Journal of Applied And Basic Machine Intelligence Research (ABMIR)

پژوهش‌های نظری و کاربردی هوش ماشینی

سال چهارم، شماره دوم، پاییز و زمستان ۱۴۰۵، صفحات ۲۹ - ۱۷

سپر تفکیک‌پذیر (GRDS)، ارائه روشی برای افزایش مقاومت شبکه‌های عصبی گرافی

علی حسین پورنادری[۱]، محمدعلی جوادزاده[۲*]، حسین حسینی[۳]

[۱] دانشجوی کارشناسی ارشد هوش مصنوعی و رباتیکز، دانشکده هوش مصنوعی و علوم شناختی، دانشگاه جامع امام حسین (ع)، تهران، ایران

[۲] استادیار، دانشکده هوش مصنوعی و علوم شناختی، دانشگاه جامع امام حسین (ع)، تهران، ایران

[۳] دانشجوی دکتری هوش مصنوعی و رباتیکز، دانشکده هوش مصنوعی و علوم شناختی، دانشگاه جامع امام حسین (ع)، تهران، ایران

مقاله پژوهشی

تاریخ دریافت:
۱۴۰۴/۱۱/۰۹

تاریخ پذیرش:
۱۴۰۵/۰۲/۱۴

کلیدواژه‌ها:
شبکه عصبی گرافی، مقاومت در برابر حملات تخاصمی، تصادفی‌سازی تفاضل‌پذیر، تابع زیان چند-هدفه، سپر گرافی ماژولار

نویسنده‌مسئول:
javadzade@ihu.ac.ir

چکیده

آسیب‌پذیری شبکه عصبی گرافی (GNN) در مقابل حملات تخاصمی یک چالش اساسی در این حوزه بوده که پیاده‌سازی آن‌ها را در کاربردهای حساس و با ریسک بالا محدود می‌سازد. در این پژوهش، یک چارچوب دفاعی هوشمند با نام تصادفی‌سازی گراف به‌عنوان سپر تفکیک‌پذیر (GRDS) را معرفی می‌کنیم که این چالش را با هزینه محاسباتی معقول برطرف می‌کند. نوآوری اصلی پژوهش ما در ارائه یک راهکار عملی بوده که برخلاف روش‌های مرسوم، مقاومت را بدون کاهش چشمگیر دقت در داده‌های تمیز، ایجاد می‌کند. روشی که در این چارچوب به کار گرفته‌ایم برپایه یک سپر ماژولار بنا شده است که با استفاده از یک تابع زیان چند-هدفه و هوشمند، با یادگیری می‌تواند به‌صورت تفکیک‌پذیر باعث تمایز یال‌های حیاتی و غیرحیاتی گراف شود. این سپر قبل از شبکه عصبی گرافی قرارگرفته و با تار کردن هدفمند ساختار گراف، باعث گمراهی مهاجمان می‌گردد. این چارچوب با Pytorch و کتابخانه PyTorch Geometric پیاده‌سازی شده است، ارزیابی جامع بر روی دیتاست‌های استاندارد Cora، Pubmed و CiteSeer و در برابر انواع روش‌های حمله (مبتنی بر گرادیان، تصادفی و مانند آن) انجام شد و نتایج استخراج‌شده نشان می‌دهد GRDS با هزینه‌ای ناچیز در دقت (افت کمتر از ٤٪) باعث افزایش مقاومت مدل نسبت به مدل پایه می‌شود (بیشتر از ۱۱٪). این یافته الهام‌بخش آن بوده که دفاع مؤثر در تصادفی سازی هوشمند بهتر از حذف کورکورانه عمل می‌کند.

: 10.22034/ABMIR.2026.24239.1220

E-ISSN: 2821-2037

۱- مقدمه

شبکه‌های عصبی گرافی (GNNs) با ارائه چارچوبی قدرتمند برای یادگیری از داده‌هایی که دارای ساختار رابطه‌ای هستند، به ابزاری کلیدی در هوش مصنوعی مدرن تبدیل شده‌اند. این معماری‌ها، با استفاده از مکانیزم انتشار پیام بین گره‌های همسایه، توانایی منحصربه‌فردی در استخراج الگوهای پیچیده از گراف‌ها، از خود نشان داده‌اند. موفقیت چشمگیرGNN ها در حوزه‌های متنوعی همچون تحلیل شبکه‌های اجتماعی [۱]، کشف دارو [۲] و سیستم‌های توصیه‌گر[۳]و طبقه‌بندی گره‌ها در گراف‌های استنادی[۱۲] و مدل‌سازی سیستم‌های گوناگون برپایه گراف گواهی براین ادعاست. درکنار این موفقیت‌ها شبکه‌های عصبی گرافی از چالش‌های اساسی آن‌ها می‌توان به آسیب‌پذیری شدید در برابر حملات تخاصمی اشاره کرد.

مطالعات صورت گرفته برای حل این چالش مانند Nettack [۱]، نشان می‌دهد که یک مهاجم می‌تواند با ایجاد تغییرات بسیار جزئی و تقریباً نامرئی در ساختار گراف (حذف یا اضافه یا حتی ویرایش چند یال) یا ویژگی‌های گره‌ها، پیش‌بینی مدل را به‌طور کامل به‌هم‌ریخته و باعث تصمیمات اشتباه شود. این چالش باعث شده است که استفاده GNN ها در حوزه‌های با ریسک بالا- مانند تشخیص کلاهبرداری مالی یا شبکه‌های زیرساختی- با تردیدهای جدی روبه‌رو شود.

برای رفع این چالش پژوهش‌هایی که در این حوزه صورت گرفته‌اند راهکارهای خود را در سه دسته پیشنهاد کرده‌اند. دسته اول، آموزش تخاصمی (Training Adversarial) بوده که با الهام‌گیری از کارهای مشابه در حوزه بینایی ماشین [۵] تلاش می‌کنند با اضافه کردن نمونه‌های حمله‌شده به داده‌های آموزشی، مقاومت مدل نیز بهبود پیدا کند، این روش مؤثر بوده اما بار محاسباتی آن نیز بسیار بالا بوده و فرایند آموزش را چندین برابر می‌کند.

دسته دوم، معماری‌های مقاوم[1] بوده که با تغییر در داخل معماری GNN در تلاش هستند تا مقاومت مدل را در برابر حملات تخاصمی بهبود بدهند، مثلاً با استفاده از لایه‌های توجه برای وزن‌دهی به همسایه‌ها (مانند مدل GAT [۶]) و یا با استفاده از روش‌های آماری مقاوم [۷] سعی برکاهش تأثیر اختلالات دارند. خود این روش‌ها نیز احتمال بیش هموارسازی[2] را بیشتر می‌کند که خود باعث از دست دادن اطلاعات مفید گراف می‌شود.

دسته سوم، دسته‌ای از پژوهش‌ها پیش‌پردازش و پاک‌سازی گراف[3] را ارائه کرده و تلاش می‌کنند پیش از ورود گراف به مدل یال‌های مشکوک یا ناهنجار را شناسایی و حذف کنند [۸]. عدم تطبیق‌پذیری و ناتوانی در مقابله با حملات هوشمند و هدف‌مند که خود را شبیه ساختار عادی گراف نشان می‌دهند را می‌توان از ضعف‌های اصلی این روش بیان کرد.

