مقاله پژوهشی

ارائه روشی مقاوم در برابر حملات تخاصمی با استفاده از فرایندهای گوسی مقیاس‌پذیر و رأی‌گیری

مهران صفایانی، پویان شالبافان، سید هاشم احمدی، مهدیه فلاح علی‌آبادی و عبدالرضا میرزایی

چکیده: در سال‌های اخیر، مسئله‌ای تحت عنوان آسیب‌پذیری مدل‌های مبتنی بر یادگیری ماشین مطرح گردیده است که نشان می‌دهد مدل‌های یادگیری در مواجهه با آسیب‌پذیری‌ها از مقاومت بالایی برخوردار نیستند. یکی از معروف‌ترین آسیب‌ها یا به بیان دیگر حملات، تزریق مثال‌های تخاصمی به مدل می‌باشد که در این مورد، شبکه‌های عصبی و به ویژه شبکه‌های عصبی عمیق بیشترین میزان آسیب‌پذیری را دارند. مثال‌های تخاصمی، از طریق افزوده‌شدن اندکی نویز هدفمند به مثال‌های اصلی تولید می‌شوند، به طوری که از منظر کاربر انسانی تغییر محسوسی در داده‌ها مشاهده نمی‌شود اما مدل‌های یادگیری ماشینی در دسته‌بندی داده‌ها به اشتباه می‌افتند. یکی از روش‌های موفق جهت مدل‌کردن عدم قطعیت در داده‌ها، فرایندهای گوسی هستند که چندان در زمینه مثال‌های تخاصمی مورد توجه قرار نگرفته‌اند. یک دلیل این امر می‌تواند حجم محاسباتی بالای این روش‌ها باشد که کاربردشان در مسایل واقعی را محدود می‌کند. در این مقاله از یک مدل فرایند گوسی مقیاس‌پذیر مبتنی بر ویژگی‌های تصادفی بهره گرفته شده است. این مدل علاوه بر داشتن قابلیت‌های فرایندهای گوسی از جهت مدل‌کردن مناسب عدم قطعیت در داده‌ها، از نظر حجم محاسبات هم مدل مطلوبی است. سپس یک فرایند مبتنی بر رأی‌گیری جهت مقابله با مثال‌های تخاصمی ارائه می‌گردد. همچنین روشی به نام تعیین ارتباط خودکار به منظور اعمال وزن بیشتر به نقاط دارای اهمیت تصاویر و اعمال آن در تابع هسته فرایند گوسی پیشنهاد می‌گردد. در بخش نتایج نشان داده شده که مدل پیشنهادشده عملکرد بسیار مطلوبی در مقابله با حمله علامت گرادیان سریع نسبت به روش‌های رقیب دارد.

کلیدواژه: **شبکه‌های عصبی، فرایندهای گوسی، فرایندهای گوسی مقیاس‌پذیر، مثال‌های تخاصمی.**

۱- مقدمه

از دیرباز تا کنون، بشر سعی داشته تا با طراحی سامانه‌های مختلف و خودکارسازی امور در زمان و انرژی خود صرفه‌جویی نماید. پیدایش و پیشرفت سیستم‌های کامپیوتری، مباحث نوینی چون هوش مصنوعی و یادگیری ماشینی را برای این جهان به ارمغان آوردند، به نحوی که الگوریتم‌ها و روش‌های امروزی قادرند بسیاری از وظایف انسانی و حتی فراانسانی را با کمترین درنگ به صورت خودکار و با دقت بالا انجام دهند. به موازات پیشرفت علم و فناوری، همواره افرادی نیز بوده‌اند که در جهت کشف نقاط ضعف سامانه‌ها تلاش می‌کردند؛ برخی با هدف رفع ایرادات و تقویت آنها و برخی با هدف سوء و آسیب‌رساندن به آنها. از این رو به تدریج امنیت در مدل‌های یادگیری ماشینی به موضوعی داغ در تحقیقات بدل شده است. آغاز این رویداد به سال ۲۰۰۴ برمی‌گردد، زمانی که برای اولین بار سیستم‌های فیلتر هرزنامه‌ها فریب خوردند [۱].

ظهور و تکامل شبکه‌های عصبی موجب شد که در بسیاری از کاربردها نظیر دسته‌بندی، دست‌یابی به دقت‌های بالا که تا پیش از آن غیر قابل حصول بود، امکان‌پذیر گردد. اما اخیراً موضوعی تحت عنوان حملات تخاصمی بر روی شبکه‌های عصبی و مخصوصاً شبکه‌های عصبی عمیق مطرح گردیده که هدف آن به اشتباه‌انداختن مدل‌هایی است که در بستر خود از شبکه‌های عصبی چندلایه استفاده می‌کنند. در این حملات که اصطلاحاً به آنها حملات تخاصمی[1] گفته می‌شود، داده‌های ورودی به گونه‌ای تغییر می‌یابند که از نگاه انسان، تغییر محسوسی قابل مشاهده نیست اما مدل‌های یادگیری را به شدت به اشتباه می‌اندازند [۱] و [۲]. این حملات می‌توانند با تغییری بسیار کوچک، بعضاً حتی با یک پیکسل در تصویر [۳]، شبکه را دچار خطا نمایند. چنانچه یکی از این حملات در کاربردهای حساسی چون خودروهای خودران رخ دهد، می‌تواند عواقب فاجعه‌آمیزی به بار بیاورد. اگرچه بیشتر کاربردهای مثال‌های تخاصمی در شناسایی اشیا[2] بوده است ولی در کاربردهای دیگری نظیر تشخیص نرم‌افزارهای مخرب[3] [۴] و [۵]، یادگیری تقویتی[4] [۶]، شناسایی صحبت[5] [۷]، شناسایی چهره[6] [۸]، بخش‌بندی معنایی[7] [۹]، پردازش ویدئو[8] [۱۰] و تشخیص شیء[9] [۱۱] نیز این مسأله بررسی شده است.

پژوهش [۱۲]، دلیل آسیب‌پذیری مدل‌های یادگیری عمیق را ذات خطی آنها در فضاهای با بعد بالا عنوان کرده است. از دیگر ضعف‌های مدل‌های مبتنی بر شبکه‌های عصبی عمیق، عدم عملکرد مناسب بر روی

این مقاله در تاریخ ۱۹ فروردین ماه ۱۳۹۹ دریافت و در تاریخ ۴ مرداد ماه ۱۴۰۰ بازنگری شد.

مهران صفایانی (نویسنده مسئول)، دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان، اصفهان، ایران، (emails: safayani@iut.ac.ir).

پویان شالبافان، دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان، اصفهان، ایران، (emails: p.shalbafan@ec.iut.ac.ir).

سید هاشم احمدی، دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان، اصفهان، ایران، (emails: hashem.ahmadi@ec.iut.ac.ir).

مهدیه فلاح علی‌آبادی، دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان، اصفهان، ایران، (emails: mahdieh.fallah@ec.iut.ac.ir).

عبدالرضا میرزایی، دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان، اصفهان، ایران، (emails: mirzaei@iut.ac.ir).

1. Adversarial Attack
2. Object Recognition
3. Malware Detection
4. Reinforcement Learning
5. Speech Recognition
6. Face Recognition
7. Semantic Segmentation
8. Video Processing
9. Object Detection

داده‌هایی در فضای جستجو می‌باشد که احتمال پایینی در توزیع احتمال روی داده‌های آموزش دارند. این موضوع بیانگر این است که داده‌هایی که تنها تفاوت اندکی با داده‌های آموزش دارند، می‌توانند در دسته‌های کاملاً متفاوت قرار گیرند. یکی از روش‌های مقابله با مثال‌های تخاصمی، پایدارسازی دسته‌بندها است. در این روش، مسأله محدود به مثال‌های تخاصمی نبوده و می‌توان بدون نیاز به شبکه‌های کمکی، مقاومت شبکه را تا حد زیادی افزایش داد، فارغ از این که حمله تخاصمی چگونه طراحی شده است. روش‌های مبتنی بر مقاوم‌سازی مدل‌ها نسبت به روش‌هایی نظیر آموزش با مثال‌های تخاصمی یا تشخیص مثال‌های تخاصمی با شبکه‌های کمکی حجم محاسبات بسیار کمتری دارند و مناسب برای کاربردهای عملی هستند، از این رو در این مقاله یک مدل مقاوم در برابر مثال‌های خصمانه پیشنهاد شده است. رویکرد این مقاله برای مقاوم‌سازی، در نظر گرفتن عدم قطعیت داده‌ها در هنگام مدل‌سازی است. شبکه‌های عصبی معمولاً عدم قطعیت را در داده‌ها در نظر نمی‌گیرند و بنابراین هنگامی که در فضای مسأله به دلیل داده‌های کم آموزشی یا انتقال دامنه[1]، عدم قطعیت زیادی وجود دارد، بدون توجه به این عدم قطعیت پیش‌بینی را انجام می‌دهند. این امر می‌تواند شبکه‌های عصبی را در قبال حملات تخاصمی آسیب‌پذیر کند. از طرفی مدل‌های احتمالاتی در مواجهه با داده‌های ورودی نویزی، بهتر از سایر مدل‌ها عمل می‌کنند [۱۳]. با وجودی که فرایندهای گوسی نیز در زمره مدل‌های احتمالاتی قرار می‌گیرند و توانایی زیادی در مدل‌کردن عدم قطعیت در داده‌ها دارند، چندان به جهت مقابله با مثال‌های تخاصمی مورد توجه قرار نگرفته‌اند. یک دلیل این امر می‌تواند حجم بالای محاسبات و وابستگی محاسبات به تعداد داده‌های آموزشی در این روش‌ها باشد. در [۱۴] نشان داده شده است که چگونه می‌توان از ویژگی‌های تصادفی برای تقریب کواریانس یک فرایند گوسی استفاده کرد. در [۱۵] از ویژگی‌های تصادفی برای تقریب کوواریانس تابع پایه شعاعی[2] و کوواریانس آرک کسینوس[3] استفاده استفاده گشته و نشان داده شده است که یک فرایند گوسی عمیق را می‌توان از طریق ویژگی‌های تصادفی با یک شبکه عصبی عمیق مدل کرد. هرچند مدل‌های ارائه‌شده در این مراجع به منظور مقابله با مثال‌های تخاصمی در نظر گرفته نشده‌اند و بحثی در مقالات در این باره وجود ندارد. برای استفاده از این مدل‌ها برای مواجهه با مثال‌های خصمانه بایستی رویکردی جهت مقاوم‌سازی در آنها در نظر گرفته شود و همچنین با ارائه راهکارهایی مقاومت این مدل‌ها در مواجهه با حملات خصمانه افزایش یابد.

در این مقاله از شبکه‌های با ویژگی تصادفی ارائه‌شده در [۱۵] به منظور مقابله با مثال‌های تخاصمی استفاده گردیده است. این روش بار محاسباتی بسیار کمتری نسبت به فرایندهای گوسی استاندارد دارد و در نتیجه از سرعت خوبی در مرحله آموزش بهره می‌برد. سپس برای مقابله با مثال‌های تخاصمی یک رویکرد مبتنی بر رأی‌گیری پیشنهاد می‌شود. فرایند کار بدین صورت است که در شبکه چندین بار از توزیع وزن‌ها نمونه‌گیری می‌گردد و در هر حالت، خروجی مدل به دست می‌آید. نهایتاً با استفاده از روش رأی‌گیری در مورد نتیجه نهایی دسته‌بندی تصمیم‌گیری می‌شود. در این حالت رفتار مدل شبیه به روش کمیته ماشین تصمیم می‌تواند باعث افزایش دقت در مواجهه با مثال‌های تخاصمی گردد.

همچنین به کمک یک روش به نام تعیین ارتباط خودکار، مدل پایه به گونه‌ای توسعه داده می‌شود که بتواند به ویژگی‌هایی که از اهمیت بالاتری در تصمیم‌گیری برخوردار هستند، وزن بیشتری بدهد و این وزن را به گونه‌ای در داخل تابع هسته مدل گوسی وارد نماید. این کار باعث افزایش مقاومت مدل در مواجهه با حملات تخاصمی می‌گردد. در بخش ارزیابی، نتایج مدل پیشنهادی بر روی پایگاه داده‌های استاندارد یادگیری ماشین مورد ارزیابی قرار می‌گیرد و مقاومت مدل‌ها با سایر الگوریتم‌های رقیب مقایسه می‌گردد.

ساختار مقاله در ادامه بدین صورت است که ابتدا در بخش دوم، کارهای پیشین انجام‌شده در زمینه تولید مثال‌های تخاصمی و مقاوم‌سازی شبکه‌های عصبی مرور می‌گردد. در بخش سوم پیش‌زمینه‌های مورد نیاز از جمله شبکه‌های عصبی بیزین و فرایندهای گوسی مقیاس‌پذیر معرفی می‌گردند. در بخش چهارم، روش پیشنهادی برای مقاوم‌سازی در مقابل مثال‌های تخاصمی ارائه می‌شود. در بخش پنجم به ارزیابی روش پیشنهادی و مقایسه آن با مدل‌های دیگر پرداخته می‌شود و در نهایت نتیجه‌گیری و ارائه پیشنهادهایی برای ادامه کار در بخش ششم آمده است.

