¹Prior

²Discriminator

یکی از جدیدترین ایده های مطرح شده در زمینه پاکسازی نمونه های تخاصمی DiffPure [۳۳] است. در این چارچوب از یک مدل انتشاری^۱ برای پاکسازی نویز تخاصمی تزریق شده به یک تصویر استفاده می شود^۲. نحوه عملکرد این چارچوب در شکل ۱۳-۲ نمایش داده شده است. مدل های انتشاری از دو فرایند انتشار پیش رو^۳ و انتشار معکوس^۴ تشکیل می شوند. ایده ای اصلی DiffPure آن است که با اعمال



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



شکل ۲-۳ : نحوه عملکرد DiffPure [۳۳]

^۱Diffusion Model

^۲ برای اطلاعات بیشتر راجع به مدل های انتشاری می توانید به بخش ۲-۴-۲ مراجعه کنید

^۳Forward Diffusion

^۴Reverse Diffusion

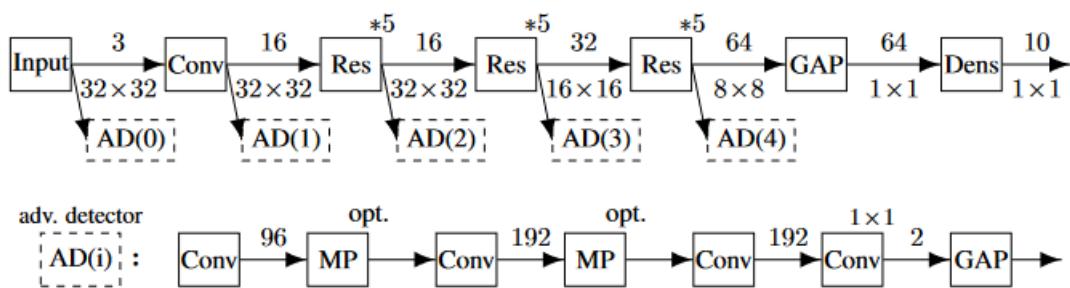
انتشار پیش‌روی ورودی تخاصمی احتمالی و به تبع آن افزودن مقداری نویز ایزوتروپیک گاوی به آن، تاثیر نویز تخاصمی تزریق شده به ورودی از بین خواهد رفت. سپس برای آن که خروجی دسته‌بند از نویزی بیش از حد اعمال شده به ورودی دچار تغییر نشود، به همان تعداد گام پیش‌روی، فرایند انتشار معکوس انجام خواهد شد تا در نهایت 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 تخاصمی، پیاده‌سازی یک دسته‌بند دو-کلاسه برای تمیز دادن نمونه‌های سالم از نمونه‌های تخاصمیست. این ایده در [۳۴] مورد بررسی قرار گرفته است. یکی از نقاط قوت اصلی این روش آن است که هیچ پیش‌فرضی راجع به مدلی که از آن دفاع می‌کند ندارد. ضعف اصلی ذکر شده برای این روش در همین مقاله، این است که چنین دسته‌بندی قدرت تعمیم‌پذیری کمی دارد و خصوصاً نسبت به نوع حمله مورد استفاده شده بسیار حساس است. در این پژوهش که از حملات FGSM و JSMA [۳۵] برای ارزیابی مدل استفاده شده است، نتایج تجربی نشان می‌دهند که تشخیص دهنده‌هایی که روی حملات FGSM شده‌اند، نمی‌توانند در برابر حملات JSMA به خوبی دفاع کنند و بر عکس.

به طور مشابه در [۳۶] از یک زیر-شبکه تشخیص دهنده درون دسته‌بند استفاده می‌شود. بدین ترتیب که پس از یکی از لایه‌های پنهان^۱، شبکه اصلی به دو شاخه تقسیم می‌شود و ویژگی‌های مستخرج از لایه‌ی پنهان قبلی به عنوان ورودی زیر-شبکه تشخیص دهنده و نیز ادامه دسته‌بند، داده خواهند شد. این معماری در شکل ۱۴-۲ نشان داده شده است. برای آموختن این شبکه، ابتدا دسته‌بند اصلی روی نمونه‌های



شکل ۱۴-۲: معماری تشخیص دهنده Metzen [۳۶]

^۱Hidden Layers

سالم آموزش داده خواهد شد. سپس، زیرشبکه‌های تشخیص دهنده در لایه‌های میانی به دسته‌بند اضافه می‌شوند و نمونه‌های تخاصمی به تعداد برابر نمونه‌های آموزشی، تشکیل خواهد شد. در نهایت وزن‌های دسته‌بند اصلی freeze شده و فقط زیرشبکه‌های تشخیص دهنده روی نمونه‌های سالم و تخاصمی به عنوان یک دسته‌بند دو- کلاسه، آموزش داده خواهد شد. در [۳۷] ایده‌ی مشابه ولی با ارتباط بین زیرشبکه‌های دسته‌بند با الهام گرفتن از ایده‌ی Boosting برای دسته‌بندهای آبشاری^۱ [۳۸]، مطرح شده است.

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

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

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

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

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

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

¹Cascade

²Fully Connected

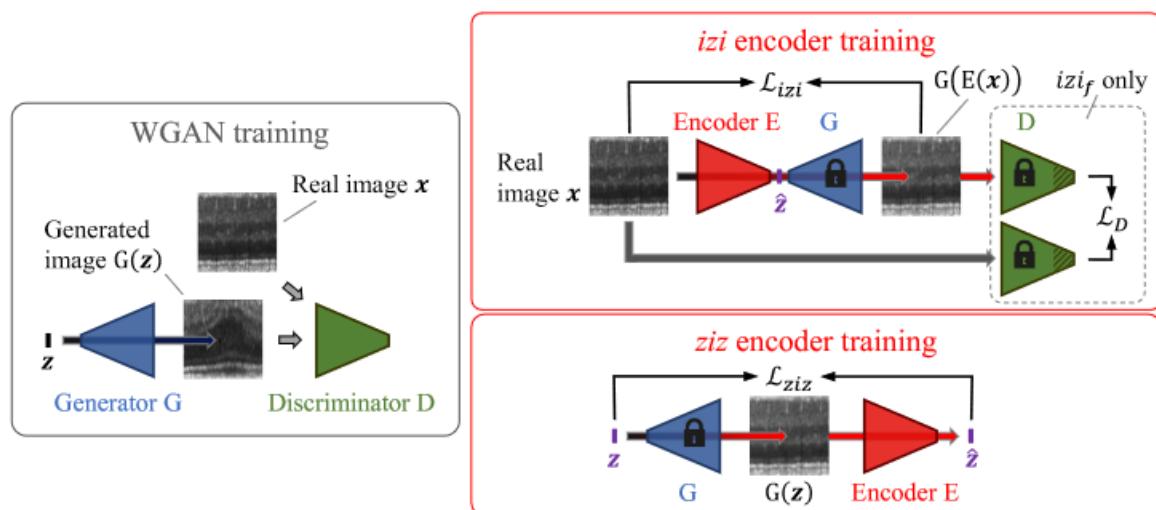
³Gaussian Mixture Model

با یک مقدار مرجع (مثلاً Φ) جایگزین شده است، محاسبه می‌شود. به عبارت دقیق‌تر:

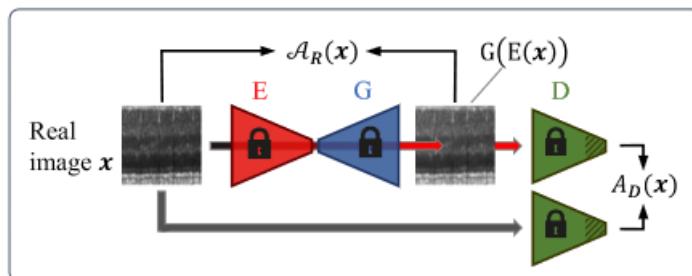
$$\Phi(x)_i := f(x)_c - f(x_{(i)})_c, \quad \text{s.t. } c = \arg \max_{j \in C} f(x)_j.$$

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

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



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



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

^۱Wasserstein Generative Adversarial Network

ترکیب دو سنجه‌ی $A_R(x)$ و $A_D(x)$ استفاده می‌شود (شکل ۱۶-۲):

