چارچوب پیش‌بینی پیوند با استفاده از شبکه عصبی گرافی مبتنی بر زیرگراف

سید مهدی وحیدی پور[1*]، استادیار، ریحانه کرمی [2]، دانشجوی کارشناسی ارشد

[1] دانشکده مهندسی برق و کامپیوتر-دانشگاه کاشان -کاشان ـ ایران ـ vhaidipour@kashanu.ac.ir

[2] دانشکده مهندسی برق و کامپیوتر-دانشگاه کاشان -کاشان ـ ایران ـ r.karami@grad.kashanu.ac.ir

چکیده: پیش‌بینی پیوند یکی از موضوع‌های مهم در تجزیه و تحلیل شبکه‌های پیچیده است. پیش‌بینی پیوند می‌تواند توسط یک رده‌بند انجام شود؛ به‌طوری‌که بردار ویژگی یک جفت گره، ورودی آن باشد. خروجی رده‌بند نشان می‌دهد که آیا میان آن جفت گره پیوندی پیش‌بینی می‌شود یا خیر (رده یک یا رده صفر). برای استخراج بردار ویژگی یک جفت گره می‌توان از شبکه‌های عصبی گرافی (GNN) استفاده نمود که در اینصورت روش حل مسئله پیش‌بینی پیوند مبتنی بر شبکه عصبی گرافی خواهد بود. در این مقاله، یک روش حل مسئله پیش‌بینی پیوند مبتنی بر شبکه عصبی گرافی به نام GAE (Graph Auto Encoder) به‌عنوان روش پایه در نظر گرفته شده است. یکی از مشکل‌های اساسی در این روش آن است که بردار ویژگی استخراج شده توسط شبکه عصبی گرافی به ازای جفت گره‌های متفاوت، می‌تواند یکسان باشد. برای رفع این مشکل، در این مقاله با استفاده از مفهوم زیرگراف روش پایه بهبود داده شده و چارچوب جدیدی با نام SGAE (Sub-Graph Auto Encoder) پیشنهاد شده است. چارچوب پیشنهادی بر اساس معیارهای مختلف ارزیابی و با روش پایه مقایسه شده‌است که نتایج نشان‌دهنده‌ی بهبود عملکرد آن است. به‌طور مثال روش SGAE به‌طور متوسط نسبت به روش پایه GAE در معیارهای دقت، F1-Score، متوسط صحت و مساحت زیر نمودار صحت-فراخوانی، بهبود ۵,۵، ۵، ۵,۷۵ و ۵,۸۷ را ایجاد کرده است.

واژه‌های کلیدی: شبکه‌های پیچیده، شبکه عصبی گرافی، پیش‌بینی پیوند، رده‌بندی، زیرگراف، یادگیری بازنمایی گراف.

* **سیدمهدی وحیدی پور، vahidipour@kashanu.ac.ir**

Link prediction framework using graph neural network based on subgraph

S. Mehdi Vahidipour[1*], Assistant Professor, Reyhane Karami[2], Master Student

[1] Electrical and Computer Engineering Department, University of Kashan, Kashan, Iran, vahidipour@kashanu.ac.ir

[2] Electrical and Computer Engineering Department, University of Kashan, Kashan, Iran, r.karami@grad.kashanu.ac.ir

***Abstract*:** Link prediction is one of the important topics in complex network analysis. Link prediction can be done by a classifier such that the feature vector of a pair of nodes is its input. The output of the classifier indicates whether a link is predicted between that pair of nodes (class one or class zero). To extract the feature vector of a pair of nodes, graph neural networks (GNN) can be used, in which case the method of solving the link prediction problem will be based on graph neural networks. In this paper, a GNN-based link prediction problem solving method called GAE is considered as the basic method. One of the basic problems in this method is that the feature vector extracted by graph neural networks can be the same for different pairs of nodes. To solve this problem, in this paper, using the concept of subgraph, the basic method is improved and a new framework called SGAE is proposed. The proposed framework has been compared with the basic method based on different evaluation criteria, the results show the improvement of its performance. For example, the SGAE method has improved 5.5, 5, 5.75 and 5.87 compared to the basic GAE method in terms of accuracy, F1-Score, average precision and area under the precision-recall curve.