در این مقاله، ما برای حل این چالش، ایده تصادفی‌سازی هوشمند را در قالب یک سپر دفاعی ماژولار و تفکیک‌پذیر (GRDS) معرفی می‌کنیم. آزمایش‌های اولیه نشان داد که یک تصادفی‌سازی ساده و تهاجمی، هرچند گاهی مقاومت را افزایش می‌دهد، اما به قیمت از دست دادن اطلاعات مفید و افت شدید دقت تمام می‌شود. این مشاهدات، ما را به سمت نوآوری اصلی این پژوهش که همان طراحی تابع زیان چندهدفه و هوشمند که به سپر دفاعی، قوه تشخیص اضافه می‌کرد هدایت کرد. این سپر یاد می‌گیرد که به‌طور هم‌زمان سه هدف (۱-محافظت از یال‌های حیاتی که اطلاعات معنایی مهم را منتقل می‌کنند. ۲-حفظ ساختار کلی گراف برای جلوگیری از افت دقت ۳-افزایش قاطعیت برای حذف یا نگهداری یال‌ها) را دنبال کند. این رویکرد به مدل اجازه می‌دهد که به یک تعادل بهینه بین مقاومت و دقت برسد.

هدف نهایی این پژوهش که می‌توان گفت ترکیبی از رویکردهای دسته اول و دوم است، ارائه یک چارچوب نوین منظم‌سازی چندهدفه برای آموزش سپرهای دفاعی GNN و طراحی یک ماژول دفاعی کاملاً تفکیک‌پذیر و ماژولار (GRDS[4]) که به‌راحتی برروی هر شبکه عصبی گرافی استاندارد (مانند GCN) قابل اضافه شدن باشد.

[1] Robust Architectures
[2] Over Smoothing
[3] Graph Preprocessing
[4] Graph Randomization As A Differentiable Shield

اجرای آزمایش‌های جامع که نشان می‌دهد GRDS با هزینه‌ای ناچیز در دقت (افت کمتر از ٤٪)، مقاومت در برابر انواع حملات را به شکل چشمگیری (بیش از ۱۱٪) افزایش می‌دهد. ارائه تحلیل‌های عمیق از رفتارهای سپر نشان می‌دهد مدل ما به‌صورت هوشمندانه‌ای رفتار خود را برای محافظت از ساختار معنایی گراف تطبیق می‌دهد.

دستاوردهای ما در این پژوهش را می‌توان در سه محور به‌صورت زیر خلاصه کرد: معرفی یک چارچوب نوین با نام GRDS، که یک سپر دفاعی ماژولار و تفکیک‌پذیر برای شبکه‌های عصبی گرافی است؛ طراحی یک تابع زیان چندهدفه هوشمند که برخلاف تصادفی‌سازی‌های مخرب، به‌طور هم‌زمان سه هدف حفاظت از یال‌های حیاتی، حفظ ساختار کلی گراف و افزایش قاطعیت در تشخیص یال‌های مخرب را دنبال می‌کند؛ و درنهایت، ارائه رویکردی که به مدل اجازه می‌دهد تا تعادلی بهینه بین مقاومت و دقت برقرار سازد. نتایج به‌دست‌آمده نشان می‌دهد این چارچوب با افت دقت کمتر از ٤ درصد در زمان حملات تخاصمی، موجب افزایش مقاومت بیش از ۱۱ درصدی می‌شود.

در ادامه مقاله در بخش ۲ به پیشینه تحقیق یا کارهای مرتبط که درراستای این پژوهش انجام‌شده پرداخته می‌شود، سپس در بخش ۳ به ارائه روش پیشنهادی GRDS پرداخته‌شده و در بخش ٤ نیز شرح آزمایش‌ها و پیاده‌سازی آن‌ها توضیح داده‌شده و در بخش ۵ نتایج و تحلیل‌های خروجی از بخش قبلی ارائه‌شده و بخش ٦ نیز به نتیجه‌گیری اختصاص دارد.

۲- کارهای مرتبط

در سال‌های اخیر افزایش آسیب‌پذیری GNN ها در برآر حملات تخاصمی، منجر به پژوهش‌های زیادی در مورد روش‌های دفاعی شده است که به سه دسته اصلی شامل آموزش تخاصمی، معماری‌های مقاوم، و پیش‌پردازش و پاک‌سازی گراف طبقه‌بندی می‌شود. در ادامه، چندین پژوهش را در مورد این دسته‌ها را با مرور انتقادی بررسی و انگیزه اصلی توسعه چارچوب پیشنهادی خود (GRDS) را بیان خواهیم کرد.

۲-۱ آموزش تخاصمی

ایده اصلی آموزش تخاصمی، فرموله‌شدن یادگیری به‌صورت یک بازی کمینه بیشینه (min-max) میان مدل یادگیرنده و یک مهاجم است. در این چارچوب، حلقه درونی به دنبال تولید بدترین اغتشاش ممکن روی داده ورودی (ساختار یا ویژگی‌های گراف) با هدف بیشینه شدن خطای مدل است، درحالی‌که حلقه بیرونی پارامترهای GNN را به‌گونه‌ای به‌روزرسانی می‌کند که عملکرد آن در برابر این اغتشاشات بهینه شود. این ایده ابتدا در حوزه بینایی ماشین و توسط Madry و همکاران [۵] به‌عنوان یکی از قوی‌ترین دفاع‌ها در برابر حملات تخاصمی معرفی شد. در حوزه گراف، Zügner و همکارانش [۱] از نخستین پژوهشگرانی بودند که این روش را برای GNN ها به‌کار گرفتند و نشان دادند که می‌توان با در نظر گرفتن اغتشاشات ساختاری و ویژگی‌محور، مدل‌هایی مقاوم‌تر آموزش داد. با وجود اثربخشی مفهومی، مهم‌ترین محدودیت آموزش تخاصمی در گراف‌ها، هزینه محاسباتی بسیار بالای آن است. به‌طور مشخص، حل مسئله بهینه‌سازی درونی که اغلب نیازمند چندین تکرار گرادیانی روی فضای گسسته گراف است، مقیاس‌پذیری این روش را به‌شدت محدود می‌کند و آن را برای گراف‌های بزرگ و دنیای واقعی عملاً غیرقابل استفاده می‌سازد.

۲-۲ معماری‌های مقاوم

دسته دوم از روش‌ها، به‌جای تغییر فرایند آموزش، مستقیماً معماری GNN را به‌گونه‌ای طراحی یا اصلاح می‌کنند که ذاتاً در برابر اغتشاشات مقاوم باشد. ایده اصلی در این رویکرد، کاهش تأثیر همسایگان نامعتبر یا نویزی در فرایند تجمیع پیام است. به‌عنوان مثال، شبکه‌های توجه گرافی[1] یا GAT [٦] با یادگیری وزن‌های توجه، به‌طور ضمنی اهمیت همسایگان مختلف را تنظیم کرده و می‌توانند اثر یال‌های کم‌اهمیت یا مخرب را کاهش دهند. در همین راستا، روش‌هایی مانند RobustGCN [۷] با بهره‌گیری از تجمیع‌گرهای آماری مقاوم، سعی در محدود کردن تأثیر نقاط پرت[2] دارند. همچنین، معماری‌های تخصصی‌تری نظیر GNNGuard [10] با افزودن مکانیزم‌های صریح برای تشخیص

[1] Graph Attention Networks

[2] Outliers

و تضعیف یال‌های مشکوک پیشنهاد شده‌اند؛ درحالی‌که نقطه‌ضعف مشترک این دسته از روش‌ها، خطر از دست رفتن اطلاعات و پدیده بیش‌هموارسازی است. ازآنجاکه این معماری‌ها اغلب به‌صورت محافظه‌کارانه طراحی می‌شوند، ممکن است نه‌تنها نویز تخاصمی، بلکه سیگنال‌های مفید و پرفرکانس گراف را نیز فیلتر کنند. این امر می‌تواند به افت عملکرد مدل در داده‌های پاک و بدون حمله منجر شود و یک مبادله نامطلوب میان مقاومت و دقت ایجاد کند.