۲- کارهای مرتبط

۲-۱ روش‌های تولید مثال‌های تخاصمی

اولین بار مفهوم مثال‌های تخاصمی در شبکه‌های عصبی عمیق در [۲] مطرح شد. در این مقاله با استفاده از روش L-BFGS به حل مسأله بهینه‌سازی به صورت (۱) پرداخته شده است

$$\begin{aligned} &\min_{x'} c\|\eta\| + J_\theta(x', l') \\ &\text{s.t. } x' \in [0,1] \end{aligned} \quad (1)$$

در اینجا و در ادامه، l و l' به ترتیب برچسب‌های مربوط به داده اصلی x و داده دارای اختلال x' است. همچنین $\eta = x - x'$ میزان اختلال اضافه‌شده به داده اصلی را نشان می‌دهد. J نیز تابع هزینه‌ای نظیر خطای میانگین مربعات است. در این حمله برای یافتن مقدار مناسب برای ثابت c از روش جستجوی خطی[4] $(c>0)$ استفاده می‌شود. در [۱]، روش علامت گرادیان سریع[5] برای تولید مثال‌های تخاصمی ارائه گردیده است. در این روش، به روز رسانی گرادیان در جهت علامت گرادیان هر پیکسل و تنها در یک گام انجام می‌شود. اختلال به صورت (۲) محاسبه می‌گردد

$$\eta = \varepsilon \operatorname{sign}(\nabla_x J_\theta(x, l)) \quad (2)$$

که در آن ε بزرگی اختلال را تعیین و عبارت sign علامت را مشخص می‌کند. بنابراین مثال‌های تخاصمی از افزودن اختلال به داده‌های اصلی به صورت (۳) محاسبه می‌گردند

$$x' = x + \eta \quad (3)$$

برای محاسبه این اختلال می‌توان از الگوریتم پس‌انتشار[6] استفاده کرد. روش علامت گرادیان سریع ضمن باسرعت و عملیاتی‌بودن، روی همه مدل‌هایی که حمله‌کننده به مدل و پارامترها دسترسی دارد، قابل اجرا است. در این مقاله نیز از این حمله برای ارزیابی مقاومت مدل‌های مطرح‌شده استفاده می‌گردد.

1. Domain Shift
2. Radial Basis Function
3. Arc Cosine
4. Linear Search
5. Fast Gradient Sign Method
6. Back Propagation

حملات به دو دسته حملات جعبه سفید[1] و جعبه سیاه[2] تقسیم می‌شوند. در حملات جعبه سفید، فرض بر این است که حمله‌کننده اطلاعات کاملی از مدل آموزش داده شده شامل ساختار مدل، پارامترها، تعداد لایه‌ها، وزن‌های شبکه و ... در اختیار دارد و از این رو می‌تواند گرادیان مدل را محاسبه نموده و از آن برای تدارک حمله استفاده کند. حملات جعبه سیاه دسته دیگر از حملات هستند که حمله‌کننده به مدل دسترسی ندارد بلکه مانند یک کاربر معمولی صرفاً ورودی و خروجی‌های مدل را در اختیار دارد.

در [۱۶]، محققان به جای در نظر گرفتن علامت گرادیان، مقدار خام گرادیان را به عنوان اختلال در نظر می‌گیرند ($\eta = \nabla_x J_\theta(x,l)$). در این روش هیچ قیدی روی پیکسل‌ها وجود ندارد، لذا می‌توان تصاویری با تفاوت‌های محلی بیشتر تولید کرد. در [۱۷] نیز از ایده علامت گرادیان سریع استفاده می‌شود. در این روش مثال‌های تخاصمی در یک فرایند تکرارشونده تولید می‌گردند. همچنین آنها از طریق به کارگیری یک روش یادگیری گروهی[3]، قابلیت انتقال مثال‌ها را بهبود می‌دهند. استفاده از مثال تخاصمی بر روی مدل‌های دیگر به جز مدلی که مثال از آن ساخته شده است را قابلیت انتقال مثال تخاصمی می‌نامند. در [۱۸] نشان داده شده که روش علامت گرادیان سریع به همراه آموزش با مثال‌های تخاصمی در حمله‌های آشکار یا جعبه سفید پایداری بیشتری داشته و برای حمله‌های مخفی یا جعبه سیاه، استفاده از روش نقاب[4] گرادیان به پایداری بیشتری منتج می‌شود. به علاوه، آنها حمله جدیدی به نام علامت گرادیان سریعِ تصادفی نیز ارائه دادند که در هنگام آموزش از مقداری تصادفی برای به روز رسانی مثال‌های تخاصمی استفاده می‌کند.

در [۱۹]، مثال‌هایی تخاصمی از نوع جعبه سیاه تولید می‌شود. در این تحقیق که در واقع توسعه بهینه‌تری از روش علامت گرادیان سریع است، مثال‌های تخاصمی در قالب یک فرایند تکرارشونده تولید می‌گردند، به نحوی که در هر تکرار برای جلوگیری از تغییرهای بزرگ بر روی پیکسل‌ها، مقادیر هر پیکسل در بازه خاصی برش زده می‌شود. موسوی دزفولی و همکارانش جهت رفع مشکل غیر خطی بودن در ابعاد بالا روشی به نام DeepFool ارائه نمودند که یک حمله تکراری با یک تخمین‌گر خطی است [۲۰]. آنها نشان می‌دهند که هر داده را می‌توان با حرکت‌دادن به سمت مرز تصمیم به دست آمده توسط دسته‌بند، به یک داده تخاصمی تبدیل نمود. همچنین برای مرزهای تصمیمی که خطی نیستند یک راه حل تکرارشونده ارائه نمودند. این روش در مقایسه با روش علامت گرادیان سریع، اختلال کمتری ایجاد می‌کند [۲۱]. در تحقیقی دیگر [۲۲]، همین پژوهشگران یک حمله تخاصمی کلی توسعه دادند به گونه‌ای که اختلال کلی از کمینه اختلال به دست آمده برای هر نمونه ورودی در هر تکرار به دست می‌آید. این کار تا زمانی که بیشتر نمونه‌ها فریب بخورند، ادامه خواهد داشت. آنها همچنین حمله‌ای تدوین نمودند که تنها با تغییر یک پیکسل، مثال‌های تخاصمی تولید می‌کند. سپس یک روش تکامل دیفرانسیلی که از نوع الگوریتم تکاملی است برای یافتن جواب بهینه ارائه می‌دهند. این روش به گرادیان‌های شبکه عصبی نیاز ندارد و می‌تواند از توابع مشتق‌ناپذیر نیز استفاده کند.

در [۲۳] از خروجی لاجیت[5] شبکه عصبی عمیق (خروجی قبل از بیشینه هموار[6]) برای توصیف تصاویر استفاده شده است. همچنین ضریب همبستگی پیرسون[7] که برای بررسی تأثیر متقابل بین دو ورودی مستقل (که در اینجا نویز و تصویر داخل مجموعه داده) می‌باشد به کار گرفته شده است. این مقاله مدعی است که اختلال عمومی (یعنی اختلال به دست آمده به صورت مستقل از تصویر ورودی) دارای ویژگی‌های غالبی است که سبب می‌شود ویژگی‌های تصویر ورودی نسبت به آن مانند نویز به نظر برسد. این مقاله از این یافته استفاده می‌کند و مدلی را ارائه می‌دهد که در آن بدون داشتن داده‌های تصویر آموزشی و به کمک فقط یک مجموعه تصویر مستقل از این داده‌ها، اختلال عمومی جدیدی تولید کند و نشان می‌دهد که اختلال تولیدی، نتایج قابل رقابتی با روش‌هایی دارد که در هنگام ساخت اختلال، تصاویر داده‌های آموزشی را نیز در اختیار دارند. مقاله [۲۴] نشان می‌دهد که مثال‌های تخاصمی، قابل انتقال بین مدل‌های مختلف هستند و این قابلیت به اندازه گرادیان ورودی (برای دسته‌بند مقصد) و پراکندگی[8] تابع هزینه برای دسته‌بند پایه بستگی دارد. مدل‌هایی که پیچیدگی کمتری دارند و به عبارتی منتظم[9] شده‌اند دارای گرادیان کوچک‌تری هستند و در نتیجه در برابر مثال‌های تخاصمی مقاومت‌ترند. مرجع [۲۵] مشابه [۲] دو قید را به صورت هم‌زمان کمینه می‌کند، یکی میزان اختلالی که باعث می‌شود نمونه به عنوان مثال تخاصمی شناخته شود (به عنوان مثال با عدم کلاس‌بندی صحیح توسط مدل) و دیگری نرم[10] L_2 اختلال. در این روش به جای استفاده از روش‌های بهینه‌سازی مقید- جعبه‌ای[11]، پیشنهاد کرده‌اند که با استفاده از تابع tanh بر روی قیود تغییر متغیر انجام شود و به جای بهینه‌سازی تابع آنتروپی متقابل مثال‌های تخاصمی از تفاوت مابین لاجیت‌ها استفاده کنند. برای یک حمله برای ایجاد کلاس هدف T با فرض این که Z خروجی لاجیت باشد، این روش تابع (۴) را بهینه می‌کند

$$\min_\eta [\|x' - x\|_2^2 + Cf(x')] \qquad (۴)$$

که در این رابطه $f(x') = \max(\max_{i\neq t}\{Z(x')_i\} - Z(x')_t, -\kappa)$، $x' = (1/2)(\tanh(\operatorname{arctanh} x + \eta) + 1)$ و $Z(x')_i$ لاجیت مربوط به کلاس i است. با افزایش پارامتر اطمینان κ، نمونه تخاصمی با اطمینان بیشتری به صورت نمونه تشخیص داده نشده در نظر گرفته می‌شود. در این رابطه C ضریب ثابتی است که میزان وزن عبارت دوم نسبت به اول را تنظیم می‌کند و توسط الگوریتم جستجوی خطی[12] مقدارش پیدا می‌شود و این باعث می‌گردد که الگوریتم کند شود چرا که به تعداد تکرار زیادی نیاز دارد.

در [۲۶] ابتدا بردار گرادیان برای تولید مثال خصمانه محاسبه می‌شود و سپس در امتداد بردار گرادیان، اندازه اختلال توسط نگاشت آن به یک کره با شعاع اپسیلون[13] در اطراف تصویر اصلی محدود می‌شود. مقدار اپسیلون توسط بررسی یک شرط تغییر می‌کند، بدین صورت که اگر در گام فعلی تصویر با اضافه‌شدن اختلال یک مثال تخاصمی است مقدار اپسیلون برای گام بعدی کاهش می‌یابد و اگر خلاف آن رخ دهد، مقدار اپسیلون افزایش

1. White Box
2. Black Box
3. Ensemble Learning
4. Mask
5. Logit
6. Soft Max
7. Pearson
8. Variance
9. Regularized
10. Norm
11. Box-Constraint
12. Line Search
13. Epsilon

می‌یابد. بدین ترتیب این روش، مثال‌های تخاصمی نزدیک مرز تصمیم تولید می‌کند، هر چند به دلیل این که در هر گام یک مقدار ثابت به اپسیلون زیاد یا کم می‌شود این الگوریتم کند است. در [۲۷] پیشنهاد شده است که میزان افزایش یا کاهش اپسیلون به صورت خودکار با نگاه‌کردن به نتایج دو گام قبلی تنظیم گردد، به این صورت که اگر در گام فعلی تصویر تولیدی یک مثال تخاصمی باشد (نباشد)، آن گاه دو گام قبلی بررسی می‌شود و اگر نتیجه دو گام قبلی یکسان بود، اپسیلون به میزان بیشتری نسبت به حالتی که نتیجه دو گام قبلی یکسان نباشد کاهش (افزایش) می‌یابد.

۲-۲ مقابله با مثال‌های تخاصمی

دو نوع رویکرد دفاعی برای مقابله با مثال‌های تخاصمی وجود دارد [۲۱]: ۱) پیشگیرانه[1]: ایجاد شبکه‌های عصبی عمیق پایدار قبل از تولید مثال‌های تخاصمی توسط مخرب‌ها و ۲) واکنشی[2]: تشخیص مثال‌های تخاصمی بعد از ساخت شبکه‌های عصبی عمیق. در [۲۸] از تقطیر شبکه که روشی پیشگیرانه برای مقابله با مثال‌های تخاصمی است، بهره گرفته می‌شود. در این روش، ضمن کاهش اندازه شبکه‌های عصبی عمیق، دانش یاد گرفته شده از یک شبکه بزرگ‌تر به شبکه کوچک‌تر منتقل می‌گردد. به این صورت که ابتدا یک شبکه عمیق طراحی می‌گردد و سپس احتمالات به دست آمده در خروجی، به عنوان ورودی به شبکه دوم داده می‌شود. برای نرمال‌سازی لایه آخر معمولاً از تابع بیشینه هموار استفاده می‌گردد. می‌توان در این تابع از یک پارامتر دما برای کنترل سطح تقطیر دانش استفاده نمود. فرایند تقطیر شبکه می‌تواند از اتصال چندین شبکه عصبی عمیق حاصل شود و به این ترتیب، شبکه قادر است ضمن دستیابی به پایداری بالاتر، حساسیت کمتری نسبت به اختلال‌های کوچک از خود نشان دهد. در این مقاله به جای این که از چندین شبکه عصبی در هنگام آموزش استفاده گردد، فقط از یک شبکه استفاده شده و برخلاف فرایند پیشنهادی مقاله از ساختار یکسانی در هنگام آموزش و تقطیر استفاده گردیده است.