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

که در آن

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

معیاری از خطای بازسازی^۱ 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} ویژگی‌های یکی از لایه‌های میانی شبکه ممیز است، با استدلالی مشابه $A_R(x)$ و با الهام گرفتن از روش تطبیق ویژگی^۲ [۴۲] به عنوان سنجه‌ی 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 نگاهی به روش ارائه شده در [۴۳] خواهیم انداخت^۳. در این روش - که نزدیک ترین کار انجام شده به کار ماست - بار دیگر از شبکه‌های مولد تخصصی، ولی این بار مشروط بر کلاس^۴ کمک گرفته شده است. در ACGAN-ADA ابتدا یک شبکه ACGAN^۵ را نمونه‌های سالم آموزش داده می‌شود. سپس از تمام بخش‌های این شبکه برای ایجاد سنجه‌هایی برای تشخیص حمله استفاده می‌شود. به طور مشخص، سنجه‌های مورد استفاده موارد زیر هستند:

$$S_R = D(x)$$

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

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

¹ Reconstruction Error

² Feature Matching

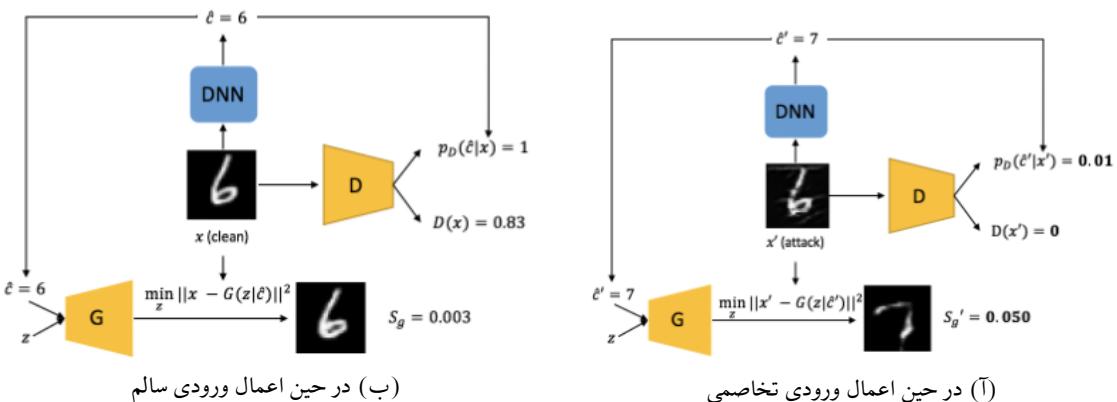
³ برای ارجاع دادن در این روش و با توجه به این که محققین اسم خاصی در مقاله به آن نسبت نداده‌اند، از این جا به بعد این روش را ACGAN-ADA خطا خواهیم کرد

⁴ Class Conditional

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. احتمال پسین^۱ کلاس تشخیص داده شده توسط مدل مورد دفاع، به شرط ورودی داده شده ولی از S_C دسته‌بند اضافی^۲ شبکه‌ی ACGAN است. مشخص است در صورتی که مقدار S_C کم باشد احتمال بروز حمله وجود دارد چرا که بین دسته‌بند اضافی و مدل مورد دفاع روی کلاس نهایی توافق کمی وجود دارد. در نهایت S_g کمینه‌ی نرم افکنش ورودی به شرط کلاس تشخیص داده شده توسط مدل مورد دفاع است. با استدلالی همانند روش ارائه شده در Defense-GAN، کوچک بودن این مقدار به منزله احتمال بیشتری برای سالم بودن ورودیست. نحوه عملکرد این روش در شکل ۱۷-۲ نشان داده شده است. در حین اعمال ورودی سالم (شکل ۱۷-۲ ب) مقدار $p_D(\hat{c}|x)$ و $D(x)$ بالا بوده در حالی که مقدار S_g ناچیز است. در مقابل، در حین اعمال ورودی تخاصمی (شکل ۱۷-۲ آ) دقیقاً بر عکس این حالت اتفاق خواهد افتاد. در ادامه این پژوهش ترکیب‌های مختلفی از این سنجه‌های 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_R و S_C) استفاده می‌کند.

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

برای تشخیص نمونه‌های تخاصمی که دور از خمینه نمونه‌های واقعی قرار دارند روشی تحت عنوان KD-



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

^۱Posterior

^۲Auxiliary Classifier

[۴۵] Detection پیشنهاد شد. در این روش از تخمین چگالی هسته‌ای^۱ (KDE) برای تخمین چگالی نمونه‌های سالم از روی ویژگی‌های استخراج شده توسط آخرین لایه پنهان برای کمک به تشخیص نمونه‌های تخاصمی استفاده می‌شود. به طور کلی، تخمین چگالی به عنوان معیاری از فاصله یک نمونه از یک خمینه هدف مورد استفاده قرار می‌گیرد. فرض کنید

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

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

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

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

ایده‌ی دیگر مطرح شده در این زمینه، ADA [۴۶] است. بنای اصلی این پژوهش، این فرضیه است که نمونه‌های تخاصمی موفق در فضای مرز تصمیم ایجاد شده توسط دسته‌بند، به اندازه کافی به کلاس هدف شبیه هستند ولی از آنجایی که نرم نویز تخاصمی اعمال شده به نمونه باید محدود شده باشد، انتظار است که این نمونه‌های تخاصمی از کلاس نمونه‌ی سالم متناظرشان نیز چندان دور نخواهند بود. در همین راستا، در این پژوهش با استفاده از مدل‌های تخمین چگالی، فضای تولید شده توسط لایه‌های پنهان دسته‌بند مورد دفاع را به صورت ریاضی مدل می‌کنند. سپس از دیورژانس^۲ (KL) Kullback-Leibler (KL) [۴۷] بین معیارهای تخمین زده شده توسط خود دسته‌بند و مدل‌های تخمین چگالی و یک حد آستانه مناسب، برای تشخیص حملات استفاده می‌شود.

¹Kernel Density Estimation

²Divergence

در [۴۸] روش دیگری تحت عنوان Mahalanobis Detector (MD) مطرح شده است. فرض کنید یک دسته‌بند عصبی softmax آموزش دیده شده در اختیار داشته باشیم که احتمال پسین

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

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

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

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

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

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

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

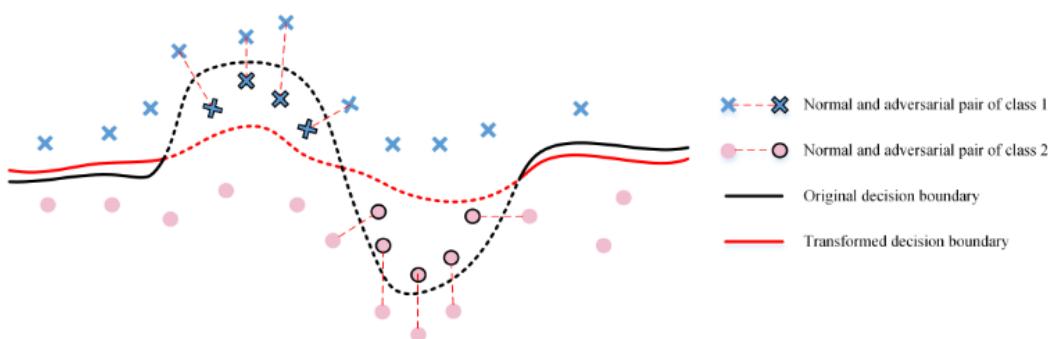
$$f_{y,z}(x) = f_z(x) - f_y(x) \quad (15-2)$$

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

¹Mahalanobis Distance

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 $F(\hat{x}) = \hat{y} \neq F(x) = y$ در اختیار باشد که $\hat{x} = x + \eta$ مقاوم نیست، بنابراین اگر نمونه تخاصمی $\delta = x + \eta$ در اختیار باشد که $f_{\hat{y},y}(\hat{x} + \eta) > f_{\hat{y},y}(\hat{x})$ چرا که نویز کوچک η اضافه شده مقدار کمی از آنگاه انتظار می‌رود که $f_{\hat{y},y}(\hat{x} + \eta)$ جفتی بین کلاس به اشتباه تشخیص داده شده تاثیر نویز تخاصمی δ 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 η های مختلف) می‌توان به عنوانی معیاری برای تشخیص نمونه‌های تخاصمی با استفاده از یک حد آستانه مطلوب بهره برد.

در چارچوبی تحت عنوان SID^۱ [۵۰] این ایده مطرح شده است که نمونه‌های تخاصمی از هل دادن نمونه‌های سالم به سمت دیگر مرز تصمیم در جاهایی که مرز تصمیم یک دسته‌بند دارای تلاطم زیادیست، تشکیل می‌شوند(شکل ۱۸-۲). برای حل این مشکل ایده‌ی مطرح شده آن است که مرز تصمیم دسته‌بند را طوری تغییر داد که دیگر نتوان با تغییرات اندک نمونه‌های سالم را به سوی دیگر مرز تصمیم هل داد و موجب تشکیل نمونه‌های تخاصمی شد بدون آن که مرز تصمیم در نقاط کم تلاطم دچار تغییرات 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 j شبیه یک دیگر باشند که این مطلوب را می‌توان به تولید بردارهای احتمال مشابه در دو دسته‌بند در تمام نقاط غیر از نقاط مذکور، کاهش داد. این مسئله را می‌توان به صورت یک مسئله بهینه‌سازی کمینه-بیشینه بیان کرد که هزینه متحمل شده در بدترین حالت اختلاف مرز تصمیم