۳-۲ پیش‌پردازش و پاک‌سازی گراف

سومین دسته از دفاع‌ها، به‌عنوان یک لایه فیلتر مستقل پیش از GNN عمل می‌کنند. این روش‌ها گراف ورودی را تحلیل کرده و با شناسایی ساختارهای مشکوک، اقدام به حذف یا اصلاح آن‌ها می‌کنند. نمونه‌های شاخص شامل روش‌هایی هستند که یال‌ها را بر اساس شباهت ویژگی‌های گره‌ها هرس می‌کنند؛ به‌طوری‌که یال‌های میان گره‌هایی با ویژگی‌های بسیار متفاوت حذف می‌شوند [۸]. همچنین، برخی رویکردها با استفاده از فیلترهای پایین‌گذر در حوزه طیفی گراف، مؤلفه‌های پرفرکانس و نویزی را که اغلب به حملات نسبت داده می‌شوند، تضعیف می‌کنند. چالش و محدودیت اساسی این روش‌ها، عدم انطباق‌پذیری آن‌هاست. این دفاع‌ها عموماً مستقل از مدل[1] هستند و هیچ آگاهی از معماری یا پارامترهای GNN مورد حفاظت ندارند. درنتیجه، در برابر حملات تطبیقی و پیشرفته‌ای که به‌طور خاص برای عبور از این فیلترهای عمومی طراحی شده‌اند، آسیب‌پذیر باقی می‌مانند.

درمجموع، آموزش تخاصمی علی‌رغم قدرت تئوریک، به دلیل هزینه محاسباتی بالا مقیاس‌پذیر نیست؛ معماری‌های مقاوم با خطر از دست دادن اطلاعات مفید و کاهش دقت روی داده‌های پاک مواجه‌اند؛ و روش‌های پیش‌پردازش، به سبب ماهیت غیرانطباقی و مدل ناآگاه خود، در برابر حملات تطبیقی ناکارآمد هستند. در مقابل، روش پیشنهادی ما، GRDS، با معرفی یک سپر تصادفی‌سازی قابل تفاضل که به‌صورت انتهابه‌انتها همراه با GNN آموزش می‌بیند، این محدودیت‌ها را برطرف می‌کند. GRDS از نظر محاسباتی به‌مراتب کارآمدتر از آموزش تخاصمی است، زیرا نیازمند حل یک مسئله بهینه‌سازی درونی پرهزینه نیست؛ در مقایسه با معماری‌های مقاوم، با یادگیری این‌که چه چیزی و تا چه حد باید تصادفی‌سازی شود، دقت روی داده‌های پاک را حفظ می‌کند؛ و برخلاف روش‌های پیش‌پردازش، به دلیل آموزش مشترک با مدل هدف، یک دفاع انطباقی و مدل‌آگاه را فراهم می‌آورد. این ویژگی‌ها، GRDS را به‌عنوان یک راهکار جامع و برتر در چشم‌انداز دفاع‌های تخاصمی برای GNN ها مطرح می‌سازد.

۳- روش پیشنهادی (GRDS)

در این بخش، ما در پاسخ به چالش‌های مطرح‌شده، چارچوب "تصادفی‌سازی گراف به‌عنوان سپر تفکیک‌پذیر" (GRDS) را به‌عنوان یک راه حل نوین ارائه می‌دهیم. ایده اصلی GRDS، معرفی یک لایه دفاعی ماژولار و هوشمند، موسوم به "سپر" (S_φ)، است که به‌صورت پویا، ساختار گراف را برای افزایش مقاومت مدل‌های شبکه‌های عصبی گرافی (GNN) در برابر حملات ساختاری بهینه می‌کند و درشرایط بدون حمله نیز افت دقت شبکه‌های عصبی گراف را به حداقل می‌رساند. وظیفه اصلی این سپر، تبدیل گراف ورودی به یک نسخه دستکاری‌شده و تصادفی ($\tilde{G}$) به گونه‌ای است که از یک سو، الگوهای ساختاری که مهاجم برای طراحی حمله به آن‌ها تکیه می‌کند را "تار" و مبهم سازد و از سوی دیگر، ساختار اطلاعاتی ضروری برای انجام وظیفه اصلی (مانند طبقه‌بندی گره) را تا حد امکان حفظ نماید. معماری کلی این چارچوب تعاملی در شکل (۱) به تصویر کشیده شده است.

این چارچوب از سه جزء اصلی تشکیل‌شده است که در ادامه به تفصیل شرح داده شده است: یک مکانیزم تصادفی‌سازی تفکیک‌پذیر برای دستکاری ساختار گراف، یک فرآیند بهینه‌سازی دو سطحی برای آموزش مدل در یک محیط تخاصمی، و یک تابع زیان چند-هدفه پیشرفته که رفتار سپر را هوشمندانه هدایت می‌کند.

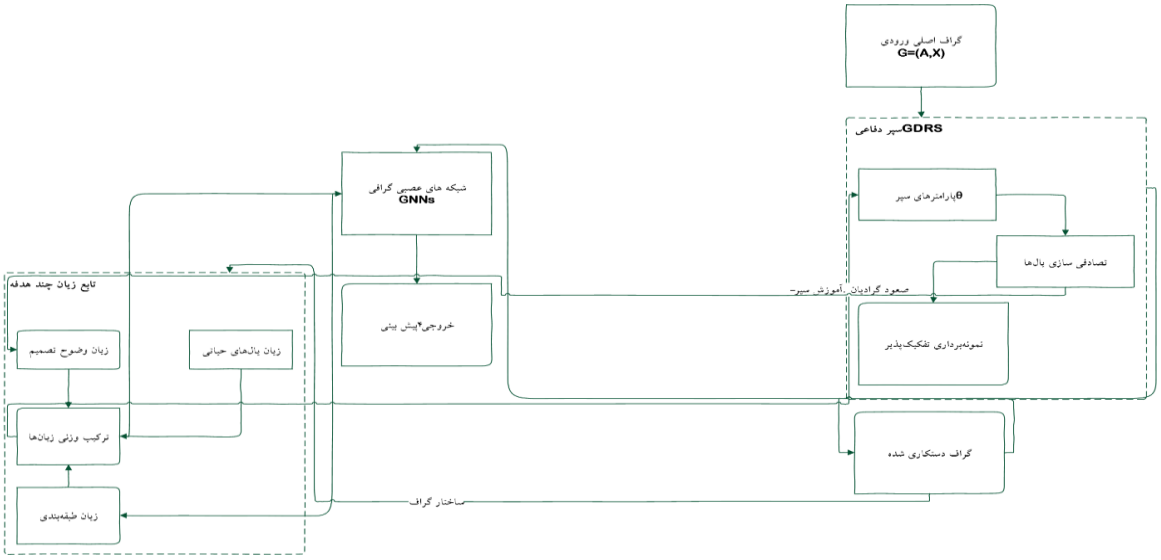
۳- ۱ سپر GRDS: یک ماژول یادگیرنده

هسته اصلی چارچوب معرفی شده، ماژول GRDShield بوده که موجب توانایی سپر در تصمیم‌گیری احتمالی برای نگه‌داشتن یا حذف هر یال به صورت تفکیک‌پذیر[2] می‌شود. این ویژگی حیاتی،

[1] Model Agnostic

[2] Differentiable

به ما اجازه می‌دهد که پارامترهای این فرآیند تصادفی را، نه به صورت دستی، بلکه از طریق الگوریتم پس‌انتشار خطا[1] و به صورت سرتاسری[2] همراه با خود GNN، بهینه کنیم. براساس فرمول ۱ برای هر یال (u,v)درگراف، سپر یک پارامتر یادگیرنده θ_{uv}را در نظر گرفته که باعث کنترل شدن احتمال حذف هر یال می‌شوند.