استفاده از مثال‌های تخاصمی در کنار داده‌های اصلی، روشی دیگر برای پایدارترکردن شبکه‌های عصبی عمیق می‌باشد [۱]. مشکل این دسته از روش‌ها، حجم بالای محاسباتی مورد نیاز جهت آموزش و نیز آماده‌کردن مثال‌های مناسب تخاصمی است. به طور کلی کیفیت این روش‌ها به مثال‌های تخاصمی ساخته‌شده وابسته است. دسته دیگر از روش‌های پیشگیرانه، پایدارسازهای دسته‌بند هستند که تلاش می‌کنند از ابتدا ساختارهای مقاوم‌تری برای شبکه عصبی عمیق ارائه نمایند. در [۲۹]، محققان ابتدا یک معماری شبکه عصبی عمیق طراحی نمودند و سپس خروجی‌های به دست آمده از لایه استخراج ویژگی را به یک فرایند گوسی مقیاس‌پذیر تُنُک می‌دهند و در انتهای شبکه از یک تابع بیشینه‌سازی احتمالاتی استفاده می‌کنند. هرچند این روش وابستگی زیادی به تعداد نقاط الحاقی انتخابی دارد و تعداد این نقاط می‌تواند تابعی از تعداد داده‌های آموزشی باشد، بنابراین با افزایش حجم داده‌های آموزشی زمان آموزش نیز افزایش زیادی خواهد داشت. دسته دیگر از روش‌های مقابله، از سیاست واکنشی بهره گرفته و به تشخیص مثال‌های تخاصمی در مرحله آزمون می‌پردازند. در [۳۰] از یک شبکه عصبی کمکی در کنار شبکه عصبی اصلی استفاده می‌شود. این شبکه کمکی که از آن به عنوان شناساگر تخاصمی یاد می‌شود، داده‌ها را به دو رده تخاصمی یا غیر تخاصمی دسته‌بندی می‌کند. در این روش به دو شبکه برای آموزش نیاز دارد و علی‌رغم حجم محاسبات زیاد در برابر برخی حملات مقاومت خوبی ندارد. در [۳۱] نشان داده شده که پس از حذف اختلال از تصاویر و با اعمال روش تحلیل مؤلفه‌های اصلی (PCA)[3]، تصاویر تخاصمی ضریب متفاوتی در مؤلفه‌های با رتبه پایین کسب می‌کنند، لذا می‌توان از این خصوصیت به عنوان شناساگر استفاده نمود. همچنین مثال‌های تخاصمی از طریق بازسازی می‌توانند به داده‌های اصلی تبدیل شوند، هرچند نشان داده شده است که اضافه‌کردن نویز گوسی باعث عدم موفقیت روش می‌شود [۳۲]. در [۳۳] بیان شده است که مثال‌های تخاصمی می‌توانند باعث افزایش دقت مدل‌های شناسایی شوند. این روش از یک هنجارساز دسته‌ای[4] کمکی برای مثال‌های تخاصمی استفاده می‌کند و ادعا دارد که این کار باعث جبران تفاوت توزیع داده‌های تمیز و تخاصمی در شبکه می‌شود. این روش نتایج خوبی بر روی بسیاری از مجموعه داده‌ها داشته است ولی حجم محاسبات بالایی دارد. مرجع [۳۴] نشان می‌دهد که قابلیت انتقال زیادی مابین مدل‌ها در دوره‌های[5] همسایه وجود دارد، یعنی مثال‌های تخاصمی تولیدشده در یک دوره در دوره‌های بعدی نیز خاصیت خود

را حفظ می‌کنند و هنوز مثال‌های تخاصمی هستند. با استفاده از این ویژگی، در این مقاله الگوریتمی ارائه شده که می‌تواند مقاومت مدل‌های آموزش‌دیده را ارتقا داده و کارامدی آموزش را به وسیله جمع‌کردن اختلال‌های تخاصمی در دوره‌های آموزش افزایش دهد. هرچند این روش نیاز به مثال‌های تخاصمی برای آموزش دارد و نسبت به روش‌هایی که به ارتقای مدل می‌پردازند حجم محاسبات بالاتری دارد. مرجع [۳۵] فرض می‌کند که شبکه از قبل آموزش یافته است و سعی می‌نماید که مثال‌های تخاصمی را تشخیص دهد. در این روش ابتدا ویژگی‌های استخراج‌شده از شبکه، منتظم[6] می‌گردند و سپس لایه آخر شبکه با استفاده از این ویژگی‌ها دوباره آموزش داده می‌شود. سپس با مقایسه خروجی شبکه اصلی و شبکه تغییریافته می‌توان مثال‌های تخاصمی را تشخیص داد. در روشی دیگر در این مقاله ابتدا بافت نگاشت‌های[7] لایه‌های مخفی شبکه عصبی استخراج می‌شود و سپس این بافت نگاشت‌ها با هم ترکیب شده و به یک دسته‌بند ماشین بردار پشتیبان[8] برای تشخیص مثال‌های تخاصمی تخاصمی داده می‌شود. هرچند این روش اگر بخواهد در یک مسأله شناسایی مورد استفاده قرار گیرد نیاز است که یک شبکه مستقل برای تشخیص مثال‌ها آموزش داده شود و از آن در کنار شبکه اصلی استفاده گردد که این کار باعث افزایش حجم زیاد پردازش می‌شود. در [۳۶] شبکه عصبی به کمک نمونه‌هایی که از ترکیب محدبی از نمونه‌های دوتایی داخل داده‌های آموزشی و برچسب‌هایشان تشکیل شده است، آموزش می‌بینند. این کار سبب می‌شود که شبکه به گونه‌ای منتظم گردد که به نفع رفتار خطی ساده مابین نمونه‌های آموزشی رفتار کند و قابلیت توسعه‌پذیری[9] شبکه افزایش یابد. هرچند عملکرد این روش بیشتر به صورت تجربی ارزیابی شده است و علت کارامدی روش بایستی با عمق

1. Proactive
2. Reactive
3. Principal Component Analysis
4. Batch Normalization
5. Epochs
6. Regularize
7. Histogram
8. Support Vector Machine
9. Generalization

بیشتری بررسی گردد. در [۳۷] یک چهارچوب برای آموزش شبکه عصبی پیشنهاد گردیده که در این چهارچوب سعی می‌شود در هنگام آموزش، ثابت لپسکیتز[1] که یک معیار حساسیت است کوچک نگه داشته شود. تابع $f:\mathbb{R}^n\to\mathbb{R}^m$ به صورت پیوسته لپسکیتز است اگر یک $L\geq 0$ وجود داشته باشد به طوری که (۵) برقرار باشد

$$\|f(x)-f(y)\|\leq L\|x-y\|\ ,\ \forall x,y\in\mathbb{R}^n \qquad (۵)$$

حداقل مقدار L که در (۵) صدق کند ثابت لپسکیتز گفته می‌شود. اگر ورودی از x به y تغییر کند، این ثابت یک حد بالا برای تغییرات F خواهد بود و بنابراین کوچک‌بودن این ثابت نشان‌دهنده حساسیت پایین به ورودی و به عبارتی مقاومت بالاتر به این تغییرات است. در این مقاله سعی شده است که با کمینه‌کردن این ثابت در خلال آموزش، مقاومت شبکه بالاتر رود. این مقاله از آزمون برنامه‌نویسی نیمه‌معین (SDP)[2] در آموزش استفاده می‌کند. با وجود این به دلیل پیچیدگی برنامه‌نویسی نیمه‌معین، زمان آموزش ثبت‌شده تقریباً ۵۰ برابر بیشتر از شبکه‌های غیر مقید است. در [۳۸]، محققان از خروجی‌های به دست آمده از لایه‌های مخفی انتهایی مدل در تابع هزینه بهره گرفته و از این طریق سعی کردند تا ویژگی‌های ارزشمندتری برای هر کلاس به دست آورند، به نحوی که کلاس‌های مختلف در فضای خروجی بهتر از یکدیگر تمایز داده شوند تا به تبع آن مقاومت مدل در برابر حملات تخاصمی افزایش یابد. آنها همچنین مدل خود را با مثال‌های مخرب بازآموزی نموده و به دقت بالاتری نسبت به مدل اولیه دست یافتند.

۳- پیش‌زمینه‌های مورد نیاز

۳-۱ شبکه‌های عصبی بیزین

وزن‌ها در شبکه‌های عصبی استاندارد به صورت احتمالاتی در نظر گرفته نمی‌شوند و توزیع بر روی آنها تعریف نمی‌گردد. آموزش شبکه‌های عصبی استاندارد از طریق روش‌های بهینه‌سازی، معادل تخمین بیشترین درست‌نمایی[3] وزن‌های شبکه است. استفاده از بیشترین درست‌نمایی سبب سبب می‌شود که عدم قطعیت در وزن‌های شبکه در نظر گرفته نشود. شبکه‌های عصبی بیزین برای وزن‌های شبکه توزیع احتمالاتی در نظر می‌گیرد. در صورتی که هیچ داده‌ای وجود نداشته باشد، این توزیع با توزیع پیشین[4] $p(w)$ توصیف می‌گردد که در آن $w\equiv(w_1,\ldots,w_W)$ برداری است که در آن همه وزن‌ها و پیش‌قدرهای[5] شبکه قرار دارد. هنگامی که یک مجموعه داده D به عنوان داده‌های آموزشی در دسترس باشد، می‌توان توزیع پسین[6] $p(w|D)$ بر روی وزن‌ها را به صورت (۶) به دست دست آورد

$$p(w|D)=\frac{p(D|w)p(w)}{P(D)} \qquad (۶)$$

که در آن $p(D|w)$ درست‌نمایی[7] و مخرج یک ضریب هنجارسازی[8] است و به صورت (۷) به دست می‌آید

$$p(D)=\int p(D|w)p(w)\mathrm{d}w \qquad (۷)$$

توزیع برچسب دیده‌نشده $\hat{y}$ داده آزمون $\hat{x}$ توسط (۸) به دست می‌آید

$$p(\hat{y}|\hat{x})=E_{p(w|D)}[p(\hat{y}|\hat{x},w)] \qquad (۸)$$

در این رابطه E نماد امید ریاضی است. به عبارتی در این رابطه هر پیکربندی w توسط توزیع پسین وزن‌دهی می‌شود و با دیدن داده $\hat{x}$ پیش‌بینی‌ای در مورد برچسب $\hat{y}$ انجام می‌دهد. بنابراین امید ریاضی بر روی توزیع پسین وزن‌ها معادل در نظر گرفتن بی‌نهایت شبکه عصبی است. البته این رابطه برای شبکه‌های عصبی، رام‌نشدنی[9] است. در [۳۹] یک راه حل مبتنی بر تقریب تغییراتی[10] تابع توزیع پسین وزن‌ها ارائه شده است. یادگیری تغییراتی سعی در یافتن پارامترهای توزیع q که با θ نشان داده می‌شود، دارد به گونه‌ای که معیار واگرایی کولبک- لیبلر (KL)[11] را به صورت (۹) کمینه کند

$$\begin{aligned}\theta^* &= \arg\min_\theta KL[q(w|\theta)\|p(w|D)]\\ &= \arg\min_\theta \int q(w|\theta)\log\frac{q(w|\theta)}{p(w)p(D|w)}\mathrm{d}w\end{aligned} \qquad (۹)$$

پس از ساده‌سازی، (۱۰) که به نام ELBO[12] شناخته می‌شود به دست می‌آید

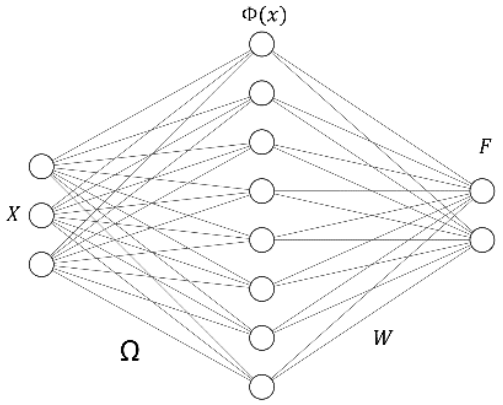
$$\arg\min_\theta KL[q(w|\theta)\|p(w)]-E_{q(w|\theta)}[\log p(D|w)] \qquad (۱۰)$$

در این رابطه سعی می‌شود که پارامترهای توزیع $q(w|\theta)$ به گونه‌ای یاد گرفته شود که این توزیع به توزیع پیشین $p(w)$ نزدیک باشد، در عین این که پیچیدگی داده‌ها را نیز در نظر داشته باشد [۱۳].

۳-۲ فرایندهای گوسی مقیاس‌پذیر

بسیاری از مسایل یادگیری ماشین به دنبال یافتن تابعی هستند که ارتباط بین ورودی‌ها و خروجی‌ها را توصیف نمایند. فرایندهای گوسی، این امکان را فراهم می‌کنند که بتوان مستقیماً بر روی این توابع، توزیع احتمالاتی تعریف نمود. اگرچه ممکن است این کار به دلیل فضای نامحدود توابع دشوار به نظر برسد، اما می‌توان مقادیر توابع را فقط در نقاط داده‌های آموزشی و آزمون که تعداد محدودی هستند، در نظر گرفت. هر فرایند گوسی به عنوان یک توزیع احتمال بر روی تابع $y(x)$ تعریف می‌شود، به گونه‌ای که مجموعه مقادیر $y(x)$ متناظر با مجموعه ورودی دلخواه $x_1,\ldots,x_n$ توزیع گوسی توأم داشته باشند [۴۰]. هر توزیع گوسی از طریق میانگین و ماتریس کواریانس آن مشخص می‌گردد. در بیشتر کاربردها، میانگین برابر با صفر در نظر گرفته می‌شود. در این حالت، ماتریس کواریانس به تنهایی توصیف‌گر یک فرایند گوسی خواهد بود. مؤلفه K_{nm} ام ماتریس کواریانس با استفاده از تابع هسته‌ای که روی نقاط x_n و x_m عمل می‌کند، محاسبه می‌شود. توابع هسته مختلفی برای فرایندهای گوسی وجود دارد. محاسبه توزیع پسین توابع به سادگی و با استفاده از ویژگی‌های توزیع گوسی انجام می‌شود. اگر $\mathcal{N}(y|\mu,K)$ یک توزیع گوسی باشد و بردار y به صورت $[y_a \quad p(y_b|y_a)]y_a^T$ به

1. Lipschitz
2. Semi Definite Programming
3. Maximum Likelihood Estimation
4. Prior
5. Biases
6. Posterior
7. Likelihood
8. Normalization
9. Intractable
10. Variational Approximation
11. Kullback-Leibler
12. Expected Lower Bound



شکل ۱: ساختار مدل فرایند گوسی مقیاس‌پذیر با ویژگی‌های تصادفی.