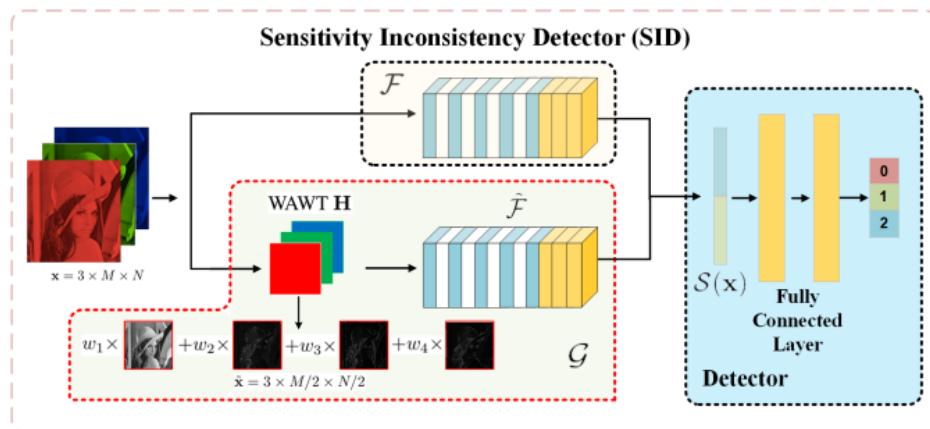
شکل ۱۸-۲: تشکیل نمونه‌های تخاصمی در اطراف خمیدگی‌های مرز تصمیم یک دسته‌بند [۵۰]

^۱Sensitivity Inconsistency Detector

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

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

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



شکل ۱۹-۲: عملکرد SID در زمان تست [۵۰]

¹ Slack Variable

² Weighted Average Wavelet Transform (WAWT)

۴-۲ مختصری در مورد هوش مولد

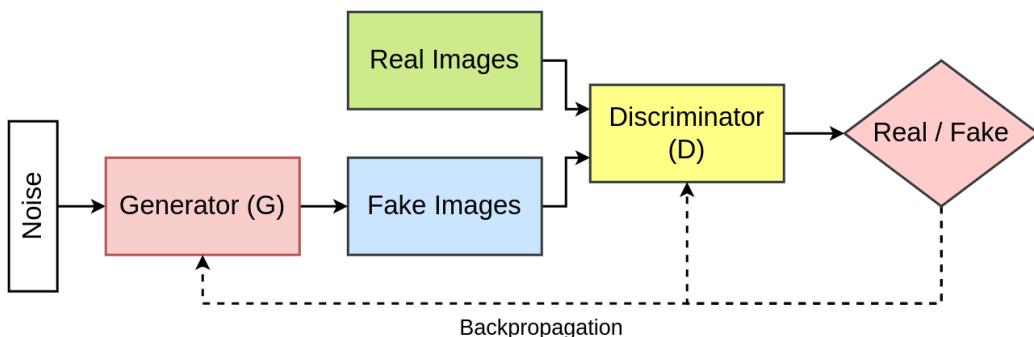
هوش مولد^۱ عبارتیست که به یکی از زیر شاخه های هوش مصنوعی اطلاق می شود که تمرکز اصلی آن تولید رسانه های مختلف از جمله متن، تصویر، فیلم و مدل های سه بعدی وغیره است. در هوش مولد از محتوای موجود کنونی و برای آموزش مدل هایی استفاده می شود که توانایی خلق محتوای جدید را داشته باشند بدون آن که لزوماً نمونه های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

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

۱-۴-۲ شبکه های مولد تخصصی

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

همانطور که در شکل ۲۰-۲ نشان داده شده است، ایده‌ی کلی GAN ها برقراری یک بازی مجموع-صفرا^۳ بین دو شبکه به نام های مولد^۴ و ممیز است. هدف شبکه های مولد در حالت ایده‌آل تولید تصاویری از توزیع تصاویر حقیقی موجود است. برآورده کردن این هدف به کمک بازی ایجاد شده میان مولد و ممیز صورت می‌گیرد.



شکل ۲۰-۲: دورنمای شبکه های مولد تخصصی

¹Generative AI

²Unsupervised Learning

³Zero-sum

⁴Generator

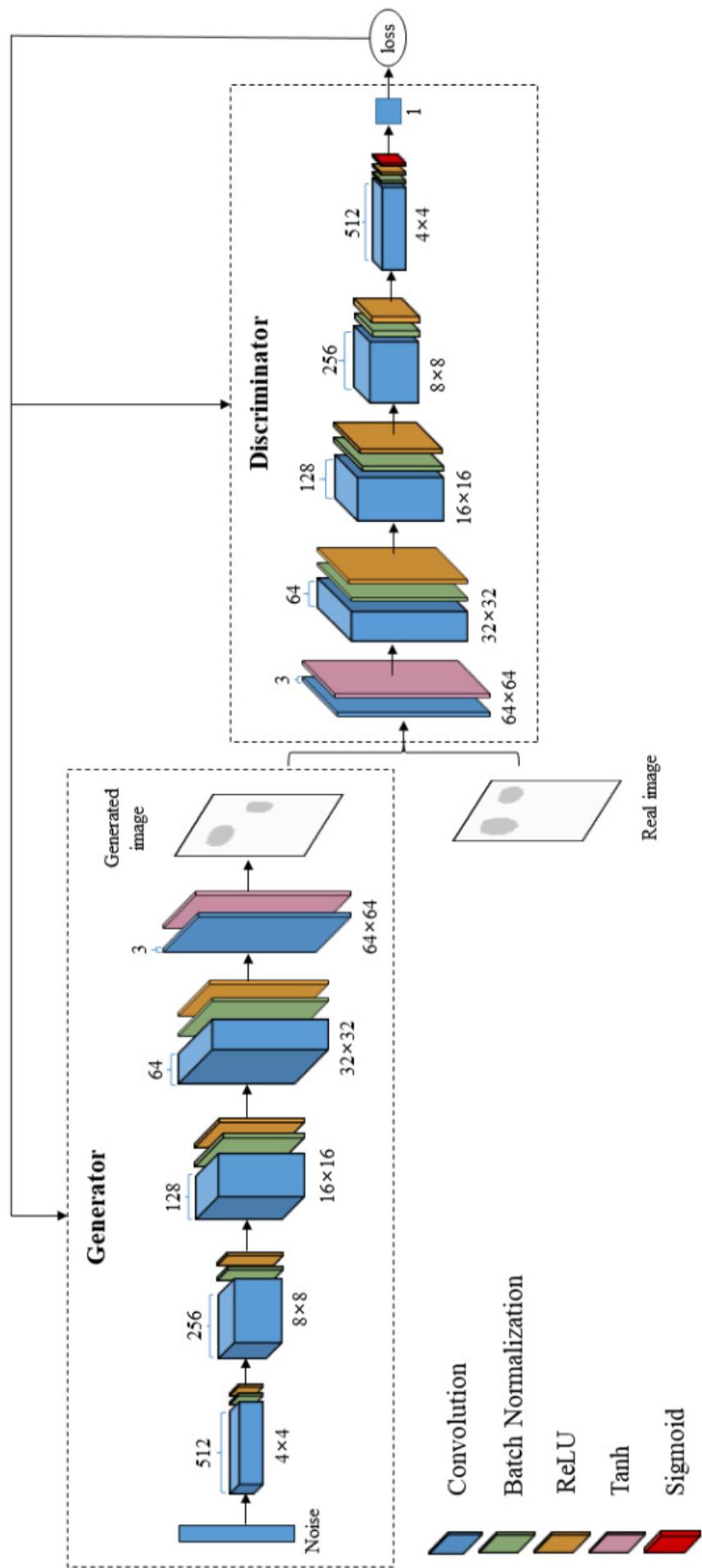
مولد تلاش می‌کند با تولید نمونه‌هایی که به نمونه‌های واقعی شبیه هستند، ممیز را فریب دهد. از طرف دیگر، ممیز در تلاش است که با تقویت خودش در برابر فریب خوردن از مولد مصون بماند و همچنان بتواند نمونه‌های ساختگی و واقعی را از هم تفکیک کند. به طور دقیق‌تر، مولد G و ممیز D بازی کمینه–بیشینه زیر را با تابع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