شکل (۱): معماری کلی چارچوب GRDS

در بازه (۰, ۱) قرار می‌گیرند که بیانگر احتمال عدم حذف آن یال است

$$(p_{uv} = \sigma(\alpha_{uv})) \quad (۱)$$

۳-۲ نمونه‌برداری تفکیک‌پذیر از ساختار گراف

یکی از چالش‌های اصلی در یادگیری ساختار گراف، ماهیت گسسته (حضور یا عدم حضور) یال‌ها است که فرآیند بهینه‌سازی مبتنی بر گرادیان را غیرممکن می‌سازد. برای غلبه بر این مشکل، از ترفند Gumbel-Softmax بهره می‌بریم. این تکنیک امکان نمونه‌برداری تفکیک‌پذیر از یک متغیر تصادفی دودویی را فراهم می‌کند.

احتمال حفظ p_{uv} برای هر یال (u, v) ابتدا با استفاده از یک تابع سیگموئید بر اساس پارامترهای یادگیرنده محاسبه می‌شود. سپس، یک ماسک نرم m_{uv} که مقداری در بازه [0,1] دارد، به صورت فرمول ۲ نمونه‌برداری می‌شود:

$$m_{uv} = \text{Softmax}\left(\frac{[\log(p_{uv}) + g_1,\ \log(1-p_{uv}) + g_0]}{\tau}\right) \quad (۲)$$

که در آن g_0 و g_1 نمونه‌های مستقل از توضیع *Gumbel* (۰،۱) بوده و τ پارامتر دما[3] است که میزان نرمی یا تخمین‌پذیری، نمونه‌برداری را کنترل می‌کند. با کاهش τ در طول آموزش، نمونه‌برداری به سمت گسسته‌سازی میل می‌کند.

در طول آموزش این ماسک[4] به‌عنوان وزن یال‌ها عمل کرده و به گرادیان اجازه انتشار می‌دهد.

۳-۳ تابع زیان چندهدفه برای آموزش هوشمندانه

یکی دیگر از نوآوری‌های این چارچوب در تابع زیانی بوده که رفتار سپر را هدایت می‌کند. تابع زیان چند هدفه همانگونه که در فرمول ۳ دیده‌می‌شود؛ یک ترکیب وزن‌دار از چهار جمله‌ای است که هریک هدفی مشخص را دنبال می‌کنند.

$$L_{total} = L_{class} + \omega_{imp} \cdot L_{imp} + \omega_{other} \cdot L_{other} + \omega_{clarity} \cdot L_{clarity} \quad (۳)$$

[1] Backpropagation
[2] End To End
[3] Temperature
[4] Mask

اجزای فرمول ۳ را تشریح می‌کنیم.

۱-زیان طبقه‌بندی (L_{class}): این جمله، زیان استاندارد آنتروپی متقاطع روی گره‌های آموزشی است و هدف آن بیشینه‌سازی دقت پیش‌بینی مدل است.

زیان طبقه‌بندی به‌صورت فرمول ٤ محاسبه می‌شود که در آن $\hat{y}_i$ و y_i به ترتیب برچسب‌های واقعی و بردارهای پیشبینی احتمال کلاس برای گره i هستند.

$$L_{class} = -\Sigma_{(i\in V_{train})} y_i \cdot log(\hat{y}_i) \quad (٤)$$

۲-زیان یال‌های مهم (L_{imp}): بااستفاده از فرمول ۵، این جمله، سپر را برای حذف «یال‌های مهم» جریمه می‌کند. یال‌های بین دو گره از یک کلاس در مجموعه آموزش را به‌عنوان یال مهم تعریف می‌کنیم. این زیان، سپر را به حفظ ساختار معنایی[1] گراف تشویق می‌کند.

$$L_{imp} = \left(\frac{1}{|E_{imp}|}\right) \cdot \Sigma_{(u,v)\in E_{imp}} (1 - p_{uv}) \quad (٥)$$

که $E_imp = (u,v) \in y | E_u = y_v$ مجموعه یال‌های مهم است.

۳-زیان سایر یال‌ها (L_{other}): یک جریمه کوچک برای حذف هر یال دیگر، به‌صورتی‌که در فرمول ٦ مشاهده می‌شود، در نظر می‌گیرد تا از حذف بی‌رویه یال‌ها و ازهم‌گسیختگی گراف جلوگیری کند. این کار به حفظ پایداری کلی ساختار کمک می‌کند.

$$L_{other} = ReLU\left(\rho_{target} - \left(\frac{1}{|E|}\right)\Sigma_{(u,v)\in E} p_{uv}\right) \quad (٦)$$

در این تابع ρ_{target} نقش یک آستانه هدف برای نرخ میانگین حفظ یال‌ها در کل گراف را دارند.

٤-زیان شفافیت تصمیم ($L_{clarity}$): این جمله با جریمه کردن احتمالات p_{uv} که به ۰/۵ نزدیک هستند، سپر را به گرفتن تصمیمات قطعی (حفظ با احتمال نزدیک به ۱ یا حذف با احتمال نزدیک به ۰) هدایت می‌کند. این کار که‌در فرمول ۷ مشاهده می‌شود، عدم قطعیت را کاهش داده و به تفسیرپذیری مدل کمک می‌کند.

$$L_{clarity} = \left(\frac{1}{|E|}\right) \cdot \Sigma_{(u,v)\in E} [p_{uv} \cdot (1 - p_{uv})] \quad (٧)$$

[1] Semantic

٤-۳ بهینه‌سازی دو سطحی

برای اینکه سپر یاد بگیرد کدام یال‌ها را با چه احتمالی تغییر دهد تا به بهترین مقاومت ممکن برسد، ما از یک رویکرد الهام گرفته‌شده از آموزش کمینه، بیشینه[2]دو سطحی را استفاده می‌کنیم. این فرآیند، که فلوچارت آن در شکل (۲) نمایش داده شده، یک بازی استراتژیک بین "سپر" (که سعی در ایجاد چالش دارد) و "GNN" (که سعی در غلبه بر چالش دارد) ایجاد می‌کند.

حلقه داخلی (بیشینه‌سازی): در این مرحله که شامل K گام بهینه‌سازی است، پارامترهای مدل GNN ثابت بوده و پارامترهای سپر θ_{uv} سعی بر بیشینه‌کردن زیان طبقه‌بندی ($\mathcal{L}_{\text{class}}$)به‌روز می‌شوند و باعث صعود گرادیان می‌شوند. این فرآیند، سپر را وادار به ساخت بدترین ساختار ممکن برای فریب مدلGNN می‌کند و که موجب به‌چالش کشیدن چارچوب می‌شود.

حلقه خارجی (کمینه‌سازی): بعد از انجام حلقه داخلی، پارامترهای سپر ثابت شده و پارامترهای مدل GNN سعی بر کمینه ساختن تابع زیان کل $\mathcal{L}_{\text{total}}$(فرمول ۳) به‌روز می‌شوند (نزول گرادیان). این گام، مدل GNN را نسبت به ساختارهای تخاصمی که توسط سپر کشف شده‌اند، مقاوم می‌سازد.