دست می‌آید [۴۰]

$$\mu = \begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix}$$
$$K = \begin{pmatrix} K_{aa} & K_{ab} \\ K_{ba} & K_{bb} \end{pmatrix} \quad (۱۱)$$

دو قسمت تقسیم شود، میانگین و کواریانس این توزیع به صورت (۱۱) به که در این رابطه μ_a و K_{aa} به ترتیب بردار میانگین و ماتریس کواریانس بردار y_a و K_{ab} ماتریس کواریانس بردارهای y_a و y_b می‌باشد. μ_b، K_{bb} و K_{ba} نیز به طریق مشابه تعریف می‌گردند. همچنین $p(y_b|y_a)$ نیز یک توزیع گوسی دارد که میانگین و کواریانس آن با (۱۲) مشخص می‌شود

$$\mu_{b|a} = \mu_b + K_{ba}K_{aa}^{-1}(y_a - \mu_a)$$
$$K_{b|a} = K_{bb} + K_{ba}K_{aa}^{-1}K_{ab} \quad (۱۲)$$

اگر y_a متناظر با مجموعه داده آموزشی و y_b متناظر با داده آزمون در نظر گرفته شود، آن گاه $p(y_b|y_a)$ نشان‌دهنده توزیع پسین مقادیر تابع در نقاط آزمون پس از مشاهده داده‌های آموزشی خواهد بود. از روابط مربوط به $\mu_{b|a}$ و K_{ba} مشخص است که محاسبه توزیع پسین نیاز به محاسبه معکوس ماتریس K_{aa} دارد. اگر تعداد نمونه‌های آموزشی را n در نظر بگیریم، ماتریس K_{aa} دارای ابعاد $n\times n$ خواهد بود و هزینه محاسباتی لازم برای به دست آوردن معکوس آن از مرتبه $O(n^3)$ خواهد شد. البته می‌توان معکوس ماتریس را یک بار محاسبه نمود و حاصل را برای پیش‌بینی‌های آتی ذخیره کرد. در این حالت، حافظه مورد نیاز از مرتبه $O(n^2)$ و پیش‌بینی در مورد داده‌های آزمون از مرتبه $O(n)$ می‌شود. هزینه محاسباتی و حافظه مصرفی از جمله موارد محدودکننده در استفاده از فرایندهای گوسی برای مجموعه داده‌های بزرگ است. برای حل این مسأله می‌توان فرایندهای گوسی را مقیاس‌پذیر نمود. در این راستا، راهکارهای مختلفی ارائه گردیده که یکی از راهکارها استفاده از روش‌های تُنُک است. در این روش می‌توان به جای استفاده از کل n نمونه داده آموزشی، زیرمجموعه‌ای از آنها به تعداد m انتخاب و استفاده نمود و یا در حالت پیچیده‌تر می‌توان متغیرهای جدیدی با نام متغیرهای القایی ایجاد و به جای داده‌های اصلی با آنها کار نمود [۴۱]. بخش‌بندی داده‌ها راهکار دیگری است که در آن، مسأله اولیه با داده آموزشی بزرگ به مجموعه‌ای از مسایل کوچک‌تر با داده‌های آموزشی کمتر شکسته می‌شود. سپس فرایندهای گوسی جداگانه‌ای در هر قسمت آموزش می‌یابد و در نهایت از ترکیب نتایج تمامی فرایندها برای پیش‌بینی در مورد داده‌های آزمون استفاده می‌شود [۴۲] و [۴۳]. روش‌های مبتنی بر هسته نیز راهکار دیگری است که می‌توان از آن برای مقیاس‌پذیر کردن فرایندهای گوسی استفاده کرد. این روش‌ها بر خلاف راهکارهای قبلی که تقریبی بودند تخمین دقیقی ارائه می‌دهند. آنچه این روش‌ها را مقیاس‌پذیر می‌کند، فرض‌هایی است که در مورد هسته و درست‌نمایی در نظر می‌گیرند. در این روش‌ها، همیشه درست‌نمایی را به صورت توزیع گوسی فرض می‌کنند و هسته قابل استفاده هم محدود به هسته اشتراکی افت نگاشت است [۴۴]. راهکار دیگر، استفاده از ویژگی‌های تصادفی است. این روش‌ها با داشتن داده آموزشی کافی قادرند هر تابع یا مرز تصمیمی را تخمین بزنند [۱۴]. آنها معمولاً فضای ویژگی مسأله را به فضای $\phi(x)$ که دارای بعد بزرگ‌تری است، نگاشت می‌کنند و در فضای جدید با یک تخمین‌گر یا دسته‌بند خطی، مسأله را حل می‌کنند. این انتقال به صورت ضمنی صورت می‌پذیرد، یعنی روابط مسأله به گونه‌ای نوشته می‌شوند که در آن $\phi(x)$ها به صورت ضرب داخلی $\phi(x)^T\phi(x')$ وارد می‌شوند و به جای محاسبه این ضرب از $k(x,x')$ بهره گرفته می‌شود تا هزینه محاسباتی کاهش یابد. با این ترفند، پیش‌بینی در مورد داده آزمون از طریق (۱۳) به دست می‌آید

$$f(x^*) = \sum_{i=1}^{n} c_i k(x_i, x^*) \quad (۱۳)$$

که هزینه محاسباتی آن از مرتبه $O(nd)$ است که n تعداد داده‌های آموزشی و d ابعاد داده می‌باشد. مشخص است که این روش برای داده‌های بزرگ، کارامد نیست. در [۱۴] برای مقیاس‌پذیر کردن روش‌های مبتنی بر هسته، نگاشت فضای ویژگی به صورت صریح انجام می‌گیرد. در این حالت از یک نگاشت ویژگی تصادفی به نام $z(x)$ استفاده می‌گردد. این فضا ضمن داشتن ابعاد کمتر، تقریب خوبی از $\phi(x)$ نیز ارائه می‌کند. به این ترتیب، پیچیدگی محاسباتی به تعداد نقاط داده‌های آموزشی وابسته نیست. چگونگی به دست آوردن فضای ویژگی z که تحت عنوان ویژگی‌های تصادفی معرفی می‌شود در [۱۴] توضیح داده شده است.

٤- مدل پیشنهادی

در این بخش، ابتدا روشی برای ارزیابی مقاومت فرایندهای گوسی مقیاس‌پذیر مطرح می‌شود. سپس مدل پایه توسعه داده شده و ایده تعیین ارتباط خودکار با هدف افزایش دقت و مقاومت مدل پیشنهاد می‌شود.

٤-١ تحلیل فرایند گوسی با ویژگی‌های تصادفی در برابر حملات

یکی از مزایای مدل‌های مبتنی بر شبکه عصبی بیزینی، وجود عدم قطعیت در مدل‌های تصمیم‌گیری است [۱۳]. در این مدل‌ها، تمامی وزن‌های شبکه به جای مقادیر ثابت، توسط توزیع‌های احتمالاتی بیان می‌شوند و بنابراین مدل یادگیری‌شده در برابر اختلال‌های ایجادشده در وزن‌ها، از مقاومت بالاتری برخوردار خواهد بود. حملات تخاصمی نیز چیزی جز مقادیر نویزی اضافه‌شده به داده‌ها نیستند. در [۴۵] گفته شده است که مدل‌های مبتنی بر فرایندهای گوسی مقیاس‌پذیر هم از جهت منابع مورد استفاده در زمان آموزش و هم از حیث میزان دقت نهایی، عملکرد بهتری از سایر مدل‌های پایه بیزینی دارند. بر این اساس، در این مقاله از فرایندهای گوسی مقیاس‌پذیر مبتنی بر ویژگی‌های تصادفی [۱۵] به عنوان مدل پایه استفاده می‌شود. شمای کلی این مدل در شکل ۱ نشان داده شده است. این مدل یک شبکه عصبی دولایه است که لایه

۱) **ورودی‌ها:** مجموعه تصاویر آموزشی و آزمایشی، تعداد دفعات نمونه‌گیری، ضریب اختلال (برای روش FGSM)
۲) **خروجی:** کلاس پیش‌بینی شده برای هر نمونه آزمایشی
۳) **شروع**
۴) مقداردهی اولیه مدل و سپس آموزش آن با تصاویر موجود در مجموعه آموزشی
۵) محاسبه گرادیان مدل نسبت به تصاویر آزمایشی
۶) تولید مثال‌های تخاصمی با استفاده از روش علامت گرادیان سریع (معادلات (۲) و (۳))
۷) به ازای هر تصویر تخاصمی تکرار کن:
۸) به تعداد دفعات نمونه‌گیری تکرار کن:
۹) نمونه‌گیری از پارامترهای مدل
۱۰) دادن تصویر تخاصمی به بخش پیش‌خور شبکه و پیش‌بینی کلاس خروجی برای آن
۱۱) رأی‌گیری روی تمامی نتایج و تعیین کلاس با بیشترین تکرار به عنوان کلاس خروجی
۱۲) کلاس‌های خروجی متناظر با مجموعه آزمایش را برگردان
۱۳) **پایان**

شکل ۲: مکانیزم حمله و ارزیابی مقاومت مدل.

اول ویژگی‌های تصادفی تولید کرده و لایه دوم یک شبکه عصبی بیزین است. خروجی لایه اول که با ϕ نشان داده می‌شود همان تابع هسته است که برای محاسبه آن از دو تابع هسته تابع پایه شعاعی به صورت (۱۴) و تابع آرک کسینوس به صورت (۱۵) استفاده می‌شود

$$\phi_{arc} = \sqrt{\frac{2\sigma^2}{N_{RF}}} \max(0, x.\Omega) \qquad (۱۴)$$

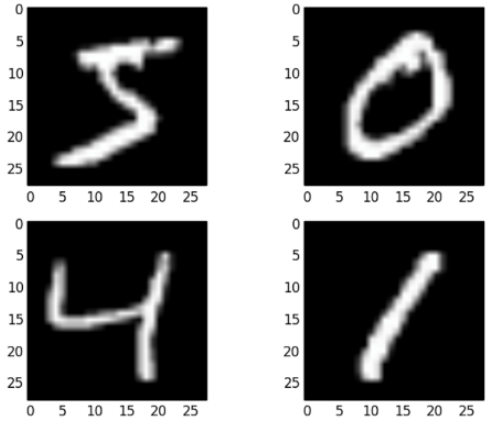
$$\phi_{rbf} = \sqrt{\frac{\sigma^2}{N_{RF}}} [\cos(x.\Omega), \sin(x.\Omega)] \qquad (۱۵)$$

که x ورودی شبکه، σ^2 پراکندگی، N_{RF} تعداد ویژگی‌های تصادفی، Ω ماتریس وزن شبکه در لایه اول با توزیع پیشین $\Omega \sim N(0, (\Lambda)^{-1})$ و عبارت $x.\Omega$ به معنی ضرب داخلی x و Ω است. خروجی شبکه به صورت (۱۶) به دست می‌آید

$$F = \phi W \qquad (۱۶)$$

که W وزن‌های شبکه در لایه دوم با توزیع پیشین $W \sim N(0, I)$ و I ماتریس همانی است.

بعد از آموزش مدل، در این مقاله برای مواجهه با داده‌های تخاصمی، یک روش مبتنی بر رأی‌گیری پیشنهاد می‌شود. در این روش به ازای هر تصویر در مجموعه آزمایش، گرادیان آن محاسبه شده و سپس به کمک الگوریتم علامت گرادیان سریع، تصویری تخاصمی تولید می‌شود. حال این تصویر به مسیر پیش‌خور مدل وارد می‌شود. در ورودی مدل، چندین بار نمونه‌گیری روی پارامترهای شبکه انجام می‌گیرد و برای هر حالت خروجی شبکه محاسبه می‌گردد. در آخر، نتیجه نهایی از رأی‌گیری میان کلاس‌های پیش‌بینی شده توسط این نمونه‌ها حاصل می‌شود. بخش‌های اصلی این مدل، در شکل ۲ آمده است. با نمونه‌گیری از پارامترهای مدل هر بار مدل جدیدی حاصل می‌شود و با رأی‌گیری از نتایج خروجی به طور ضمنی شبیه به ایده کمیته ماشین‌های تصمیم عمل می‌گردد. از آثار استفاده از کمیته ماشین‌های تصمیم، کاهش پراکندگی و افزایش قابلیت توسعه مدل است و در نتیجه، مدل ارائه‌شده نیز شبیه به این روش‌ها



شکل ۳: نمونه‌هایی از مجموعه اعداد دست‌نویس MNIST.