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

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

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

^۱Transposed Convolution

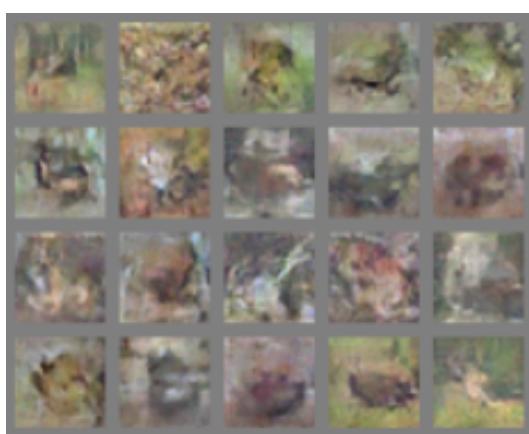


شکل ۲-۱: معماری شبکه DCGAN

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

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

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



(ب) نمونه‌های تولید شده روی مجموعه داده CIFAR



(آ) نمونه‌های تولید شده روی مجموعه داده MNIST

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

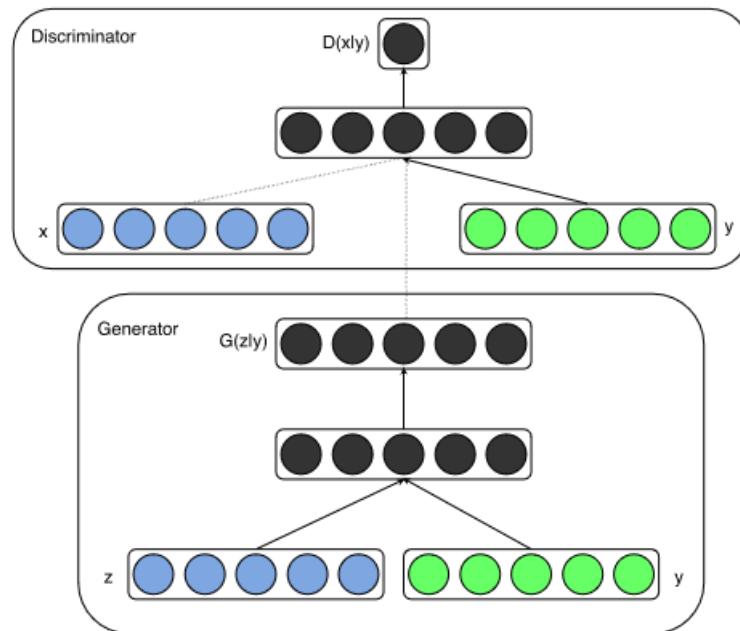
¹Mode Collapse²Nash Equilibrium

مولد به مراتب ساده‌تر است، ممیز به سرعت از مولد در بازی پیشی می‌گیرد و می‌تواند به راحتی نمونه‌های ساختگی و واقعی را از هم تفکیک کند. بنابراین از آنجایی که خروجی ممیز به ازای هر ورودی دلخواه عددی بسیار نزدیک به صفر (برای ورودی‌های ساختگی) و یا بسیار نزدیک به ۱ (برای ورودی‌های واقعی) است، مولد دچار پدیده‌ی گرادیان محو شونده^۱ 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 [۵۴، ۵۵]، مشروط کردن خروجی مولد به برچسب مورد انتظار است. این ایده منجر به معرفی ساختارهای جدیدی به نامهای cGAN و کمی بعدتر ACGAN شد که در ادامه کمی بیشتر این دو ساختار را مورد بررسی قرار خواهیم داد.

cGAN ۱-۱-۴-۲

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



شکل ۲-۲: [۵۶] معماری cGAN

^۱Vanishing Gradient

تزریق یک شرط به ورودی‌های مولد و ممیز در این پژوهش به صورت بسیار ابتدایی و بدیهی مطرح شده است و به عنوان عامل مشروط کننده از برچسب کلاس نمونه‌ها استفاده می‌شود. مطابق شکل 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^۱ می‌شود و لایه‌های بعدی مانند یک GAN معمولی عمل خواهد کرد. در این وضعیت مولد G و ممیز D توزیع‌های شرطی $G(z|y)$ و $D(x|y)$ را تخمین می‌زنند و بازی ۱۷-۲ به شکل ساختار یافته تر زیر در خواهد آمد:

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

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



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

^cConcatenate

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

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

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

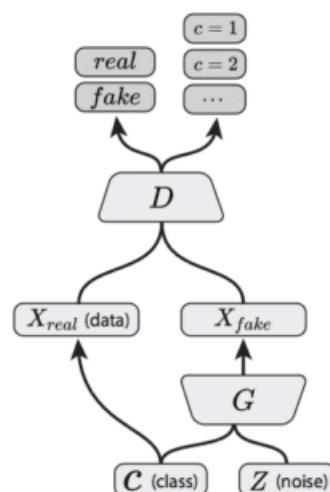
به تبع این موضوع،تابع هزینه‌ی آموزش یک ACGAN از دو بخش تشکیل می‌شود:

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

و

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

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



شکل ۲۵-۲: معماری ACGAN

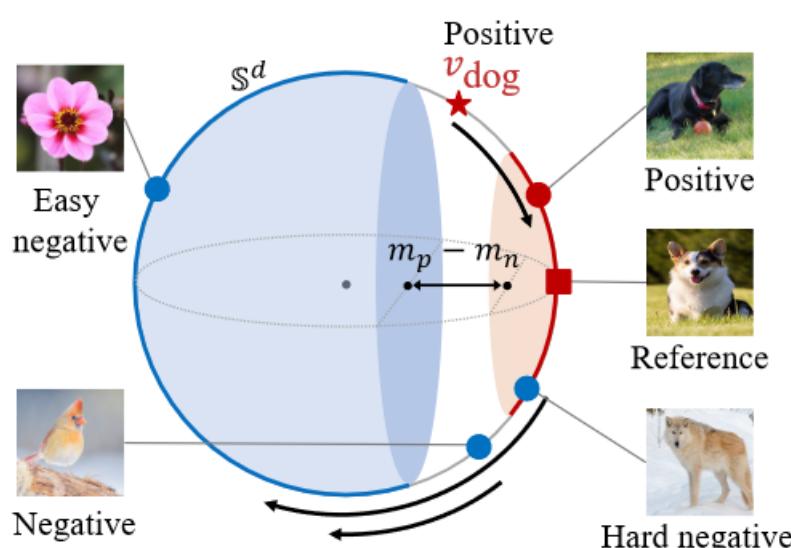
بدین ترتیب ممیز در راستای کمینه کردن $L_C + L_S$ و مولد در راستای کمینه کردن $L_C - L_S$ آموزش خواهد دید. با وجود بهبودهای این روش نسبت به GAN، این معماری همچنان خالی از ایراد نبود. یکی از اصلی‌ترین ایراد‌های وارد به این معماری، عدم توانایی مولد در یادگیری هنگامیست که تعداد کلاس‌های مجموعه داده‌ها زیاد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

ReACGAN ۳-۱-۴-۲

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

۱. عدم پایداری آموزش زمانی که تعداد کلاس‌های مجموعه داده‌ها زیاد باشد.
۲. نمونه‌های تولید شده توسط ACGAN دارای تنوع نسبتاً کم و به راحتی قابل تشخیص و دسته‌بندی هستند.

علت بروز مشکل اول، انفجار گرادیان در دسته‌بند جانبی ACGAN در گام‌های اولیه‌ی فرایند آموزش مطرح شده است. در این پژوهش به صورت تجربی نشان داده می‌شود که افکندن ورودی‌های دسته‌بند جانبی ACGAN به یک ابرکره‌ی شعاع واحد می‌تواند از این مشکل جلوگیری کند. در ادامه برای برطرف کردن ایراد دوم و مجبور کردن مولد به تولید نمونه‌های متنوع، جمله‌ی جدیدی تحت عنوان Data-to-Data Cross-Entropy (D2D-CE) به عنوان تابع هزینه‌ی آموزش جدید مدل معرفی می‌شود. هدف این تابع هزینه، استفاده از ویژگی‌های مستر در خود نمونه‌های آموزش با در نظر گرفت تفاوت‌ها و شباهت‌های آن‌ها نسبت به فاصله‌ی آن‌ها در فضای embedding است. همانطور که در شکل ۲۶-۲ مشاهده می‌شود فضای embedding یک ReACGAN برخلاف روش‌های



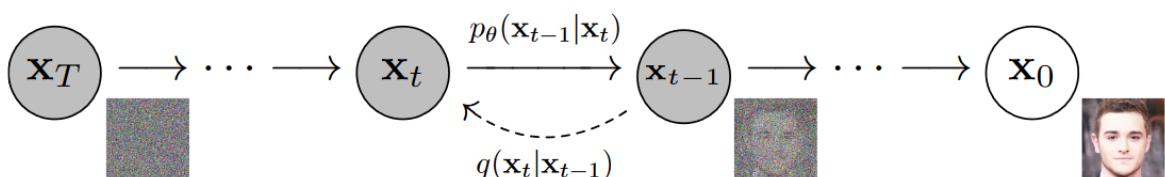
شکل ۲۶-۲: فضای embedding تصاویر در یک ReACGAN

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

۲-۴-۲ مدل‌های انتشاری

دسته‌ی دیگری از مدل‌های مولد که اخیراً معرفی شده‌اند و به سرعت در حال گسترش هستند، مدل‌های انتشاری هستند. ایده‌ی مدل‌های انتشاری از مبحثی در فیزیک تحت عنوان ترمودینامیک غیرتعادلی^۳ الگو برداری شده است. با وجود این که اولین پژوهش در رابطه با مدل‌های انتشاری در سال ۲۰۱۵ و تقریباً به صورت همزمان با GAN منتشر شد [۵۸]، آوازه این مدل‌ها اولین بار در ۲۰۲۰ پس از انتشار مقاله معروف [۵۹] تحت عنوان مدل‌های انتشار احتمالی حذف کننده نویز^۴ بر سر زبان‌ها آمد. مدل‌های انتشاری یک زنجیره مارکوف از گام‌های انتشار را تعریف می‌کنند که به تدریج مقداری نویز تصادفی به یک نمونه می‌افزایند و سپس یاد می‌گیرند که عکس این فرایند انتشار را انجام دهند تا بتوانند با شروع از نویز به یک نمونه‌ی ساختگی که به توزیع نمونه‌های واقعی نزدیک است، برسند (شکل ۲۷-۲).