این فرآیند تناوبی، یک حمله تخاصمی را شبیه‌سازی کرده تا درنهایت منجر به ساخت یک سپر دفاعی هوشمند و یک مدل GNNمقاوم در برابر حملات ساختاری شود.

۵-۳ تحلیل پیچیدگی محاسباتی

برای درک کارایی عملی چارچوبGRDS ، تحلیل پیچیدگی زمانی آن در مقایسه با مدل پایه GCN ضروری است. پیچیدگی محاسباتی GRDS از دو بخش تشکیل‌شده است: (۱) هزینه محاسباتی سپر GRDS برای به‌روزرسانی و نمونه‌برداری از ماسک یال‌ها، و (۲) هزینه محاسباتی GCN پایه.

برای یک گراف با |E| یال، |V|گره و F ویژگی، پیچیدگی هر لایه GCN از مرتبه $\mathcal{O}(|E|.F)$ است. در چارچوب GRDS، سپر نیازمند محاسبه پارامتر θ_{uv}برای هر یال (هزینه$\mathcal{O}(|E|)$) و اجرای نمونه‌برداری Gumbel-Softmax (هزینه$\mathcal{O}(|E|)$) است.

[2] Min Max

بنابراین، پیچیدگی کل GRDS در هر ایپاک به صورت زیر خواهد بود:

- **فاز آموزش**: با احتساب K گام بهینه‌سازی داخلی برای سپر، پیچیدگی کل $\mathcal{O}(K \cdot |E| + |E|.F)$ است. با انتخاب K=3 (مطابق جدول ۲)، سربار محاسباتی GRDS نسبت به GCN استاندارد حدود ۳ برابر هزینه پردازش یال‌ها است. با توجه به اینکه $|E|.F$ معمولاً بسیار بزرگ‌تر از |E| است (به‌ویژه در دیتاست‌های با ویژگی‌های بالا مانند Cora با ۱۴۳۳ ویژگی)، سربار عملی GRDS نسبتاً ناچیز است.
- **فاز تست**: در این فاز، نمونه‌برداری تصادفی غیرفعال شده و از ماسک قطعی (بر اساس آستانه‌گذاری روی p_{uv}) استفاده می‌شود. بنابراین پیچیدگی فاز تست GRDS دقیقاً برابر با GCN استاندارد ($\mathcal{O}(|E|F)$) است و هیچ سربار اضافه‌ای در زمان استنتاج ندارد .

۴- آزمایش‌ها (Experiments)

در این بخش، ما مجموعه‌ای از آزمایش‌ها را برای ارزیابی عملکرد چارچوب GRDS طراحی و اجرا می‌کنیم. هدف از انجام این آزمایش‌ها، پاسخ به سه سؤال پژوهشی کلیدی است:

۱-(مقاومت و دقت): آیا GRDS می‌تواند مقاومت[1] مدل GNN را در برابر حملات ساختاری با حداقل کاهش دقت افزایش دهد؟

۲-(رفتار هوشمندانه): آیا سپر دفاعی این چارچوب به‌طور هوشمندانه بین یال‌های مهم و غیرمهم تمایز قائل می‌شود؟

۳-(اهمیت اجزا): هر یک از اجزای تابع زیان چندهدفه ما (L_{imp}, L_{other}, $L_{clarity}$) چه تأثیری بر عملکرد نهایی دارند؟

۴-۱ مجموعه‌داده‌ها

ما همان‌گونه که در جدول ۱ دیده‌می‌شود آزمایش‌های خود را بر روی سه مجموعه داده استاندارد در حوزه شبکه‌های استنادی[2] انجام دادیم که به‌طور گسترده برای ارزیابی مدل‌های GNN استفاده می‌شوند [۱۱]: Cora، CiteSeer و PubMed. این دیتاست‌ها به دلیل داشتن ویژگی هموفیلی[3] بالا، یعنی گره‌های متصل به هم به‌احتمال زیاد دارای کلاس یکسانی هستند، برای ارزیابی L_{imp} بسیار مناسب‌اند.

جدول (۱): مجموعه داده‌های استفاده‌شده در آزمایش

دیتاست	گره‌ها	یال‌ها	کلاس‌ها	ویژگی‌ها
Cora	۲۷۰۸	۵۴۲۹	۷	۱۴۳۳
CiteSeer	۳۳۲۷	۴۷۳۲	۶	۳۷۰۳
PubMed	۱۹۷۱۷	۴۴۳۳۸	۳	۵۰۰

۴-۲ مدل‌های پایه

برای نشان دادن برتری چارچوب آن را در دو دسته مدل پایه آزمایش کردیم.

۱-GCN استاندارد (Vanilla GCN): یک مدل GCN دو لایه‌ای که هیچ‌گونه مکانیزم دفاعی ندارد. این مدل به‌عنوان حد پایین[4] عملکرد مقاومتی ما عمل می‌کند.

۲-GNNGuard : یکی از روش‌های پیشرفته state of the art که مخصوص دفاع در برابر حملات تخاصمی از شبکه‌های عصبی گرافی ساخته‌شده و با استفاده از مکانیزم توجه[5] سعی در شناسایی و کم‌اهمیت کردن یال‌های مخرب دارد.

۴-۳ معیارهای ارزیابی

برای ارزیابی از سه دسته معیار ارزیابی استفاده‌شده است که تشریح خواهد شد.

دقت در حالت پاک[6]: دقت طبقه‌بندی مدل روی گراف اصلی و دستکاری‌نشده بوده که هزینه یا افت عملکردی که مکانیزم دفاعی تحمیل می‌کند را نشان می‌دهد.

دقت در حالت مقاوم[7]: دقت طبقه‌بندی مدل پس از اعمال یک حمله ساختاری روی گراف بوده که از حملات متداول برای سنجش آن استفاده می‌شود.

شکاف حفاظتی[8]: این معیار اختصاصی که با فرمول ۸ برای این پژوهش تعریف‌شده، تفاوت میانگین احتمال حفظ یال‌های مهم و

[1] Robustness
[2] Citation Networks
[3] Homophily
[4] Lower Bound
[5] Attention
[6] Clean Accuracy
[7] Robust Accuracy
[8] Protection Gap

یال‌های غیرمهم را اندازه‌گیری کرده و نشان‌دهنده رفتار هوشمندانه سپر است.

$$ProtectionGap = mean(p_{important}) - mean(p_{nonimportant}) \quad (۸)$$

٤-٤ پیاده‌سازی

همه مدل‌ها با استفاده از کتابخانه PyTorch و PyTorch Geometric پیاده‌سازی شده و با python ۳/۱۰ اجرا شدند. مدل GNN پایه در همه آزمایش‌ها یک GCN دو لایه با ۳۲ واحد مخفی است. برای بهینه‌سازی از Adam استفاده شد. هایپرپارامترهای اصلی در جدول ۲ آورده شده‌اند که از طریق اعتبارسنجی[1] بر روی بخشی از داده‌های آموزشی انتخاب‌شده‌اند.

برای سناریوهای حملات تخاصمی از سه روش Gradient-based، Random Edge Flip و Nettack[٤] استفاده شد.