می‌تواند در مواجهه با مثال‌های تخاصمی که حاصل از اضافه‌شدن اختلال بر روی تصویر است عملکرد بهتری داشته باشد.

۴-۲ مدل Boosted-GPRF (B-GPRF)

در [۴۶] مفهومی تحت عنوان تعیین ارتباط خودکار در فرایندهای گوسی معرفی شده که متناظر با داده ورودی x_d (در بعد d)، میزان اهمیت و ارتباط ویژگی ورودی dام را در تابع هسته مشخص می‌کند. این مفهوم به صورت یک پارامتر λ_d در تابع هسته وارد می‌شود. مثلاً تابع هسته تابع پایه شعاعی به صورت (۱۷) در نظر گرفته می‌شود

$$K(x, x') = \sigma^2 \exp[-\frac{1}{2}\sum_{d=1}^{D}(\frac{x_d - x'_d}{\lambda_d})^2] \qquad (۱۷)$$

در این رابطه D تعداد ابعاد داده است و می‌توان این پارامتر را به عنوان یک پارامتر در نظر گرفت که همراه با آموزش مدل مقدار آن تنظیم گردد. این کار سبب می‌شود که تعداد پارامترهای مدل به میزان زیادی افزایش یابد (یک پارامتر به ازای هر بعد داده) و از این رو، در این مقاله یک روش ابتکاری برای تنظیم آن ارائه می‌شود. در مسأله دسته‌بندی اعداد دست‌نویس، مشاهده می‌شود که اطلاعات مهم در مرکز تصویر واقع شده‌اند (پیکسل‌های حاوی اعداد) و گوشه‌های تصویر، اطلاعات مفیدی در بر ندارند (شکل ۳). از این رو میانگین تمام پیکسل‌ها بر روی پایگاه داده‌های آموزشی محاسبه و از مقادیر پیکسل‌ها کم می‌شود. برای جلوگیری از منفی‌شدن، حاصل به توان ۲ رسانده می‌شود. این کار را برای تمامی داده‌ها انجام داده و نهایتاً میانگین همگی حساب می‌شود. برای این که مقدار به دست آمده در محدوده مقادیر پیکسل‌ها باشد، جذر آن را حساب می‌کنیم. واضح است که فرایند ذکرشده، همان محاسبه انحراف از معیار داده‌ها می‌باشد. مقدارهای به دست آمده تحت عنوان پارامتر "تعیین ارتباط خودکار" در یک ماتریس با ابعادی مشابه با تصویر ورودی قرار می‌گیرد. این ماتریس که با Λ نشان داده می‌شود در تابع هسته به صورت (۱۸) و (۱۹) وارد می‌گردد

$$\phi_{arc} = \sqrt{\frac{2\sigma^2}{N_{RF}}} \max(0, (\Lambda^{-1}x).\Omega) \qquad (۱۸)$$

$$\phi_{rbf} = \sqrt{\frac{\sigma^2}{N_{RF}}} [\cos((\Lambda^{-1}x).\Omega), \sin((\Lambda^{-1}x).\Omega)] \qquad (۱۹)$$

جدول ۱: معماری مدل‌های مختلف.

نام مدل	معماری شبکه
۱ Layer NN	$\text{In}(۷۸۴) \rightarrow \text{Softmax}(۱۰)$
۳ Layer NN	$\text{In}(۷۸۴) \rightarrow \text{FC}(۱۰۰) \rightarrow \text{FC}(۱۰۰) \rightarrow \text{Softmax}(۱۰)$
۱ Layer BNN	$\text{In}(۷۸۴) \rightarrow \text{Softplus}(۱۰)$
۱ Layer BNN+mean	$\text{In}(۷۸۴) \rightarrow \text{Softplus}(۱۰)$
CNN CNN+ARD	$\text{In}(۲۸\times۲۸) \rightarrow \text{Conv}۱(F=۵,S=۱,۲۰) \rightarrow \text{Conv}۲(F=۵,S=۱,۵۰) \rightarrow \text{FC}۱(۵۰۰) \rightarrow \text{FC}۲(۱۰۰) \rightarrow \text{Softmax}(۱۰)$
[۲۸]	$\text{In}(۷۸۴) \rightarrow \text{FC}(۵۱۲) \rightarrow \text{Dropout}(۰/۵) \rightarrow \text{FC}(۲۵۶) \rightarrow \text{Dropout}(۰/۵) \rightarrow \text{FC}(۱۲۸) \rightarrow \text{FC}(۱۰)$
[۳۸]	$\text{In}(۲۸\times۲۸) \rightarrow [\text{conv}۱(F=۵,S=۱,۳۲) \rightarrow \text{PReLu}(۲)]\times۲ \rightarrow [\text{conv}۱(F=۵,S=۱,۱۶۴) \rightarrow \text{PReLu}(۲)]\times۲ \rightarrow [\text{conv}۱(F=۵,S=۱,۱۲۸) \rightarrow \text{PReLu}(۲)]\times۲ \rightarrow \text{FC}(۵۱۲) \rightarrow \text{FC}(۶۴) \rightarrow \text{FC}(۱۰)$
GPRF+ARC	$\text{In}(۷۸۴) \rightarrow \text{FC}(۴۰۰۰) \rightarrow \text{Softplus}(۱۰)$
GPRF+RBF	$\text{In}(۷۸۴) \rightarrow \text{FC}(۴۰۰۰) \rightarrow \text{Softplus}(۱۰)$
B-GPRF	$\text{In}(۷۸۴) \rightarrow \text{FC}(۴۰۰۰) \rightarrow \text{Softplus}(۱۰)$

جدول ۲: مقایسه مقاومت مدل‌های مختلف در حمله گرادیان سریع بر حسب دقت دسته‌بندی به درصد بر روی MNIST.

ضریب اختلال / نام مدل	۰	۰/۰۵	۰/۱	۰/۱۵	۰/۲	۰/۲۵	۰/۳	۰/۳۵	۰/٤	۰/٤۵	۰/۵
۱ Layer NN	۹۱	۵۹	۲۰	۰	۰	۰	۰	۰	۰	۰	۰
۳ Layer NN	۹۶	۸۸	۶۹	۴۳	۲۷	۱۷	۱۱	۸	۵	۴	۲
۱ Layer BNN	۹۲	۸۳	۶۹	۵۴	۴۵	۳۶	۲۸	۲۱	۱۵	۱۱	۷
۱ Layer BNN+mean	۹۱	۷۵	۴۵	۳۷	۲۸	۱۴	۸	۵	۳	۱	۰
CNN	۹۹	۹۲	۶۷	۳۹	۲۵	۱۷	۱۲	۹	۶	۵	۴
CNN+ARD	۹۹	۹۳	۷۰	۴۰	۲۴	۱۶	۱۰	۷	۵	۴	۳
[۲۸]	۹۸	۷۹	۴۳	۳۳	۲۸	۲۵	۲۳	۲۲	۲۰	۱۹	۱۸
[۳۸]	۹۹	**۹۷**	**۹۳**	۸۰	۶۹	۵۷	۳۹	۲۲	۱۹	۱۶	۹
GPRF+ARC	۹۲	۹۱	۸۱	۶۲	۹۳	۲۵	۱۹	۱۳	۹	۴	۱
GPRF+RBF	۹۵	۹۱	۸۴	۷۳	۶۰	۴۶	۳۳	۲۲	۱۴	۸	۴
B-GPRF	۹۶	۹۴	۹۱	**۸۶**	**۸۱**	**۷٤**	**٦٦**	**۵۷**	**٤۸**	**۳۹**	**۳۱**

در این رابطه $\circ$ نشان‌دهنده ضرب هادامارد[1] است. نتایج عملی نشان می‌دهد که با انجام این کار، ضمن حفظ سرعت و دقت، مقاومت مدل نیز تا حد قابل قبولی افزایش می‌یابد. این مدل B-GPRF نامیده می‌شود.

۵- ارزیابی و مقایسه

در این بخش به ارزیابی و مقایسه روش‌های پیشنهادشده در این مقاله پرداخته می‌شود. در هر مورد، پس از پیاده‌سازی مدل، حمله‌ای از نوع علامت گرادیان سریع روی آن اعمال شده و نتایج حاصل‌شده از مقاومت مدل، گزارش و تحلیل می‌گردد. در تمامی آزمایش‌ها از مجموعه داده اعداد دست‌نویس (MNIST) [۴۷] و همچنین مجموعه داده HODA [۴۸] شامل اعداد فارسی استفاده شده و نیز در همه حالات، نرخ یادگیری روی ۰/۰۱ تنظیم و تعداد دفعات آموزش به ۱۰ محدود گردیده است.

۵-۱ مدل‌های رقیب

ابتدا به بررسی رفتار مثال‌های تخاصمی در شبکه عصبی یک‌لایه پرداخته می‌شود. جزئیات معماری این شبکه و همچنین دقت به دست آمده بر روی داده‌ها در سطر اول جدول ۱ نشان داده شده است. پس از آموزش مدل، حمله‌ای از نوع علامت گرادیان سریع همان طور که در بخش ۲-۱ توضیح داده شده است بر روی داده‌های آزمون پیاده‌سازی می‌گردد و پس از آن، این داده‌ها مجدداً به بخش پیش‌خور شبکه وارد می‌شوند. این فرایند با ضرایب اختلال گوناگون تکرار شده و دقت‌های به دست آمده در جدول‌های ۲ و ۳ به ترتیب بر روی داده‌های MNIST و HODA گزارش شده است. ضریب اپسیلون صفر به معنی این است که تصویر بدون اختلال به مدل وارد شده است. نتایج نشان می‌دهد که مدل یک لایه با توجه به این که ساختاری خطی در فضای دسته‌بندی ایجاد می‌کند به راحتی و با کمترین میزان اختلال، دچار افت شدید در دقت می‌شود، به نحوی که دقت به دست آمده در ضرایب اختلال بالاتر از ۰/۱ تقریباً برابر با صفر است.

در گام دوم، یک شبکه عصبی سه‌لایه با معماری مشابه مدل قبل، اما با دو لایه میانی تمام‌متصل با ۱۰۰ نورون مورد آزمایش قرار می‌گیرد. دقت به دست آمده و جزئیات مدل آموزش‌دیده در سطر دوم جدول ۱ آورده شده است. حمله گرادیان سریع روی داده‌های آزمون پیاده‌سازی و به ورودی مدل داده می‌شود. نتایج نشان می‌دهد که با افزایش تعداد لایه‌های مدل، مقاومت مدل در برابر مثال‌های تخاصمی افزایش پیدا می‌کند. یکی از دلایل اصلی این اتفاق، خطی‌نبودن این مدل در فضای تصمیم‌گیری است، اما همچنان با افزایش مقادیر ضریب اختلال، دقت مدل دچار افت شدید می‌شود (جداول ۲ و ۳).

در آزمایش بعد، مدل‌های احتمالاتی مورد ارزیابی قرار می‌گیرند. بر این مبنا، مدلی از نوع شبکه عصبی بیزینی یک‌لایه پیاده‌سازی می‌گردد. معماری این شبکه مشابه با شبکه عصبی یک‌لایه است با این تفاوت که

1. Hadamard

جدول ۳: مقایسه مقاومت مدل‌های مختلف در حمله گرادیان سریع بر حسب دقت دسته‌بندی به درصد بر روی HODA.

نام مدل \ ضریب اختلال	۰	۰٫۰۵	۰٫۱	۰٫۱۵	۰٫۲	۰٫۲۵	۰٫۳	۰٫۳۵	۰٫۴	۰٫۴۵	۰٫۵
۱ Layer NN	۹۱	۶۱	۲۰	۴	۱	۰	۰	۰	۰	۰	۰
۳ Layer NN	۹۸	۹۳	۸۱	۶۰	۳۸	۲۱	۱۱	۶	۳	۲	۱
۱ Layer BNN	۹۲	۸۷	۷۹	۷۰	۶۱	۵۲	۴۲	۳۳	۲۶	۱۹	۱۴
۱ Layer BNN+mean	۹۲	۸۶	۸۰	۶۸	۵۲	۳۸	۲۶	۱۸	۱۳	۱۰	۹
CNN	۹۹	۹۴	۷۰	۴۲	۲۹	۲۴	۱۸	۱۴	۱۱	۸	۶
CNN+ARD	۹۹	۹۴	۷۲	۴۵	۳۱	۲۶	۱۹	۱۶	۱۰	۸	۵
[۲۸]	۹۸	۸۴	۶۶	۵۲	۴۲	۳۴	۲۹	۲۶	۲۴	۲۲	۲۱
[۳۸]	۹۹	**۹۶**	**۹۰**	۷۵	۶۶	۵۸	۴۲	۲۴	۲۱	۱۸	۱۱
GPRF+ARC	۹۲	۹۰	۸۴	۷۲	۶۶	۵۱	۴۷	۳۶	۲۵	۱۴	۹
GPRF+RBF	۹۴	۹۱	۸۶	۸۰	۷۳	۶۴	۵۵	۴۶	۳۶	۲۷	۱۹
B-GPRF	۹۴	۹۱	۸۹	**۸۴**	**۷۹**	**۷۳**	**۶۶**	**۵۹**	**۵۲**	**۴۴**	**۳۷**

یک توزیع نرمال روی وزن‌های این شبکه در نظر گرفته شده و همگام با آموزش شبکه، پارامترهای مربوط به وزن‌ها نیز از طریق نمونه‌گیری تنظیم می‌شوند. در لایه آخر نیز از تابع جمع هموار[1] استفاده شده است (سطر سوم جدول ۱). بررسی مثال‌های تخاصمی در این مدل نشان می‌دهد که شبکه‌های عصبی بیزینی به دلیل در نظر گرفتن عدم قطعیت در مدل از مقاومت بالاتری برخوردار هستند و به پیش‌بینی‌های درست‌تری دست می‌یابند. لازم به ذکر است که برای این مدل در زمان آزمایش نیز از نمونه‌گیری استفاده می‌گردد، به این ترتیب که در هر مرحله‌ای که داده‌ها نیاز به عبور از بخش پیش‌خور شبکه دارند، یک بار از پارامترهای شبکه، نمونه‌گیری انجام می‌شود. در شبکه‌های بیزین می‌توان به جای نمونه‌گیری در مرحله آزمایش، از میانگین پسین به دست آمده در مرحله آموزش نیز استفاده نمود. نتایج حاصل از اعمال حمله با استفاده از روش میانگین‌گیری نیز در جدول‌های ۲ و ۳ (سطر چهارم) آورده شده است. مقایسه این مدل با شبکه بیزینی که از نمونه‌گیری در مرحله آزمایش استفاده می‌کند نشان می‌دهد که دقت مدل کمتر شده است. این پدیده به دلیل آزادی عمل شبکه در انتخاب پارامترها از توزیع یاد گرفته شده اتفاق می‌افتد. به عبارت دیگر، در حالت اول به ازای هر بار که بخش پیش‌خور شبکه به اجرا درمی‌آید، وزن‌ها از توزیع به دست آمده نمونه‌گیری می‌شوند. این روند باعث ایجاد تنوع در بازه انحراف معیار توزیع انتخاب‌شده برای وزن‌ها می‌شود و از این رو نتایج بهتری را به دست می‌دهد.