همان‌طور که پیش‌تر توضیح داده شد، مدل‌های انتشاری از دو فرایند انتشار پیش‌رو و انتشار معکوس تشکیل



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

¹Reference

²Non-equilibrium Thermodynamics

³Denoising Diffusion Probabilistic Model (DDPM)

⁴Markov Chain

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

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

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

بدین ترتیب، همانطور که در شکل ۲۷-۲ نیز نشان داده شده است به مرور زمان و با بزرگ شدن t نمونه‌ی x_0 تخریب می‌شوند و اگر $\infty \rightarrow T$, می‌توان نشان داد که x_T نمونه‌ای از توزیع یک گاووسی 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 گاووسی ایزوتروپیک و اعمال T گام زمانی معکوس، به نمونه‌ای از توزیع واقعی (x_t) رسید. می‌توان نشان داد که اگر β_t به اندازه کافی کوچک باشد، انتشار معکوس $(x_{t-1}|x_t)$ نیز تقریباً یک گاووسی است. ولی از آنجایی که محاسبه‌ی $(x_{t-1}|x_t)$ نیازمند استفاده از کل مجموعه‌ی داده است (به دلیل وجود جمله‌ی امید ریاضی)، مجبوریم از یک مدل ریاضی p_θ برای تخمین q استفاده کنیم. می‌توان نشان داد که 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 [۵۹] نشان داده می‌شود که می‌توان به جای تخمین زدن پارامترهای گاووسی تعریف شده در معادله ۱۹-۲ هدف آموزش مدل p_θ را طوری تغییر داد که فقط نیاز به تخمین نویز اضافه شده ϵ_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 ϵ_t با ابعاد ورودی x_t برابر به یک معماری U-Net [۶۲] مانند نیاز خواهد بود که وظیفه‌ی آن تخمین زدن نویز اضافه شده به نمونه‌ی ورودی در هر گام زمانی $T \leq t \leq 1$ است. از آنجایی که هدف آموزشی مدل‌های انتشاری مطابق تخمین بیشینه درست‌نمایی^۱ انجام می‌شود که به مراتب

^۱Maximum Likelihood Estimation

ساختار یافته تراز بازی کمینه-بیشینه بین مولد و ممیز در یک GAN است، آموزش مدل‌های انتشاری پایدار تر بوده و مشکلاتی مانند فروپاشی مُد معمولاً^۱ در این مدل‌ها مشاهده نمی‌شود [۶۱]. از طرف دیگر، همانطور که پیش‌تر توضیح داده شد مقدار β_t در هر گام زمانی باید به اندازه کافی کوچک باشد تا بتوان انتشار معکوس را توسط یک گاوسی تخمین زد و بنابراین به مقدار T (تعداد کل گام‌های زمانی فرایند انتشار) باید بزرگ باشد (مثلاً ۱۰۰۰ گام) و در نتیجه برای تولید هر نمونه به تعداد قابل توجهی forward pass نیاز است که می‌تواند هزینه بر باشد. با این وجود، نمونه گیری از DDPM‌ها در تعداد گام‌های زمانی کمتر همچنان یک موضوع تحقیق فعال است و پژوهش‌هایی (مانند DDIM [۶۳]) در این زمینه صورت گرفته است که می‌تواند تعداد گام‌های مورد نیاز برای نمونه برداری را به تعداد بسیار کمی کاهش دهد [۶۰].

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

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

مانند GAN‌ها تولید مشروط محظوظ بر مبنای شروطی مانند برچسب کلاس و یا متن توصیف کننده محظوظ 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^۱: ایده‌ی ارائه شده در [۶۴] این بود که ابتدا یک دسته‌بند $f_\phi(y|x_t)$ روی تصاویر نویزی x_t آموزش داده شده و سپس از لگاریتم گرادیان خروجی این دسته‌بند نسبت به تصویر



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

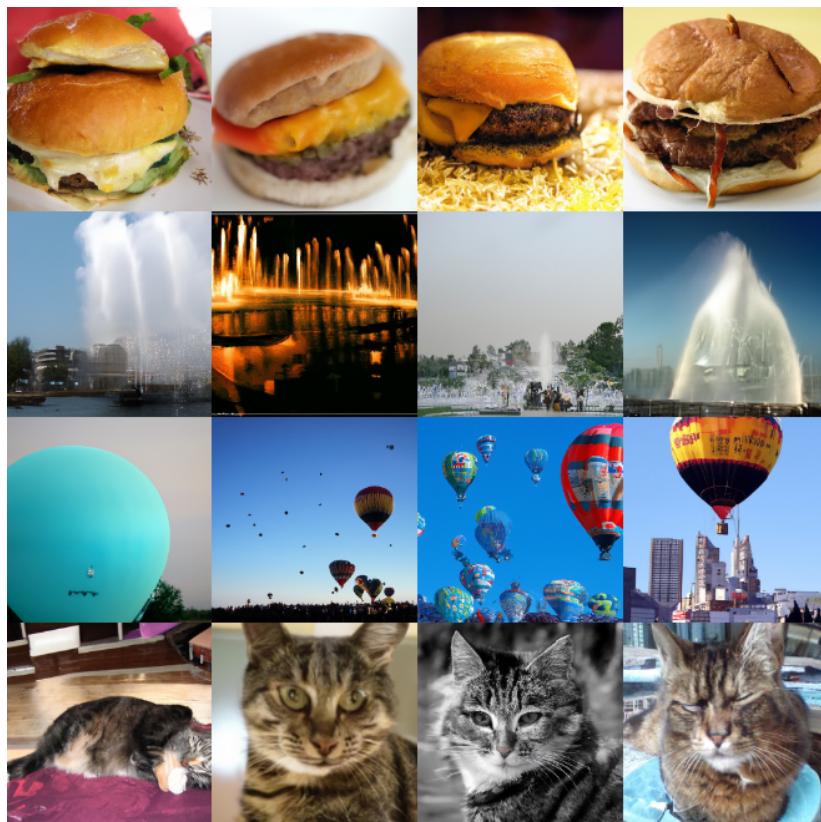
^۱Classifier Guided Diffusion Model

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

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

که guidance از گرادیان ذکر شده مشتق می‌شود و α 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^۱: هدایت یک مدل انتشاری بدون استفاده از یک دسته‌بند جانبی نیز ممکن است. اگر بتوان به نحوی با استفاده از خود مدل انتشاری و بدون هیچ شبکه اضافی، هر دو توزیع $p_\theta(x|y)$ و $p_\theta(x)$ را یادگرفت، آنگاه می‌توان از ترکیب نمونه‌های این دو توزیع، نمونه‌های جدیدی



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

^۱Diffusion Model with Classifier-Free Guidance

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



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

فصل سوم

پیشنهاد روشی نوین برای مقابله با حملات تخاصمی

۱-۳ مقدمه

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

۲-۳ بیان مسئله

دسته‌بند تصویر

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

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

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

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

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

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

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

۳-۳ روش پیشنهادی

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

۳-۳-۱ استفاده از مولدهای قوی تر

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

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 می‌شود:

$$S_R = D(x)$$

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

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

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

۱. S_R : بزرگ بودن این مقدار می‌تواند به منزله‌ی بزرگ بودن میزان نویز تخاصمی تحریق 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.

در ACGAN-ADA برای تعیین حد آستانه برای هر کدام از معیارهای ارائه شده از روش جستجوی شبکه‌ای^۱ روی آستانه‌ها استفاده شده است. در پژوهش ما، علاوه بر این سه معیار دیگر نیز برای تشخیص حمله بهره برد 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 (مثالاً احتمال 0.998 در یک دسته‌بندی 10 کلاسه) و یا قطعیت نسبتاً پایین (مثالاً 0.3 در همان دسته‌بندی 10 کلاسه) می‌تواند نمایانگر یک حمله‌ی تخاصمی باشد.

¹Grid Search

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

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

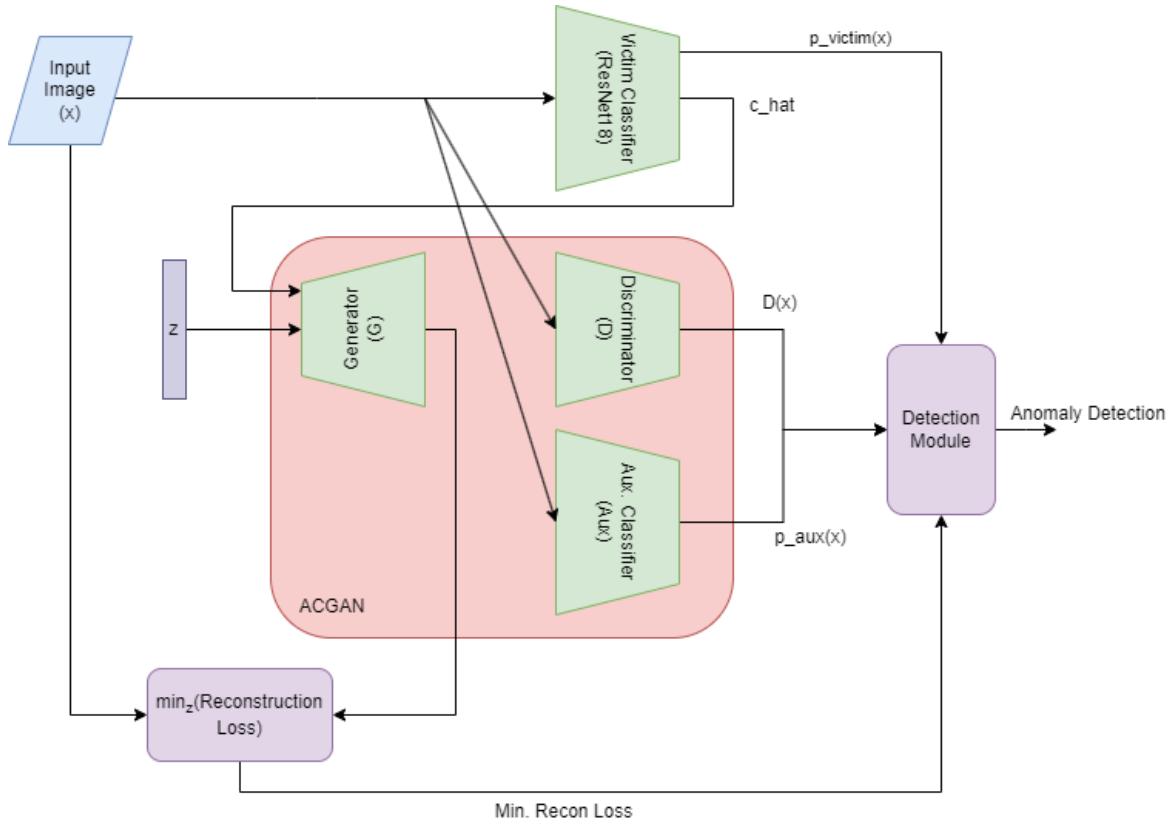
که در آن D_{KL} همان دیورژانش KL بوده و $M = \frac{1}{2}(P + Q)$.