جدول (۱): هایپرپارامترهای اصلی استفاده‌شده در پیاده‌سازی

هایپرپارامتر	مقدار	توضیح
نرخ یادگیری (GNN)	۰/۰۱	Learning rate برای پارامترهای مدل GNN
نرخ یادگیری (سپر)	۰/۰۵	Learning rate برای grdsheild
وزن Wimp	۱/۵	وزن جریمه برای حذف یال‌های مهم
وزن Wother	۰/۵	وزن جریمه برای پایدار ماندن ساختار
وزن Wclarity	۰/۸	وزن جریمه برای شفافیت تصمیم
گام‌های حلقه داخلی k	۳	تعداد گام‌های بهینه‌سازی سپر در هر ایپاک
دمای t	۱/۰	پارامتر دما برای Gumbel-softmax
تعداد ایپاک‌ها	۲۰۰	تعداد کل دوره‌های آموزش

در روش Gradient-based که به Meta-attack نیز شناخته می‌شود. بهینه سازی به‌طوری انجام می‌شود که حداکثر یک درصد یال‌ها برپایه گرادیان تغییر کنند. در روش Random Edge Flip ۵ درصد یال‌ها به‌طور تصادفی حذف یا اضافه می‌شوند. در روش Nettack یال‌ها براساس ویژگی افزوده یا حذف می‌شوند. تمامی حملات با کتابخانه torchattacks نسخه ۲ پیاده‌سازی شدند.

۵- نتایج و تحلیل

در این بخش، نتایج حاصل از آزمایش‌های گسترده برای پاسخ به سه سؤال پژوهشی مطرح‌شده در بخش ٤ ارائه می‌شود. تمامی نتایج بر اساس ۵ بار اجرا با seed های تصادفی متفاوت و به صورت "میانگین ± انحراف معیار" گزارش‌شده‌اند تا از پایایی آماری یافته‌ها اطمینان حاصل شود. معیارهای ارزیابی شامل دقت (Accuracy)و F1-Score (ماکرو) برای تحلیل جامع‌تر عملکرد مدل‌ها می‌باشند.

۵-۱ ارزیابی مقاومت و دقت

در این بخش نتایج به‌دست‌آمده برای هر سه دیتاست را در قالب جدول ۳ نمایش داده‌شده که در موارد معیار دقت پاک،F1-score ماکرو، دقت در حالت حمله تخاصمی،F1-score در حالت حمله تخاصمی و اختلاف دقت‌ها است که در ادامه توضیح خواهیم داد. با توجه به نتایج مندرج در جدول ۳، در تمامی موارد چارچوب GRDSحداکثر باعث افت ۱/۳ درصد دقت نسبت به مدل‌های پایه در معیار دقت پاک‌شده است. این در حالی است که کاهش دقت در مواجهه با حملات تخاصمی به‌طور میانگین ۱۱ درصد بهتر از GNNGuard و حدود ۱۹ درصد بهتر از GCN خالص عملکرد داشته است. این یافته حاکی از آن است که چارچوب پیشنهادی توانسته است با کمترین هزینه بر روی عملکرد در شرایط عادی، مقاومت مدل را در برابر حملات ساختاری به‌شکل چشمگیری افزایش دهد.

تحلیل عمیق‌تر نتایج نشان می‌دهد که موفقیت GRDS در ایجاد تعادل بین دقت و مقاومت، ناشی از طراحی هوشمندانه تابع زیان چندهدفه است. از یک سو، حضور عبارات L_{imp} و L_{other} در تابع زیان، سپر را قادر ساخته است تا بدون تخریب ساختار معنایی گراف، یال‌های مخرب را شناسایی و حذف نماید. از سوی دیگر، برتری عملکرد GRDS نسبت به GNNGuard را می‌توان به تفاوت بنیادین در رویکرد دفاعی این دو روش نسبت داد.

[1] Validation

جدول (۲): مقایسه دقت در مدل‌ها و دیتاست‌های مورد پژوهش

دیتاست	مدل	Clean Acc (%)	F1-Score (ماکرو)	Robust Acc (%)	F1-Score (ماکرو)	ΔAcc
Cora	Vanilla GCN	۸۱/۲ ± ۰/٤	۷۹/۸ ±۰/٥	٦۲/۸ ±۱/۲	٦۰/٥ ± ۱/٤	−۱۸/٤
	GNNGuard	۸۰/٥ ± ۰/٥	۷۹/۱ ±۰/٦	٦۸/۳ ±۰/۹	٦٦/۷ ±۱/۰	−۱۲/۲
	GRDS (پیشنهادی)	۷۹/۹ ±۰/٤	۷۸/٦ ±۰/٥	۷۳/٦ ±۰/۸	۷۲/۱ ±۰/۹	−٦/۳
CiteSeer	Vanilla GCN	۷۱/٤ ± ۰/٦	٦۸/۹ ±۰/۷	٥۱/۹ ±۱/٥	٤۹/۲ ±۱/۷	−۱۹/٥
	GNNGuard	۷۰/۸ ± ۰/٥	٦۸/۳ ± ۰/٦	٥۸/۷ ±۱/۱	٥٦/۱ ± ۱/۳	−۱۲/۱
	GRDS (پیشنهادی)	۷۰/۲ ±۰/٥	٦۷/۸ ± ۰/٦	٦۸/۱ ±۰/۹	٦٥/۸ ± ۱/۱	−۲/۱
PubMed	Vanilla GCN	۸۹/۱ ± ۰/۳	۸۷/٤ ± ۰/٤	۷۱/٥ ± ۱/۰	٦۹/۳ ±۱/۲	−۱۷/٦
	GNNGuard	۸۸/۷ ±۰/٤	۸۷/۰ ±۰/٥	۷۷/٤ ±۰/۸	۷٥/٦ ± ۰/۹	−۱۱/۳
	GRDS (پیشنهادی)	۸۸/۳ ± ۰/۳	۸٦/٦ ±۰/٤	۸۲/۹ ± ۰/۷	۸۱/۲ ± ۰/۸	−٥/٤

GNNGuard با استفاده از مکانیزم توجه، صرفاً وزن یال‌های مخرب را کاهش می‌دهد، اما این یال‌ها همچنان در ساختار گراف حضور داشته و امکان نشت اطلاعات نادرست از طریق فرآیندهای پیام‌رسانی وجود دارد. در مقابل، GRDSبا رویکرد حذف فیزیکی یال‌های مخرب، جریان اطلاعات ناصحیح را به‌طور کامل مسدود می‌کند که این امر منجر به مقاومت بالاتر در برابر حملات می‌گردد.

۵-۲ تحلیل حساسیت و رفتار سپر

به‌منظور درک عمیق‌تر رفتار چارچوب GRDS و اعتبارسنجی انتخاب هایپرپارامترهای جدول ۲، تحلیل حساسیتی بر روی سه پارامتر اصلی w_{imp} (وزن جریمه حذف یال‌های مهم)، τ(دمای نمونه‌برداری Gumbel-Softmax) و K (تعداد گام‌های بهینه‌سازی داخلی) انجام پذیرفت. نتایج این تحلیل به شرح زیر است:

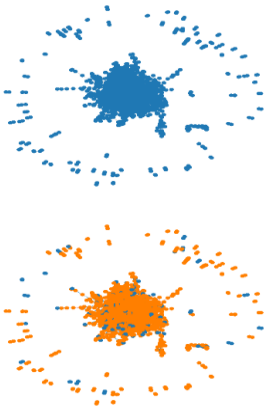
تحلیل حساسیت نسبت به wimp: با افزایش وزن w_{imp} از ۰/۵ به ۲/۰، مقدار شاخص Protection Gap از ۰/۱۲ به ۰/۲۸ ارتقا یافت. این بهبود نشان‌دهنده توانایی این پارامتر در هدایت سپر به سمت حفظ یال‌های درون‌کلاسی است. نکته حائز اهمیت اینکه دقت در حالت پاک در این بازه، نوسان بسیار کمی (≤۰/۳٪) داشت که مؤید آن است که w_{imp} صرفاً بر تمایز یال‌های مهم متمرکز بوده و تأثیر مخربی بر عملکرد کلی مدل در شرایط عادی ندارد. با عبور از ۱/۵، اگرچه شاخص Protection Gap همچنان افزایش می‌یابد، اما دقت مقاوم (Robust Accuracy) دچار افت جزئی می‌شود. این پدیده که می‌توان آن را **"بیش‌حفاظتی "**سپر نامید، ناشی از تمرکز بیش‌ازحد بر حفظ یال‌های درون‌کلاسی و حذف احتمالی برخی یال‌های بین‌کلاسی مفید است. بنابراین، مقدار بهینه w_{imp}=۱/۵ انتخاب گردید که بهترین تعادل را برقرار می‌سازد.