یکی از محبوب‌ترین مدل‌های یادگیری به ویژه برای داده‌های تصویری، شبکه‌های پیچشی عمیق هستند. از آنجایی که این شبکه‌ها ویژگی‌های متنوع و سلسله‌مراتبی از تصویر استخراج می‌کنند، به دقت بالاتری نسبت به سایر مدل‌ها دست یافته‌اند. جزئیات معماری این شبکه در سطر چهارم جدول ۱ آمده است. با وجودی که این شبکه دقت بسیار بالایی روی مجموعه آزمون به دست می‌دهد اما مقاومت پایینی در مقابله با مثال‌های تخاصمی از خود نشان می‌دهد (سطر پنجم از جداول ۲ و ۳).

همچنین نتایج الگوریتم‌ها با دو روش [۲۸] و [۳۸] مقایسه شده‌اند. ساختار شبکه این مدل‌ها در جدول ۱ (سطر ۷ و ۸) آورده شده و دقت پایه آنها بر روی دو مجموعه داده در جدول‌های ۲ و ۳ (سطر هفتم و هشتم) قابل مشاهده است.

1. Softplus

۵-۲ مدل‌های مبتنی بر فرایندهای گوسی

دیدیم که شبکه‌های عصبی مبتنی بر بیزین، رفتار مقاوم‌تری در برابر حملات تخاصمی دارند، لذا در گام بعد دیگر مشتقات شبکه‌های عصبی احتمالاتی بیزینی مورد بررسی قرار می‌گیرند. در بخش ۳-۲ گفته شد که آموزش و استفاده از فرایندهای گوسی، زمان‌بر بوده و دارای پیچیدگی زمانی از مرتبه $O(n^3)$ است که n تعداد داده‌های آموزشی است. برای حل این مشکل باید از فرایندهای گوسی مقیاس‌پذیر استفاده گردد و بنابراین مدل بعدی که مورد ارزیابی قرار می‌گیرد، فرایند گوسی مبتنی بر ویژگی‌های تصادفی است (GPRF). این مدل یک شبکه دولایه است که در لایه اول نگاشت توسط دو تابع هسته آرک کسینوس و تابع پایه شعاعی به فضای ویژگی ۴۰۰۰ بعدی انجام می‌شود و لایه دوم یک شبکه عصبی بیزین است. ساختار شبکه‌ها در جدول ۱ نشان داده شده است. بررسی رفتار این مدل‌ها در مواجهه با مثال‌های تخاصمی نشان می‌دهد که این مدل‌ها از مقاومت بهتری برخوردار هستند (جداول ۲ و ۳). لازم به ذکر است که نتایج روش‌های پیشنهادی توسط رأی‌گیری ارائه‌شده در شکل ۲ به دست آمده است.

مدل B-GPRF توسعه‌ای از مدل GPRF است که در تابع هسته آن، ضریبی تحت عنوان تعیین ارتباط خودکار وارد شده است. همان طور که در بخش ۴-۲ نیز گفته شد، این مقدار به عنوان یک ضریب اهمیت به مدل اضافه می‌گردد. بررسی حمله گرادیان سریع بر روی داده‌های آزمون در این حالت نشان می‌دهد که مقاومت مدل، افزایش قابل توجهی داشته است. این ضریب موجب می‌شود که مدل آموزش‌دیده، قسمت‌هایی را که حاوی اطلاعات مفیدتری هستند در تصمیم‌گیری لحاظ کند. در این صورت اگر در حملات طراحی‌شده نواحی از عکس که برای مدل چندان اهمیتی ندارد، مورد هدف قرار گیرد احتمال این که مدل به اشتباه بیفتد تا حد زیادی کاهش می‌یابد. بنابراین استفاده از تعیین ارتباط خودکار، به صورت ابتکاری در این مقاله موجب می‌شود که مدل به دقت و مقاومت بالاتری دست یابد. نتایج روش‌های مختلف در بازه‌ای از اختلال‌ها بر روی پایگاه داده MNIST در جدول ۳ و بر روی پایگاه داده HODA در جدول ۴ نشان داده شده است. مشابه [۳۸] میزان اختلال از ۰٫۰۵ تا ۰٫۵ با طول گام ۰٫۰۵ در نظر گرفته شده است. منظور از اختلال، ضریب اپسیلون در حمله علامت گرادیان سریع است. هرچه میزان اختلال بیشتر باشد قابلیت تشخیص صحیح مثال تخاصمی ایجادشده سخت‌تر گردیده و در نتیجه افت دقت روش مورد ارزیابی بیشتر خواهد شد.

جدول ۴: مقایسه مقاومت مدل پیشنهادی در حمله گرادیان سریع برای تعداد ویژگی‌های متفاوت بر حسب دقت دسته‌بندی به درصد بر روی MNIST.

# Random Features	Epsilon										
	۰	۰٫۰۵	۰٫۱	۰٫۱۵	۰٫۲	۰٫۲۵	۰٫۳	۰٫۳۵	۰٫٤	۰٫٤۵	۰٫۵
۱۰۰	۸۶	۶۳	۳۳	۱۰	۲	۰	۰	۰	۰	۰	۰
۳۰۰	۹۲	۸۰	۶۰	۳۵	۱۵	۵	۱	۰	۰	۰	۰
۵۰۰	۹۳	۸۵	۷۰	۵۰	۲۹	۱۳	۴	۱	۰	۰	۰
۱۰۰۰	۹۴	۹۰	۸۱	۶۸	۵۲	۳۶	۲۲	۱۲	۶	۲	۱
۱۵۰۰	۹۵	۹۱	۸۴	۷۵	۶۳	۴۹	۳۶	۲۴	۱۵	۸	۴
۲۰۰۰	۹۶	۹۲	۸۷	۷۹	۷۰	۵۹	۴۶	۳۴	۲۴	۱۶	۱۰
۲۵۰۰	۹۶	۹۲	۸۸	۸۲	۷۴	۶۵	۵۵	۴۳	۳۳	۲۴	۱۷
۳۰۰۰	۹۶	۹۳	۸۹	۸۴	۷۸	۶۹	۵۹	۴۹	۳۹	۳۰	۲۲
۳۵۰۰	۹۵	۹۳	۹۰	۸۵	۷۹	۷۱	۶۳	۵۴	۴۴	۳۶	۲۸
٤۰۰۰	۹۶	۹۴	۹۱	۸۶	۸۱	۷۴	۶۶	۵۷	۴۸	۳۹	۳۱
٤۵۰۰	۹۶	۹۴	۹۱	۸۷	۸۲	۷۶	۶۸	۵۹	۵۱	۴۲	۳۳
۵۰۰۰	۹۶	۹۴	۹۱	۸۸	۸۳	۷۷	۷۱	۶۳	۵۵	۴۷	۳۹

۳-۵ تحلیل نتایج

با توجه به بررسی جدول‌های ۲ و ۳ مشخص است که مدل B-GPRF دقت اولیه کمتری نسبت به مدل‌های دیگر نظیر شبکه‌های پیچشی دارد (۹۶ درصد در مقابل ۹۹ درصد) ولی با توجه به نتایج این جدول‌ها مشخص است که این روش مقاومت بسیار بالاتری در مقابل حملات تخاصمی از خود نشان می‌دهد. به عنوان مثال برای میزان اختلال ۰٫۱۵، ۴۷ درصد نتایج B-GPRF نسبت به CNN افزایش یافته است. همچنین برای اختلال‌های بالاتر نظیر ۰٫۴، این روش ۴۲ درصد بهتر از CNN عمل کرده است. این نتایج بسیار قابل توجه است و مقاومت بالای این روش را نسبت به سایر مدل‌های رقیب نشان می‌دهد. همچنین با مشاهده نتایج بر روی پایگاه داده HODA نیز مشخص است که روش پیشنهادی مقاومت بالایی نسبت به سایر روش‌های رقیب داشته است. به عنوان مثال نسبت به روش‌های [۲۸] و [۳۸] توانسته است به ترتیب ۱۶ و ۲۶ درصد بهبود بر روی ضریب اختلال ۰٫۵ به دست آورد. در جدول‌های ۲ و ۳ نتایجی که به صورت پررنگ نوشته شده است، بهترین نتایج در ضریب اختلال مربوط به آن ستون است. همان گونه که مشخص است برای هر دو پایگاه داده، الگوریتم پیشنهادی توانسته است از ضریب اختلال ۰٫۱۵ الی ۰٫۵ بهترین نتایج را نسبت به الگوریتم‌های رقیب به دست آورد و فقط در ضریب‌های اختلال ۰٫۰۵ و ۰٫۱ روش [۳۸] نتایج بهتری به دست آورده است.

مقایسه روش‌های مبتنی بر فرایندهای گوسی مقیاس‌پذیر با یکدیگر در جدول‌های ۲ و ۳ نشان می‌دهد که هسته تابع پایه شعاعی (RBF) نسبت به هسته آرک کسینوس (ARC) نه تنها توانسته دقت اولیه بالاتری به دست دهد (۹۴ درصد در مقابل ۹۲ درصد (جدول ۲))، بلکه در ضرایب اختلال مختلف نیز مقاوم‌تر بوده است. (سطرهای هفتم و هشتم جدول‌های ۲ و ۳). به عنوان مثال در ضریب اختلال ۰٫۱۵، در جدول ۲ هسته تابع پایه شعاعی بهبود ۱۱ درصدی نسبت به هسته آرک کسینوس نشان می‌دهد. بنابراین در روش B-GPRF که نسخه توسعه‌یافته‌ای از روش GPRF می‌باشد، فقط از هسته تابع پایه شعاعی استفاده شده است. همچنین مشاهده می‌شود که روش B-GPRF نسبت به GPRF توانسته است به میزان قابل توجهی بهبود ایجاد کند. به عنوان نمونه در ضریب اختلال ۰٫۲ بر روی پایگاه MNIST به میزان ۱۹ درصد بهتر از $\mathrm{GPRF}+\mathrm{RBF}$ عمل کرده و یا در اختلال ۰٫۲۵، ۲۸ درصد دقت بیشتر کسب کرده است. همین نتیجه بر روی پایگاه داده HODA هم قابل مشاهده است و عملکرد B-GPRF از GPRF-RBF به میزان قابل توجهی بیشتر می‌باشد.

با توجه به موفقیت به کارگیری ضریب تعیین ارتباط خودکار (ARD) در فرایندهای گوسی، در آزمایشی دیگر به بررسی تأثیر آن بر روی شبکه‌های عصبی پیچشی که بالاترین دقت اولیه روی مجموعه آزمون را از آن خود کرده‌اند، پرداخته می‌شود. برای این منظور، این مقدار به صورت ضریبی بر روی ورودی‌های شبکه اعمال می‌شود. بدین صورت که اگر x ورودی شبکه و λ ضریب ARD محاسبه شده باشد، آن گاه ورودی‌های شبکه به شکل $x=x\times\lambda$ درمی‌آیند. همان گونه که از جدول ۲ قابل تشخیص است، تکنیک تعیین ارتباط خودکار بر روی شبکه عصبی عمیق $(\mathrm{CNN}+\mathrm{ARD})$ تأثیر مثبت ناچیزی داشته و نسبت به مدل CNN بهبود چندانی از خود نشان نداده است. مثلاً در میزان اختلال ۰٫۱۵ در پایگاه MNIST تنها توانسته یک درصد بهبود ایجاد نماید. بنابراین میزان موفقیتی که این روش بر روی مدل GPRF به دست می‌آورد، بسیار قابل ملاحظه‌تر از تأثیر آن بر روی دیگر مدل‌های پایه است.