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

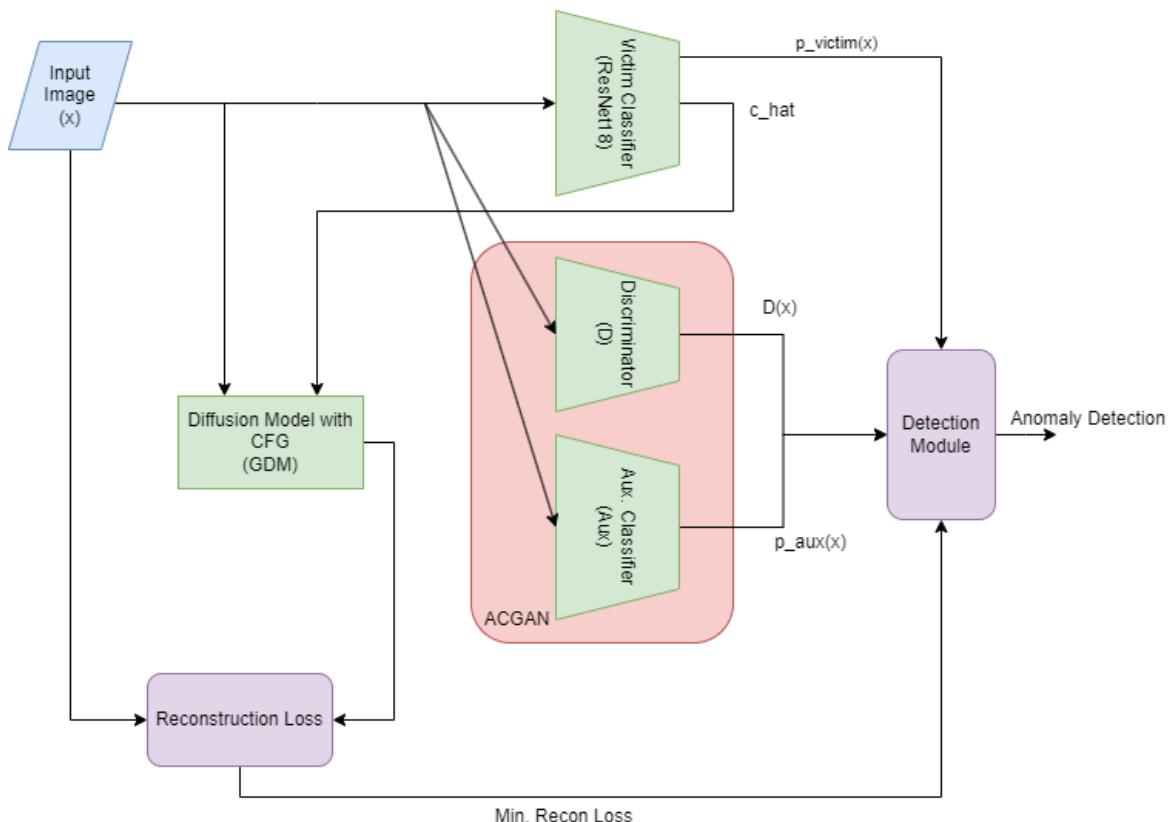
از طرف دیگر، در ARCANE برای تشخیص حمله از جستجوی شبکه‌ای برای پیدا کردن حدود آستانه استفاده نمی‌کنیم. در عوض تمامی این معیارها به صورت داده‌های جدولی^۲ به یک دسته‌بند XGBoost [۶۸] داده می‌شوند و تصمیم‌گیری نهایی توسط این دسته‌بند ثانویه صورت می‌گیرد. دورنمای کلی تشخیص حمله توسط ARCANE-Diff و ARCANE-GAN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

¹Joint

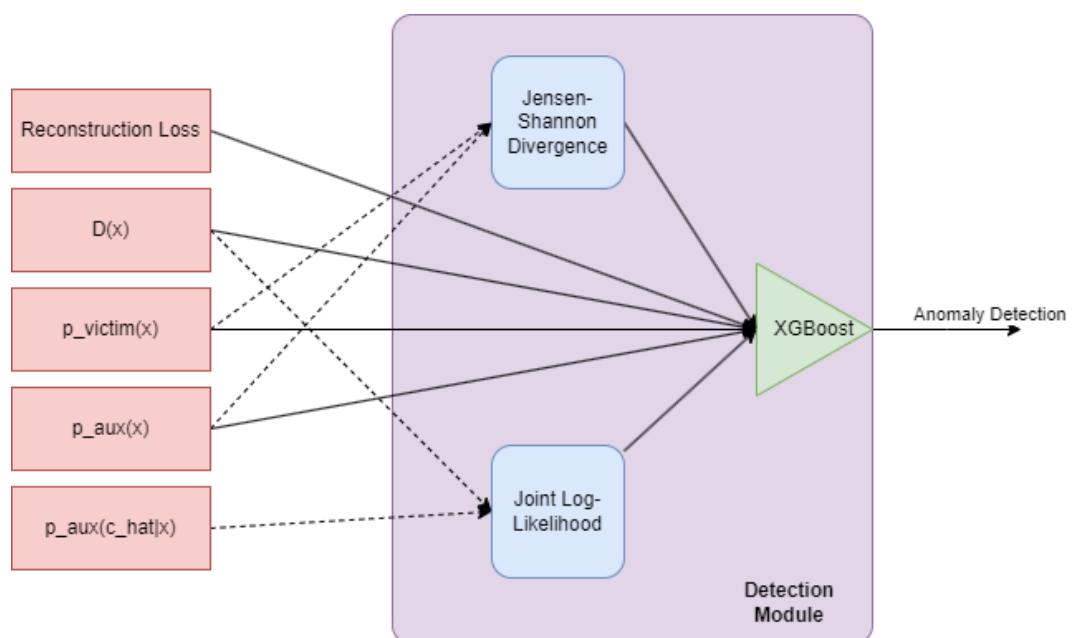
²Tabular



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



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



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

۳-۳-۳ روش بهبود یافته برای پاکسازی حمله

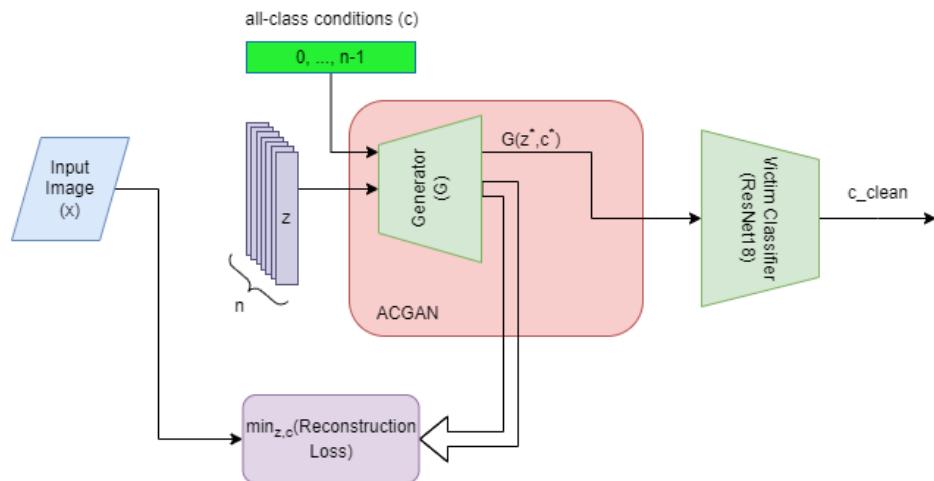
فرایند پاکسازی حملات شبیه چارچوب ارائه شده در ACGAN-ADA است با این تفاوت که مولدهای تقویت شده شرطی برای تولید نمونه‌های هدایت شده بهره خواهیم برد. علاوه بر آن پاکسازی نسبت به تمامی برچسب‌ها به صورت همزمان صورت می‌گیرد و در نهایت بین تمام نمونه‌های تولید شده، هر کدام که مقدار نرم اختلافش با نمونه‌ی تخاصمی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

برای تولید تصاویر با استفاده از مولد ReACGAN از ایده‌ای شبیه Defense-GAN [۳۰] ولی به صورت شرطی بهره بردۀ 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.