تحلیل حساسیت نسبت به دما :(τ) استراتژی کاهش دما تأثیر مستقیمی بر پایداری فرآیند آموزش دارد. کاهش سریع دما (به ۰/۱ در اپیاک ۵۰) موجب بی‌ثباتی گرادیان و نوسان شدید در تابع زیان طبقه‌بندی (L_{cls}) گردید که می‌تواند به همگرایی در نقاط بهینه محلی نامطلوب منجر شود. در مقابل، ثابت نگه‌داشتن τ=۰/۵ تا اپیاک ۱۰۰ و سپس کاهش خطی آن، هموارترین و پایدارترین روند همگرایی را به همراه داشت. این رفتار به پدیده **"انفجار گرادیان"** در نمونه‌برداری گسسته زودهنگام مرتبط است. دمای بالا در مراحل ابتدایی آموزش به سپر اجازه می‌دهد فضای حالت بزرگ‌تری را جستجو کند و با نزدیک شدن به نقطه بهینه، به‌تدریج تصمیمات خود را قطعی‌تر نماید. در آزمایش‌های اصلی، از استراتژی کاهش خطی دما استفاده‌شده است.

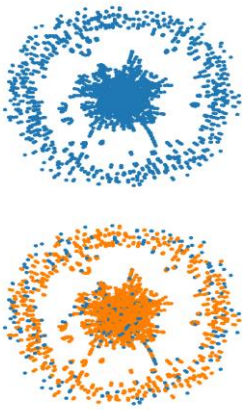
تحلیل حساسیت نسبت به K (گام‌های داخلی): تعداد گام‌های بهینه‌سازی داخلی سپر، نقش کلیدی در اثربخشی فرآیند کمینه-بیشینه ایفا می‌کند. با قرار دادن K=1، یادگیری ضعیف سپر (Protection Gap ≈۰/۰۷) و درنتیجه کاهش دقت مقاوم مشاهده گردید. افزایش K به ۳، بهبود چشمگیری در دقت مقاوم (افزایش

٪۷/۲) ایجاد کرد که نشان می‌دهد سپر برای کشف ساختارهای تخاصمی مؤثر، به چندین گام بهینه‌سازی نیازمند است. افزایش K به ۵، بهبود ناچیزی (حدود ٪۰/۹) در عملکرد ایجاد نمود درحالی‌که هزینه محاسباتی را ۲/۵ برابر افزایش داد. این یافته اهمیت انتخاب K= ۳را به‌عنوان نقطه بهینه در مبادله بین کارایی و اثربخشی تأیید می‌کند.

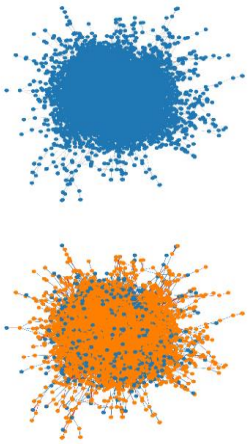
بررسی کیفی گراف‌ها که در شکل‌های ۲ تا ٤ مشاهده می‌کنید نشان‌دهنده آزمایش‌های انجام‌شده در زمان پیش و پس از اعمال GRDS بر روی دیتاست های Cora، CiteSeer و PubMed است.



شکل (۲): مقایسه گراف پیش (گراف بالا) و پس (گراف پایین) از اعمال سپر روی دیتاست Cora



شکل (۳): مقایسه گراف پیش (گراف بالا) و پس (گراف پایین) از اعمال سپر روی دیتاست CiteSeer



شکل (٤): مقایسه گراف پیش (گراف بالا) و پس (گراف پایین) از اعمال سپر روی دیتاست PubMed

همان‌طور که در شکل ۲ قابل‌مشاهده است، درصد حذف یال‌های بین گره‌های همنوع ٦٪ و بین گره‌های متفاوت ۲۳٪ است. این توزیع نشان‌دهنده این بوده که سپر به‌طور خودکار ایستادگی را در مقابل گره‌های هم‌کلاس حفظ می‌کند.

در شکل‌های ۳ و ٤ که به ترتیب مربوط به مجموعه‌داده‌های CiteSeer و PubMed است، یال‌های درون کلاس حذف‌شده توسط GRDS (حدود ٤-۵٪) نشان می‌دهد سپر به‌صورت خودکار به حفظ ارتباط‌های هم‌کلاس محور پیش‌فرض می‌پردازد، درحالی‌که حذف یال‌های بین کلاس بیشتر (حدود ۱۹-۲۳٪) است و هدف اصلی آن کاهش جریان اطلاعات نامفید و تقویت جداسازی طیفی گره‌ها است. این توزیع نهایی نشان می‌دهد که بهینه‌سازی دوسطحی (θ برای ماسک $\rightarrow$ ascent / وزن‌های descent $\rightarrow$ GCN) قادراست به‌صورت خودکار تعادل بین حفظ همبستگی‌های کلاسیک و حذف هم‌پوشانی‌های بین‌کلاسی را برقرار سازد.

۵-۳ تحلیل زمان اجرا و کارایی عملی

برای ارزیابی کارایی عملی چارچوب GRDS، زمان آموزش و استنتاج مدل‌های مختلف را بر روی دیتاست Cora اندازه‌گیری کردیم. تمامی آزمایش‌ها بر روی یک سیستم با GPU NVIDIA Tesla T4 انجام‌شده و نتایج در جدول ٤ گزارش‌شده‌اند.

جدول (۳): مقایسه زمان اجرا (بر حسب ثانیه) بر روی دیتاست Cora

مدل	زمان آموزش (هر ایپاک)	زمان استنتاج (1000 نمونه)
Vanilla GCN	۰/۰03 ±0/024	0/002 ± 0/012
GNNGuard	0/008 ± 0/087	0/004 ± 0/035
GRDS	0/005 ± 0/062	0/002± 0/013

نتایج جدول ٤ نشان می‌دهد که اگرچه GRDS در فاز آموزش نسبت به GCN خالص سربار محاسباتی دارد (حدود ۲/۵ برابر)، اما به‌طور قابل‌توجهی سریع‌تر از GNNGuard (حدود ۲۸٪ سریع‌تر) عمل می‌کند. نکته حائز اهمیت اینکه در فاز استنتاج، GRDSعملاً همان سرعت GCN خالص را دارد (۰/۰۱۳ ثانیه در مقابل ۰/۰۱۲ ثانیه)، درحالی‌که GNNGuard همچنان سربار محاسباتی قابل‌توجهی (حدود ۲/۷ برابر) دارد. این ویژگی، GRDSرا برای کاربردهای بلادرنگ و مقیاس بزرگ بسیار مناسب می‌سازد.

٤-۵ مسیرهای پژوهشی آینده

این پژوهش با اینکه در آزمایش‌های انجام‌شده نتایج قابل قبولی داشت وتنها با کاهش ٤ درصد دقت، ۱۱ درصد مقاومت در حملات تخاصمی را افزایش داد. در آینده مسیرهای این پژوهش را می‌توان در چهار مورد ادامه داد.