برای به دست آوردن تعداد ویژگی‌های مناسب روش B-GPRF، آزمایش‌هایی با فرض تعداد ویژگی متفاوت انجام گرفته که نتایج آن در جدول ۴ قابل مشاهده است. همان گونه که در این جدول دیده می‌شود به ازای تعداد ویژگی‌های مختلف، از ۱۰۰ تا ۵۰۰۰ روش مورد ارزیابی قرار گرفته است. با توجه به این که از بعد از تعداد ویژگی ۴۰۰۰ با افزایش تعداد ویژگی، میزان بهبود کم بوده است ولی حجم محاسبات به میزان زیادی افزایش یافته است، بنابراین در آزمایش‌ها از این تعداد ویژگی استفاده شده است.

اثر تعداد نمونه‌گیری بر روی نتایج در جدول ۵ نشان داده شده است. همان گونه که در این جدول مشاهده می‌شود با افزایش تعداد نمونه‌گیری، دقت مدل افزایش می‌یابد. از آنجایی که بعد از ۳۰ نمونه‌گیری افزایش دقتی مشاهده نمی‌شود، در این مقاله تعداد نمونه‌ها ۳۰ در نظر گرفته شده است. افزایش تعداد نمونه‌گیری از مدل یک اثر جالب توجه دیگر نیز دارد. در مدل پیشنهادی می‌توان میزان اطمینان را نیز محاسبه کرد، به این صورت که درصد مواردی را که در نمونه‌گیری‌های مختلف نتایج یکسان حاصل می‌شود به صورت میزان اطمینان مدل در نظر گرفته شود. در شکل ۴ میزان اطمینان مدل B-GPRF برای سه حالت بدون رأی‌گیری، ۵ و ۳۰ بار رأی‌گیری نشان داده شده است. همان گونه که

جدول ۵: مقایسه مقاومت مدل پیشنهادی در حمله گرادیان سریع با تعداد نمونه‌گیری متفاوت بر حسب دقت دسته‌بندی به درصد بر روی MNIST.

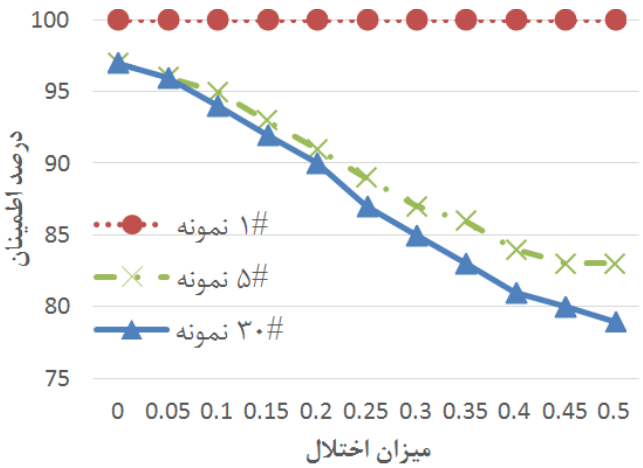
# Sampling	Epsilon	۰	۰/۰۵	۰/۱	۰/۱۵	۰/۲	۰/۲۵	۰/۳	۰/۳۵	۰/٤	۰/٤۵	۰/۵
	۱	۹۵	۹۲	۸۸	۸۴	۷۸	۷۱	۶۲	۵۴	۴۴	۳۷	۲۹
	۲	۹۵	۹۲	۸۹	۸۴	۷۹	۷۳	۶۵	۵۷	۴۹	۴۱	۳۳
	۳	۹۵	۹۳	۸۹	۸۵	۷۹	۷۲	۶۴	۵۴	۴۵	۳۷	۲۹
	٤	۹۵	۹۳	۹۰	۸۵	۷۹	۷۳	۶۵	۵۷	۴۹	۴۱	۳۳
	۵	۹۵	۹۳	۹۰	۸۵	۸۰	۷۳	۶۵	۵۶	۴۷	۳۸	۳۱
	٦	۹۵	۹۳	۹۰	۸۶	۸۱	۷۴	۶۶	۵۸	۴۹	۴۰	۳۲
	۷	۹۵	۹۳	۹۰	۸۵	۸۰	۷۳	۶۵	۵۷	۴۷	۳۹	۳۱
	۸	۹۵	۹۳	۹۰	۸۶	۸۰	۷۳	۶۵	۵۷	۴۸	۳۹	۳۱
	۹	۹۵	۸۳	۹۰	۸۶	۸۰	۷۴	۶۵	۵۶	۴۷	۳۹	۳۰
	۱۰	۹۶	۹۴	۹۱	۸۷	۸۲	۷۶	۶۸	۵۹	۵۱	۴۲	۳۳
	۲۰	۹۶	۹۴	۹۰	۸۶	۸۱	۷۴	۶۶	۵۷	۴۹	۴۰	۳۲
	۳۰	۹۶	۹۴	۹۱	۸۶	۸۱	۷۴	۶۶	۵۷	۴۸	۳۹	۳۲
	٤۰	۹۶	۹۴	۹۱	۸۷	۸۱	۷۴	۶۷	۵۷	۴۹	۴۰	۳۲
	۵۰	۹۶	۹۴	۹۱	۸۶	۸۱	۷۴	۶۶	۵۷	۴۹	۴۱	۳۳
	۱۰۰	۹۶	۹۴	۹۱	۸۶	۸۱	۷۴	۶۶	۵۷	۴۸	۳۹	۳۱

مشاهده می‌شود در حالت بدون رأی‌گیری چون فقط یک بار مدل مورد ارزیابی قرار می‌گیرد میزان اطمینان در همه اختلال‌ها ۱۰۰ درصد است، یعنی به عبارتی حتی در مواردی که مدل به دلیل اختلال دقت پایینی هم دارد ولی به خروجی خود مطمئن است و این عملکرد مناسبی نیست. چون انتظار این است که وقتی مدل دارد اشتباه می‌کند میزان اطمینانش به خروجی کاهش یابد. از طرفی در هنگامی که از رأی‌گیری استفاده می‌شود، هرچه میزان اختلال بیشتر باشد میزان اطمینان مدل کاهش می‌یابد. یعنی وقتی اختلال به بیشترین میزان خود می‌رسد میزان اطمینان مدل به خروجی خود نیز کمتر می‌شود. این می‌تواند یک معیار قابل توجه برای تشخیص مثال‌های تخاصمی باشد. به عبارتی اگر میزان اطمینان مدل از یک سطح آستانه کمتر شد می‌توان تشخیص داد که داده ورودی یک مثال تخاصمی است. همچنین از این شکل مشاهده می‌گردد که هر چه تعداد نمونه‌گیری بیشتر باشد، میزان اطمینان برای داده‌های تخاصمی کاهش می‌یابد. در شکل ۵ چند نمونه تصویر تخاصمی تولیدی نشان داده شده است. در ستون اول تصاویر اصلی نشان داده شده و در ستون‌های کناری تصاویر تخاصمی با درجات مختلف اختلال قابل مشاهده است. همان گونه که در شکل دیده می‌شود، مدل B-GPRF همه تصاویر ستون اول را به درستی پیش‌بینی کرده و درصد اطمینان آنها هم ۱۰۰٪ بوده است. در ستون آخر که تصاویر با بیشترین اختلال هستند، مدل توانسته دو مورد را به درستی تشخیص دهد. نکته قابل توجه در این شکل این است که وقتی میزان اختلال بیشتر می‌شود درجه قطعیت مدل کاهش می‌یابد. به عنوان نمونه در ستون آخر، میزان اطمینان داده‌ها به طور متوسط ۷۳٪ می‌باشد و این در حالی است که میزان اطمینان متوسط در ستون اول و دوم به ترتیب ۱۰۰٪ و ۹۷٪ است. یعنی می‌توان از این معیار هم برای تشخیص این که تصاویر مورد حمله قرار گرفته است نیز استفاده کرد.

علت بهبودهای ایجادشده در روش BGPRF به عنوان مدل پیشنهادی می‌تواند در چند جنبه باشد. یکی این که این مدل یک روش مبتنی بر فرایندهای گوسی مقیاس‌پذیر است که در آن عدم قطعیت در داده‌ها در مدل لحاظ گردیده است و از طرفی برخلاف سایر روش‌های مقیاس‌پذیر فرایندهای گوسی، نگاشت به زیرفضا به صورت صریح انجام می‌شود و در نتیجه حجم محاسباتش کاهش یافته و می‌توان ابعاد زیرفضا را افزایش داد. دیگر این که روش رأی‌گیری اعمال‌شده باعث می‌گردد که این مدل شبیه به روش‌های کمیته ماشین‌های تصمیم‌گیر به طور ضمنی از ترکیب چند مدل برای تصمیم‌گیری استفاده کند که این کار باعث می‌شود که پراکندگی کاهش پیدا کرده و توسعه‌پذیری بیشتر شود و در نتیجه در مواجهه با مثال‌های تخاصمی که با افزودن نویز به تصاویر حاصل می‌گردند، عملکرد بهتری داشته باشد. توجه به قسمت‌های با اهمیت تصویر هم می‌تواند در افزایش دقت در مواجهه با مثال‌های تخاصمی مؤثر باشد، چرا که روش حمله علامت گرادیان سریع به این نقاط مهم بی‌توجه است و ممکن است اختلال را بر روی بخش غیر مهم تصویر اعمال کند، بنابراین با عدم تمرکز بر نقاط غیر مهم می‌توان دقت شناسایی روش را افزایش داد. از نقاط ضعف روش پیشنهادی می‌توان به دقت پایین‌تر آن نسبت به مدل‌های رقیب در هنگامی که اختلال تخاصمی وجود ندارد اشاره کرد که می‌توان در آینده با ارائه مدل‌های عمیق‌تر و ترکیب این ایده با شبکه‌های پیچشی بر آن فایق آمد.

٦- نتیجه‌گیری

در سال‌های اخیر، حملات تخاصمی متعددی با انگیزه‌های مختلف، الگوریتم‌های یادگیری ماشینی را مورد هدف قرار داده‌اند. انجام این حملات در مسایلی نظیر خودروهای خودران و تشخیص بیماری‌ها از روی داده‌های پزشکی فاجعه‌آمیز خواهد بود. برای حل این مسأله، روش‌هایی در سال‌های اخیر پیشنهاد شده که در دسته‌های دفاعی یا پیشگیرانه جای می‌گیرند. در این مقاله از مدل فرایندهای گوسی مقیاس‌پذیر مبتنی بر ویژگی‌های تصادفی استفاده گردید و یک روش رأی‌گیری به جهت مقابله با مثال‌های تخاصمی پیشنهاد شد. همچنین با استفاده از روش تعیین ارتباط خودکار، مدل پایه به گونه‌ای ارتقا داده شد که به داده‌های مهم وزن بیشتری در تابع هسته داده شود و نشان داده شد که مدل نهایی مقاومت بالایی در برابر حملات تخاصمی دارد. همچنین نشان داده شد که با افزایش میزان اختلال، درصد اطمینان مدل کاهش می‌یابد که این ویژگی هم می‌تواند به عنوان معیاری به جهت تشخیص مثال‌های تخاصمی مورد نظر قرار گیرد. موارد زیر به عنوان پژوهش‌های آتی



شکل ۴: درصد اطمینان برای روش BGPRF با میزان اختلال مختلف بر روی داده MNIST.

پیشنهاد می‌شود:

- استفاده از سایر روش‌های مقیاس‌پذیر برای فرایندهای گوسی و ترکیب آن با روش مبتنی بر ویژگی تصادفی
- ارائه مدل‌های عمیق مبتنی بر ویژگی‌های تصادفی
- ترکیب روش ارائه‌شده با شبکه‌های پیچشی

مراجع

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. 3rd Int. Conf. on Learning Representations, ICLR'15,* 11 pp., San Diego, CA, USA, 7-9 May 2015.

[2] C. Szegedy, *et al*., "Intriguing properties of neural networks," in *Proc. 2nd Int. Conf. on Learning Representations, ICLR '14,* 15 pp., Banff, Canada, 14-16 Apr. 2014.

[3] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Trans. on Evolutionary Computation*, vol. 23, no. 5, pp. 828-841, Oct. 2019.

[4] N. Martins, J. M. Cruz, T. Cruz, and P. Henriques Abreu, "Adversarial machine learning applied to intrusion and malware scenarios: a systematic review," *IEEE Access*, vol. 8, pp. 35403-35419, 2020.

[5] X. Peng, H. Xian, Q. Lu, and X. Lu, "Semantics aware adversarial malware examples generation for black-box attacks," *Applied Soft Computing*, vol. 109, Article ID: 107506, Sept. 2021.

[6] Y. Y. Chen, C. T. Chen, C. Y. Sang, Y. C. Yang, and S. H. Huang, "Adversarial attacks against reinforcement learning-based portfolio management strategy," *IEEE Access*, vol. 9, pp. 50667-50685, 2021.

[7] R. Ramadan, "Detecting adversarial attacks on audio-visual speech recognition using deep learning method," *International J. of Speech Technology*, Article ID: 02.06.2021, 21 pp., Jun. 2021.

[8] L. Yang, Q. Song, and Y. Wu, "Attacks on state-of-the-art face recognition using attentional adversarial attack generative network," *Multimedia Tools and Applications*, vol. 80, pp. 1-21, 2021.

[9] Y. Zuo, H. Yao, and C. Xu, "Category-level adversarial self-ensembling for domain adaptation," in *Proc. IEEE Int. Conf. on Multimedia and Expo, ICME'20*, 6 pp., London, UK, 6-10 Jul. 2020.