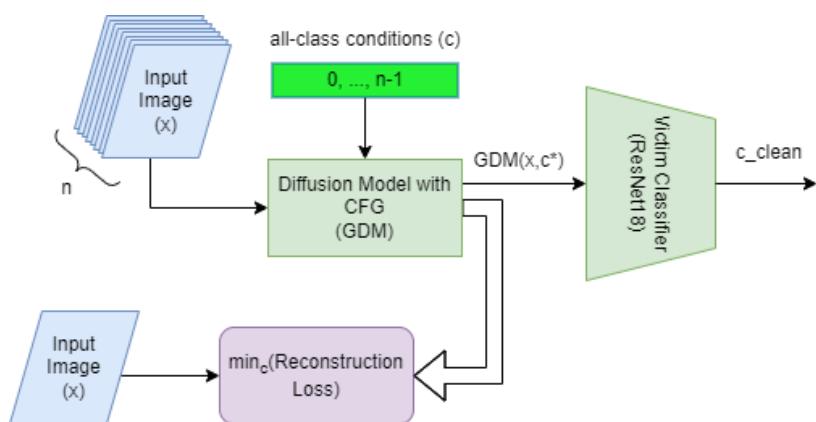
برای تولید تصاویر توسط مدل انتشاری از روش ارائه شده در DiffPure [۳۳] اما به صورت مشروط، و روی تمام کلاس‌ها استفاده می‌کنیم. در واقع هر تصویر ورودی به اندازه $T \leq t$ کام زمانی مورد تاثیر انتشار پیش‌رو قرار می‌گیرد و نویزی می‌شود. سپس به همان تعداد گام زمانی انتشار معکوس و مشروط بر تمام کلاس‌ها روی n (تعداد کلاس‌ها) کپی مختلف از تصویر، انجام خواهد شد تا به تصویری شبیه تصویر ورودی برسیم ولی احتمالاً بدون نویز 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^۱ نیز عبور خواهند کرد تا مطمئن شویم ترکیب رنگ خروجی نهایی تا حد ممکن به تصویر ورودی نزدیک است. این امر باعث بهبود دقیق پاکسازی می‌شود. چارچوب کلی پاکسازی ARCANE-GAN و ARCANE-DIFF را در شکل‌های ۴-۳ و ۵-۳ مشاهده می‌کنید.

^۱Histogram Matching



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



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

۱ - ۴ مقدمه

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

۲ - ۴ روش شبیه سازی

تمامی مدل‌ها و آزمایش‌های صورت گرفته روی بستر کتابخانه یادگیری عمیق معروف PyTorch [۶۹]^۱ و با استفاده از زبان برنامه‌نویسی پایتون^۲ پیاده سازی شده‌اند. برای آموزش مدل ReACGAN از مخزن-PyTorch [۷۰]^۳ با اندکی تغییرات استفاده شده است و نهایتاً مدل انتشاری هدایت شده مورد استفاده به کمک مخزن denoising-diffusion-pytorch^۴ پیاده سازی شده است. برای آموزش و آزمون مدل‌های مولد از مجموعه داده‌های CIFAR10 [۷۱] و Tiny-ImageNet [۷۲] استفاده شده است.

¹<https://pytorch.org/>

²Python

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

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

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

برای آزمون مدل‌ها تعداد ۱۰۰۰ نمونه از هر مجموعه داده به صورت تصادفی اما متعادل در بین تمام کلاس‌های ممکن انتخاب شدند و سپس از روی این ۱۰۰۰ نمونه به ازای هر روش حمله یک نمونه‌ی تخاصمی ایجاد شده است. به عنوان مثال، برای مجموعه داده‌ی CIFAR10 ۱۰۰۰ نمونه‌ی سالم، از هر کلاس ۱۰۰ نمونه، انتخاب شده است و سپس ۱۰۰۰ نمونه حمله‌ی FGSM و ۱۰۰۰ نمونه حمله‌ی CW متناظر با این ۱۰۰ نمونه‌ی سالم ایجاد شده و برای آزمون مدل‌ها کنار گذاشته می‌شوند. در زمان آزمون مدل‌ها برای وظایف پاکسازی و تشخیص حملات، به ازای هر نوع حمله، تمام نمونه‌های آن حمله به همراه نمونه‌های سالم به چارچوب مورد آزمایش داده خواهند شد و نتایج روی مجموع ۲۰۰۰ نمونه (۱۰۰۰ نمونه سالم و ۱۰۰۰ نمونه تخاصمی) برای هر حمله گزارش شده است. همچنین، برای آموزش تشخیص دهنده‌ی نهایی XGBoost از ۱۰۰ نمونه سالم و تعداد مساوی نمونه‌ی تخاصمی از هر نوع حمله استفاده شده است. برای حصول اطمینان از عادلانه بودن نتایج، مطمئن شده ایم که مجموعه داده‌ی مورد استفاده برای آموزش XGBoost اشتراکی با مجموعه داده آزمون مدل‌ها که پیش‌تر توضیح داده شد، ندارد. برای دستیابی به بهترین دقت ممکن روی تشخیص دهنده‌ی XGBoost از روش اعتبار‌سنجی ضربدری^۵ – لایه با جستجوی شبکه‌ای روی فرآپارامترهای XGBoost صورت گرفته است و سپس مدل‌های دارای بهترین AUC انتخاب شده‌اند. همچنین، در تمامی حملات، قربانی مورد حمله یک مدل دسته‌بند ۱۸ ResNet بوده که روی مجموعه داده متناظر آموزش دیده است (دقیقاً مطابق [۴۳]). 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^۶ های استفاده شده برای هر مدل، می‌توانید به مخزن ARCANE^۷ مراجعه کنید.

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

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

^۳<https://xgboost.readthedocs.io/>

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

^۵Cross-validation

^۶Hyperparameter

^۷<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.

۱-۳-۱ سنجش عملکرد مدل‌های مولد

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

IS: یکی از اولین معیار قدرتمند برای سنجش عملکرد مدل‌های مولد. برای محاسبهٔ این معیار از خروجی یک دسته‌بند معروف از پیش آموزش داده شده به نام InceptionV3^۱ روی تعداد نسبتاً زیادی از تصاویر تولید شده توسط مدل مولد، استفاده می‌شود. مقدار IS در دو صورت بیشینه خواهد شد (حالت مطلوب):

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

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

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

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

²Fréchet

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

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

حقیقی و نمونه های ساختگیست و بنابراین مطلوب ماست. اگر بتوان توزیع آماری ویژگی های استخراج شده توسط دسته بند را برای نمونه های حقیقی و ساختگی توسط دو گاووسی چند متغیره مدل کرد، برای محاسبه فاصله های فرشه فرم بسته زیر موجود است:

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

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

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

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

۲-۳-۴ سنجش عملکرد تشخیص حملات

یکی از مهم ترین سنجه هایی که برای تسلک های تشخیص مورد استفاده قرار می گیرد، سطح زیر نمودار^۱ Receiver Operating Characteristic (ROC) است. برای تطبیق نتایج به دست آمده با ACGAN-ADA از سطح زیر نمودار ACGAN-ADA (pAUC-۰.۲) استفاده شده است. علت ذکر شده برای این انتخاب در مقاله ACGAN-ADA اهمیت بیشتر تشخیص های صورت گرفته در نواحی ای از نمودار است که نرخ مثبت کاذب ^۲ کم می باشد، چرا که نرخ مثبت کاذب کم در عمل برای یک تشخیص دهنده بسیار مطلوب است و نواحی دیگر نمودار ROC

^۱ Area Under Curve

^۲ False Positive

^۳ FPR (False Positive Rate)