۱- فشرده‌سازی پارامتر سپر با استفاده از Sparse-Gumbel یا Low-rank factorization

۲- ترکیب GRDS با Adversarial Training برای تقویت دفاع در مقابل حملات پیشرفته‌تر

۳- ارزیابی و بهینه‌سازی سپر بر روی گراف‌های دینامیک و گراف‌های چندوجهی

٤- می‌توان قابلیت‌های GRDS را با روش‌های غنی‌سازی ویژگی گره‌ها، مانند افزودن معیارهای مرکزیت جهانی[۱] [۱] ترکیب کرد تا علاوه بر افزایش مقاومت دربرابر حملات، باعث بهبود دقت در حالت پاک نیز گردد.

٦- نتیجه‌گیری

ما چارچوب GRDS را به‌عنوان یک سپر مقاوم برای GNN ها معرفی کردیم. سپر با استفاده از Gumbel-Softmax و تابع زیان چندهدفه، به‌صورت انتهابه‌انتها (end to end) می‌آموزد؛ یال‌های مهم را حفظ و یال‌های مخرب را تار (ماسک) کند.

در ارزیابی‌های تجربی بر روی دیتاست‌های Cora، CiteSeer و PubMed، GRDS:

حداکثر ٤ ٪ در دقت پاک (Clean Acc) نسبت به مدل پایه کاهش می‌یابد.

Robust Accuracy به‌طور میانگین ۱۱ درصد نسبت به GNNGuard بهبود پیدا می‌کند.

در مقیاس Protection Gap مقدار Protection Gap تا ۰/۲۸ دست یافت، که نشان‌دهنده تمایز مؤثر بین یال‌های مهم و غیرمهم است.

نتایج نشان می‌دهند که اضافه کردن یک لایه دفاعی ماژولار می‌تواند هزینه محاسباتی را در مقایسه به روش‌های آموزش تخاصمی کاهش دهد، درحالی‌که تعادل بین دقت و مقاومت را حفظ می‌کند.

References

[1] Zügner, D., Akbarnejad, A., & Günnemann, S. (2018). Adversarial Attacks on Neural Networks for Graph Data. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 2847–2856). https://doi.org/ 10.1145/3219819.3220078

[2] Z. Wu et al., "A comprehensive survey on graph neural networks," IEEE Trans. Neural Netw. Learn. Syst., vol. 31, no. 1, pp. 4–24, Jan. 2020, doi: 10.1109/TNNLS.2020.2978386.

[3] S. Zhang, J. Fang, W. Feng, L. Chen, R. Li, and C. Li, "Simple defense methods are the best defense methods: An extensive study on graph neural network backdoor attacks and defenses," arXiv preprint arXiv:2412.08016, 2024.

[4] Cheng, H. (2020). Defending Graph Neural Networks against Adversarial Attacks [Master's thesis, Massachusetts Institute of Technology]. MIT DSpace. http://hdl.handle.net/1721.1/137496.

[5] Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models

[۱] Pagerank

resistant to adversarial attacks," in International Conference on Learning Representations (ICLR), 2018. [Online]. Available: https://openreview.net/forum?id=rJzIBfZAb.

[6] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903.

[7] Chen, J., & Cheng, H. (2023). Robust graph learning with graph convolutional network. Pattern Recognition, *136*, 109266.

[8] S. Doostali, S. M. Babamir, “An energy efficient cluster head selection approach for performance improvement in network-coding-based wireless sensor networks with multiple sinks,” *Computer Communications*, vol. 164, pp. 188-200, 2020, doi:10.1016/j.comcom.2020.10.014.

[9] Jin, W., Ma, Y., Liu, X., Tang, X., Wang, S., & Tang, J. (2020, August). Graph structure learning for robust graph neural networks. In Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 66-74).

[10] H. Hosseini, M. Mirzaei, and M. A. Javadi, "[Node classification in citation graphs using graph neural networks]," presented at the 5th Natl. Conf. Comput. Eng. Inf. Technol. Manage., Tehran, Iran, 2023. [Online]. Available: https://civilica.com/doc/2004315

[11] H. Hosseini and M. Naghavi, "[Improving node classification accuracy in graph neural networks using PageRank as an additional node feature]," presented at the 2nd Int. Conf. Adv. Res. Comput. Eng., Tehran, Iran, 2024. [Online]. Available: https://civilica.com/doc/2447199

[12] Z. Hou, M. Lin, M. A. Torkamani, S. Wang, and X. Liu, "Adversarial robustness in graph neural networks: Recent advances and new frontier," in Proc. IEEE 11th Int. Conf. Data Sci. Adv. Anal. (DSAA), San Diego, CA, USA, Oct. 2024, pp. 1–10, doi: 10.1109/DSAA.2024.00000.

[13] T. Wu et al., "Understanding the robustness of graph neural networks against adversarial attacks," Knowl.-Based Syst., vol. 323, p. 113714, Jul. 2025, doi: 10.1016/j.knosys.2025.113714.

Journal of Applied And Basic Machine Intelligence Research (ABMIR)

Applied and Basic Machine Intelligence Research

Vol. 4, No. 2, Autumn & Winter 2026, pp.17 - 29

Graph Randomization as a Differentiable Shield (GRDS): A Method for Enhancing the Robustness of Graph Neural Networks

Ali Hosseinpournaderi[1] Mohammad Ali Javadzade [2*], Hossein Hosseini[3]

[1] Master's student in Artificial Intelligence and Robotics, Faculty of Artificial Intelligence and Cognitive Sciences, Imam Hossein Comprehensive University, Tehran, Iran

[2] Assistant Professor, Faculty of Artificial Intelligence and Cognitive Sciences, Imam Hossein Comprehensive University, Tehran, Iran

[3] PhD's student in Artificial Intelligence and Robotics, Faculty of Artificial Intelligence and Cognitive Sciences, Imam Hossein Comprehensive University, Tehran, Iran

Article Information

Original Research Paper

Received:
2026 January 29

Accepted:
2026 May 04

Keywords:
Graph Neural Networks, Adversarial Robustness, Differentiable Randomization, Multi-objective Loss, Graph Defense

Corresponding Author*:
javadzade@ihu.ac.ir

Abstract

The vulnerability of Graph Neural Network (GNN) to adversarial attacks remains a fundamental challenge in the field, limiting their deployment in sensitive and high-risk applications. In this research, we introduce an intelligent defensive framework named Graph Randomization as a Differentiable Shield (GRDS), which addresses this challenge at a reasonable computational cost. The core innovation of our work lies in providing a practical solution that, unlike conventional methods, enhances model robustness without causing a significant drop in accuracy on clean data. The methodology underpinning our framework is built upon a modular shield. Utilizing a multi-objective and intelligent loss function, this shield learns to differentially distinguish between critical and non-critical edges in the graph. Positioned before the Graph Neural Network, it misleads potential attackers by intentionally obfuscating the graph structure. The framework is implemented using PyTorch and the PyTorch Geometric library. Comprehensive evaluations were conducted on standard datasets (Cora, PubMed, and CiteSeer) against various attack methods (gradient based, random, etc.). The extracted results demonstrate that GRDS incurs a negligible cost in accuracy (a decrease of less than 4%) while substantially increasing the model's robustness compared to the baseline model (an improvement of more than 11%). This finding underscores the principle that effective defense through intelligent randomization outperforms blind removal strategies.

: 10.22034/ABMIR.2026.24239.1220

E-ISSN: 2821-2037