[10] Z. Wei, *et al*., "Heuristic black-box adversarial attacks on video recognition models," in *Proc. of the 34th AAAI Conf. on Artificial Intelligence*, pp. 12338-12345, New York, NY, USA, 7-12 Feb. 2020.

[11] D. Wang, *et al*., "Daedalus: breaking nonmaximum suppression in object detection via adversarial examples," *IEEE Trans. on Cybernetics*, Early Acces, pp. 1-14, 2021.

[12] I. Goodfellow, *et al*, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., pp. 2672-2680, 2014.

[13] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *Proc. of the 32nd Int. Conf. on Machine Learning*, vol. 37, pp. 1613-1622, Lille, France, Jul. 2015.

[14] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," *Advances in Neural Information Processing Systems, NIPS'08*, pp. 1177-1184, Vancouver and Whister, Canada, 3-6 Dec. 2008.

[15] K. Cutajar, E. V. Bonilla, P. Michiardi, and M. Filippone, "Random feature expansions for deep Gaussian processes," in *Proc. of the 34th Int. Conf. on Machine Learning,* , pp. 884-893, Sydney, Australia, Aug. 2017.

[16] A. Rozsa, E. M. Rudd, and T. E. Boult, "Adversarial diversity and hard positive generation," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, *CVPR'16*, pp. 25-32, Las Vegas, NV, USA, 27-30 Jun. 2016.

[17] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, *CVPR'18*, pp. 9185-9193, Salt Lake City, UT, USA, 18-23 Jun. 2018.

[18] F. Tramer, *et al*., "Ensemble adversarial training: attacks and defenses," in *Proc. 6th Int. Conf. on Learning Representations*, *ICLR'18*, 20 pp., Vancouver, Canada, 30 Apr.-3 May 2018.

[19] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Proc. 5th Int. Conf. on Learning Representations, ICLR'17*, 15 pp., Toulon, France, 24-26 Apr. 2017.

[20] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, *CVPR'16*,, pp. 2574-2582, Las Vegas, NV, USA, 27-30 Jun. 2016.

[21] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: attacks and defenses for deep learning," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2805-2824, Sept. 2019.

[22] S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, *CVPR'17*, pp. 1765-1773, Honolulu, HI, USA, 21-26 Jul. 2017.

[23] C. Zhang, P. Benz, T. Imtiaz, and I. S. Kweon, "Understanding adversarial examples from the mutual influence of images and perturbations," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition, CVPR'20*, pp. 14509-14518, Seattle, WA, USA, 13-19 Jun. 2020.

[24] A. Demontis, M. Melis, M. Pintor, M. Jagielski, B. Biggio, A. Oprea, C. Nita-Rotaru, and F. Roli, "Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks," in *Proc. 28th USENIX Security Symp., USENIX Security*, pp. 321-338, Santa Clara, CA,USA, 14-16 Aug. 2019.

[25] N. Carlini and D. A. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. on Security and Privacy,* pp. 39-57, San Jose, CA, USA, 22-26 May 2017.

[26] J. Rony, L. G. Hafemann, L. S. Oliveira, I. B. Ayed, R. Sabourin, and E. Granger, "Decoupling direction and norm for efficient gradient-based L2 adversarial attacks and defenses," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR'19*, pp. 4322-4330, , Long Beach, CA, USA, Jun. 2019.

[27] Y. Liu and F. Cao, "Self-adaptive norm update for faster gradient based L2 adversarial attacks and defenses," in *Proc. of the 10th Int. Conf. on Pattern Recognition Applications and Methods*, *ICPRAM'21*, vol. 1, pp. 15-24, Vienna, Austria, 4-6 Feb. 2021.

[28] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. on Security and Privacy*, pp. 582-597, San Jose, CA,USA, 22-26 May 2016.

[29] J. Bradshaw, A. G. d. G. Matthews, and Z. Ghahramani, *Adversarial Examples, Uncertainty, and Transfer Testing Robustness in Gaussian Process Hybrid Deep Networks*, arXiv preprint arXiv:1707.02476, 2017.

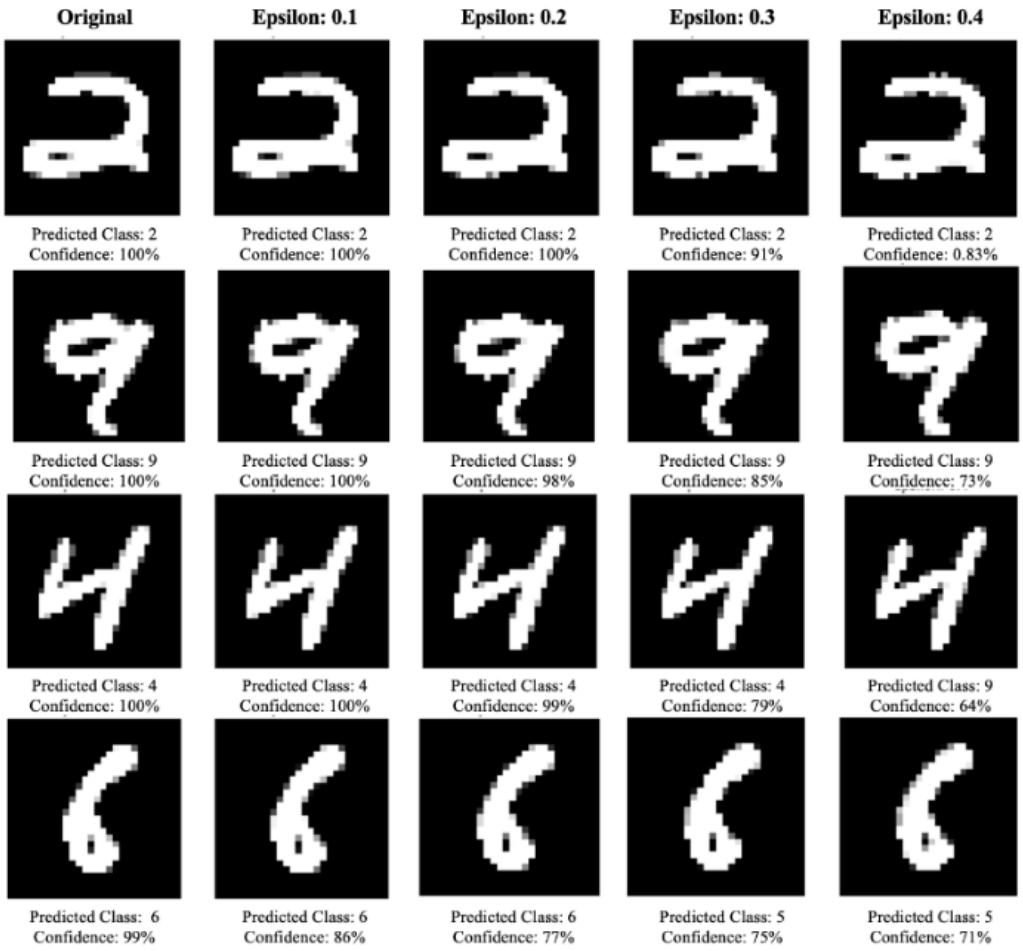
[30] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," in *Proc. 5th Int. Conf. on Learning Representations, ICLR'17,* 12 pp., Toulon, France, Apr. 2017.

[31] D. Hendrycks and K. Gimpel, "Early methods for detecting adversarial images," in *Proc. 5th Int. Conf. on Learning Representations, Workshop Track, ICLR'17*, 9 pp., Toulon, France, Apr. 2017.

[32] J. Wei, *Adversarial Examples for Visual Decompilers*, Master's Thesis, EECS Department, University of California, Berkeley, May 2017.

[33] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le, "Adversarial examples improve image recognition," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition, CVPR'20,* pp. 816-825, Seattle, WA, USA, 13-19 Jun. 2020.

[34]



شکل ۵: نمونه‌هایی از تصاویر تخاصمی با اختلال‌های مختلف و نتیجه پیش‌بینی مدل B-GPRF به همراه درصد اطمینان.

[35] H. Zheng, Z. Zhang, J. Gu, H. Lee, and A. Prakash, "Efficient adversarial training with transferable adversarial examples," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition, CVPR'20,* pp. 1178-1187, Seattle, WA, USA, 13-19 Jun. 2020.

[36] F. Guo, *et al*., "Detecting adversarial examples via prediction difference for deep neural networks," *Information Sciences*, vol. 501, pp. 182-192, Oct. 2019.

[37] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: beyond empirical risk minimization," in *Proc. 6th Int. Conf. on Learning Representations, ICLR'18,* 13 pp., Vancouver, Canada, 30 Apr.-3 May 2018.

[38] P. Pauli, A. Koch, J. Berberich, P. Kohler, and F. Allgower, "Training robust neural networks using lipschitz bounds," *IEEE Control. Syst. Lett.*, vol. 6, pp. 121-126, 2021.

[39] A. Graves, "Practical variational inference for neural networks," in *Proc. 25th Annual Conf. on Neural Information Processing Systems*, *NIPS'11,* pp. 2348-2356, Sierra Nevada, Spain, 16-17 Dec. 2011.

[40] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[41] J. Quinonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate gaussian process regression," *J. of Machine Learning Research*, vol. 6, pp. 1939-1959, Dec. 2005.

[42] V. Tresp, "A bayesian committee machine," *Neural Computation*, vol. 12, no. 11, pp. 2719-2741, Nov. 2000.

[43] T. Chen and J. Ren, "Bagging for gaussian process regression," *Neurocomputing*, vol. 72, no. 7-9, pp. 1605-1610, Mar. 2009.

[44] E. Rodner, A. Freytag, P. Bodesheim, and J. Denzler, "Large-scale gaussian process classification with flexible adaptive histogram kernels," in *Proc. European Conf. on Computer Vision, ECCV'12*, pp. 85-98, Florence, Italy, 7-13 Oct 2012.

[45] M. Sło´nski, "Bayesian neural networks and gaussian processes in identification of concrete properties," *Computer Assisted Methods in Engineering and Science*, vol. 18, no. 4, pp. 291-302, 2017.

[46] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*, MIT Press Cambridge, MA, 2006.

[47] A. Mustafa, *et al*., "Adversarial defense by restricting the hidden space of deep neural networks," in *Proc. IEEE/CVF Int. Conf. on Computer Vision, ICCV'19,* pp. 3384-3393, Seoul, South Korea, 27 Oct.-2 Nov. 2019.

[48] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[49] H. Khosravi and E. Kabir, "Introducing a very large dataset of handwritten farsi digits and a study on their varieties," *Pattern Recognit. Lett.*, vol. 28, no. 10, pp. 1133-1141, 2007.

مهران صفایانی تحصیلات خود را در مقطع کارشناسی در رشته مهندسی کامپیوتر از دانشگاه اصفهان در سال ۱۳۸۱ به پایان رساند. سپس مدارک کارشناسی ارشد و دکتری را به ترتیب در رشته مهندسی کامپیوتر گرایش معماری کامپیوتر و هوش مصنوعی در سال‌های ۱۳۸۵ و ۱۳۹۰ از دانشگاه صنعتی شریف دریافت کرد. از سال ۱۳۹۱ او به عنوان عضو هیأت علمی در دانشکده مهندسی برق و کامپیوتر دانشگاه صنعتی اصفهان بوده است. علاقه‌مندی‌های تحقیقاتی او شامل یادگیری ماشین، یادگیری عمیق، شناسایی الگو و محاسبات نرم است.

پویان شالبافان مقاطع کارشناسی و کارشناسی ارشد را در رشته مهندسی کامپیوتر به ترتیب در دانشگاه گیلان و دانشگاه صنعتی اصفهان در سال‌های ۱۳۹۴ و ۱۳۹۸ به پایان رساند. هم‌اکنون او در شرکتی خصوصی مشغول به کار است. علاقه‌مندی‌های تحقیقاتی او شامل یادگیری ماشین و یادگیری عمیق است.

سید هاشم احمدی مدارک کارشناسی و کارشناسی ارشد خود را در رشته مهندسی برق از دانشگاه صنعتی اصفهان در سال‌های ۱۳۸۰ و ۱۳۸۳ دریافت کرد. علاقه‌مندی‌های تحقیقاتی او شامل مدل‌های احتمالاتی و یادگیری ماشین است.

مهدیه فلاح علی‌آبادی مقاطع کارشناسی و کارشناسی ارشد را در رشته مهندسی کامپیوتر به ترتیب در دانشگاه‌های صنعتی اصفهان و یزد در سال‌های ۱۳۹۲ و ۱۳۹۵ گذرانده است. او هم‌اکنون دانشجوی دکتری رشته مهندسی کامپیوتر در دانشگاه صنعتی اصفهان است. علاقه‌مندی‌های تحقیقاتی او شامل یادگیری عمیق و تئوری گراف است.

عبدالرضا میرزایی در اصفهان متولد شده است. او مدرک کارشناسی‌اش را در رشته مهندسی کامپیوتر از دانشگاه اصفهان در سال ۱۳۸۰ اخذ کرد. سپس مقاطع کارشناسی ارشد و دکتری خود را در گرایش هوش مصنوعی از دانشگاه علم و صنعت ایران و دانشگاه امیرکبیر در سال‌های ۱۳۸۲ و ۱۳۸۸ به پایان رساند. او هم‌اکنون در دانشکده برق و کامپیوتر دانشگاه صنعتی اصفهان به عنوان عضو هیأت علمی حضور دارد. علاقه‌مندی‌های تحقیقاتی او شامل روش‌های شناسایی آماری و ساختاری الگو، پردازش تصاویر، بینایی کامپیوتر و یادگیری ماشین است.