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

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

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

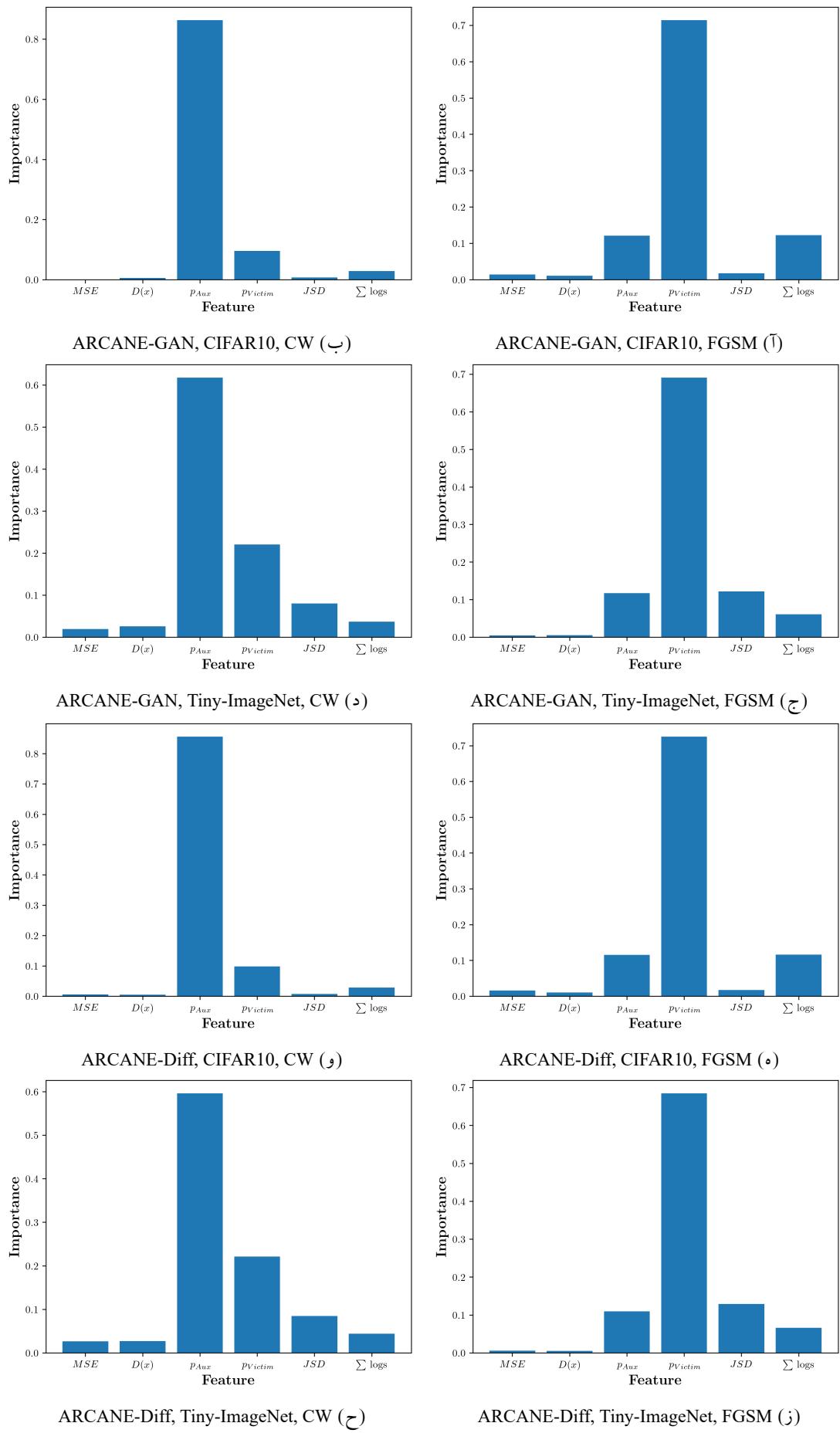
$$pAUC - U = \int_{x=0}^U ROC(x)dx$$

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

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

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

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



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

۳-۳-۴ سنجش عملکرد پاکسازی حملات

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

آمیز تلقی می‌شود:

۱. نمونه سالم بوده و برچسب پیش‌بینی شده توسط مدل قربانی قبل و پس از پاکسازی برابر است.

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

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

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

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

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

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



(ب) پاکسازی توسط ARCANE-GAN



(آ) پاکسازی توسط ARCANE-Diff

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

۴-۴ بررسی و مقایسه نتایج

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

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

CW	FGSM	
0.0576	0.0566	[۴۱] f-AnoGAN
0.0533	0.1642	[۴۵] KD
0.1042	0.1783	[۴۸] MD
0.0910	0.0436	[۴۹] ODDS
0.1489	0.1388	[۵۰] SID
0.1593	0.1782	[۴۶] ADA
0.1881	0.1819	^۱ [۴۳] ACGAN-ADA
0.1999	0.1866	ARCANE-GAN
0.1999	<u>0.1856</u>	ARCANE-Diff

^۱برای ACGAN-ADA از بین چندین روش ارائه شده، نتیجه‌ی بهترین روش در اینجا گزارش شده است.

فصل پنجم

نتیجه‌گیری و پیشنهادها

- ۱-۵ نتیجه‌گیری
- ۲-۵ پیشنهادها

واژه‌نامه انگلیسی به فارسی

A

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

B

Backpropagation	پس انتشار
Binary Search	جستجوی دودویی
Black Box	جهه سیاه
Box Constraints	محدودیت های جعبه‌ای

C

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

D

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

E

Evasion Attack	حمله گریزانه
Exploratory Attack	حمله اکتشافی

F

False Positive	مثبت کاذب
Feature Matching	تطبیق ویژگی
Feature Space	فضای ویژگی
Forward Diffusion	انتشار پیش رو
Fréchet	فریشه
Fully Connected	کاملاً متصل

G

Generative Adversarial Network (GAN)	شبکه مولد تخاصمی
Generative AI	هوش مولد
Generative Models	مدل های مولد
Generator	مولد
Gradient Descent	نزول گرادیان
Grid Search	جستجوی شبکه ای

H

Hidden Layers	لایه های پنهان
Histogram Matching	تطبیق هیستوگرام
Hyperparameter	فراپارامتر

J

Joint	توأم
-------------	------

K

Kernel Density Estimation تخمین چگالی هسته‌ای

L

Latent نهفته

Learning Rate نرخ یادگیری

Loss ضرر

M

Mahalanobis Distance فاصله مahaلانوبیس

Manifold خمینه

Markov Chain زنجیره مارکوف

Maximum Likelihood Estimation تخمین بیشینه درست‌نمایی

Metric سنجه

Min-max کمینه-بیشینه

Mode Collapse فروپاشی مُد

N

Nash Equilibrium تعادل نَش

Noise نویز

Non-equilibrium Thermodynamics ترمودینامیک غیرتعادلی

Norm نرم

P

Pipeline خط لوله

Pixel پیکسل

Poisoning Attack حمله مسموم‌کننده

Posterior پسین

Prior پیشین

Projected Gradient Descent	نزول گرادیان افکنده
Purification	پاکسازی
Python	پایتون

R

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

S

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

T

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

U

Unsupervised Learning	یادگیری بدون نظارت
Untargeted Misclassification	دسته‌بندی اشتباه غیر هدفمند

V

Vanishing Gradient	گرادیان محو شونده
Victim	قربانی

W

White Box	جعبه سفید
-----------------	-----------------

Z

Zero-sum	مجموع-صفر
----------------	-----------------

واژه‌نامه فارسی به انگلیسی

۱

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

ب

Reformer	بهساز
----------------	-------------

	پ
Purification	پاکسازی
Adversarial Purification	پاکسازی تخاصمی
Python	پایتون
Backpropagation	پس انتشار
Posterior	پسین
Prior	پیشین
Pixel	پیکسل
	ت
Auto Trader	تاجر خودکار
Sentiment Analysis	تحلیل احساسات
Maximum Likelihood Estimation	تخمین بیشینه درست‌نمایی
Kernel Density Estimation	تخمین چگالی هسته‌ای
Non-equilibrium Thermodynamics	ترمودینامیک غیرتعادلی
Detection	تشخیص
Feature Matching	تطبیق ویژگی
Histogram Matching	تطبیق هیستوگرام
Nash Equilibrium	تعادل نش
Defensive Distillation	تقطیر دفاعی
Confidence Reduction	تقلیل اطمینان
Tolerance	تلورانس
Tensor	تنسور
Joint	توأم
	ج
Tabular	جدولی
Binary Search	جستجوی دودویی

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

ح

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

خ

Pipeline	خط لوله
Reconstruction Error	خطای بازسازی
Manifold	خمینه
Autoencoder	خودرمزنگذار

د

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

ر

Attack Surface	رویهی حمله
----------------------	------------------

ز

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

س

Area Under Curve	سطح زیر نمودار
Metric	سنجه

ش

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

ض

Loss	ضرر
------------	-----

ف

Mahalanobis Distance	فاصله ماهالانوبیس
Hyperparameter	فرآپارامتر
Fréchet	فریشه
Mode Collapse	فروپاشی مُد
Feature Space	فضای ویژگی

ق

Victim	قربانی
--------------	--------

ک

Fully Connected	کاملاً متصل
Transposed Convolution	کانولوشن ترانهاده
Min-max	کمینه-بیشینه

گ

Vanishing Gradient	گرادیان محو شونده
--------------------------	-------------------

ل

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

م

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

ن

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

هوش مولد Generative AI

۵

یادگیری بدون نظارت Unsupervised Learning

یادگیری عمیق Deep Learning

مراجع

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

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

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

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

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

Arian Tashakkor

tashakkor.a@ec.iut.ac.ir

Feb 19, 2024

Department of Electrical and Computer Engineering

Isfahan University of Technology, Isfahan 84156-83111, Iran

Degree: M.Sc.

Language: Farsi

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

Abstract

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

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



Isfahan University of Technology

Department of Electrical and Computer Engineering

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

A Thesis

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

by
Arian Tashakkor

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

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

Dr. Behzad Nazari, Department Graduate Coordinator