Keywords: *complex networks; Graph Neural Networks; link prediction; classification; subgraph; Graph Representation Learning.*

*** S. Mehdi Vahidipour, vahidipour@kashanu.ac.ir**

۱. مقدمه

بسیاری از سیستم‌های اجتماعی، بیولوژیکی و اطلاعاتی می‌توانند به خوبی توسط شبکه‌های پیچیده، به‌صورت موجودیت‌هایی که با یکدیگر در ارتباط هستند، توصیف شوند [۱]. یکی از روش‌های توصیف گراف است؛ به‌صورتی که گره‌ها نشان دهنده موجودیت و پیوندها نشان دهنده ارتباط بین آن‌ها است.

یکی از موضوع‌های علمی مهم مرتبط با شبکه، پیش‌بینی پیوند در گراف است که می‌توان آن را به‌عنوان پیش‌بینی روابط در شبکه مطرح کرد. پیش‌بینی پیوند، به بررسی پیش‌بینی روابط جدید می‌پردازد [۲]. هدف پیش‌بینی پیوند، یافتن احتمال شکل‌گیری پیوند در شبکه، براساس اتصال‌های مشاهده شده کنونی است [۲]. پیش‌بینی پیوند اساسی‌ترین مسئله‌ای است که تلاش می‌کند احتمال وجود پیوند بین دو گره براساس پیوندهای مشاهده شده و ویژگی‌های گراف تخمین زده می‌شود [۳].

اخیرا از مفهوم تعبیه[1] برای حل مسئله پیش‌بینی پیوند استفاده می‌شود [۴-۷]؛ تعبیه، یک بردار ویژگی است که تعداد درایه‌های آن (اَبعاد) به نسبت اندازه گراف بسیار کم است. به ازای هر المان در گراف (مانند گره، پیوند، زیرگراف و حتی کل گراف) می‌توان تعبیه ایجاد کرد. به‌عبارت دیگر، یک المان در گراف به کمک تعبیه‌ها در یک فضای دیگری (به نام فضای تعبیه) بازنمایی[2] می‌شود. روش‌های یادگیری بازنمایی گراف[3] به روش‌های تولید بردارهای تعبیه گفته می‌شود که توسط مدل‌های یادگیری ایجاد شده باشند [۱].

یکی از روش‌های یادگیری بازنمایی گراف، شبکه‌های عصبی گرافی[4] (GNN) است [۸]، که در آن برای هر گره یک تعبیه ایجاد می‌شد. از GNN می‌تواند در حل مسئله پیش‌بینی پیوند استفاده می‌شود که یکی از معروف‌ترین آن‌ها روش GAE[5] است [۹]. در روش GAE برای پیش‌بینی پیوند میان دو گره هدف، ابتدا تعبیه‌ی آن جفت گره هدف بدست می‌آید و سپس با استفاده از ترکیب دو تعبیه یک بازنمایی (بردار ویژگی) ایجاد می‌شود؛ براساس این بازنمایی احتمال وجود پیوند تخمین زده می‌شود. یکی از ضعف‌های روش GAE این است که در این روش تنها بازنمایی گره‌ها یادگرفته می‌شود و بازنمایی پیوند از روی آنها ساخته می‌شود. به‌همین دلیل اگر جفت گره‌های دو سر دو پیوند غیرهمریخت، به‌صورت دو به دو همریخت[6] باشند، بازنمایی یکسانی برای پیوندها ایجاد می‌شود [۱۰].

برای رفع این ضعف روش GAE، در این مقاله پیشنهاد می‌شود تا مفهوم و ویژگی زیرگراف به آن اضافه شود. در همین راستا، چارچوب پیشنهادی SGAE[7] پیشنهاد می‌شود. زیرگراف‌ها زیرمجموعه‌ای از گراف را نشان می‌دهد که شامل تعدادی گره و روابط بین آن‌ها است. با توجه به اینکه، برای حل مسئله پیش‌بینی پیوند از زیرگراف استفاده شده است، نوآوری‌های این مقاله به شرح زیر است:

- ارائه چارجوب SGAE مبتنی بر ترکیب زیرگراف و روش GAE که مسئله پیش‌بینی پیوند را به‌طور کاراتری حل می‌کند. برای ایجاد SGAE یک الگوریتم دو مرحله‌ای شامل استخراج زیرگراف و ایجاد بردار ویژگی زیرگراف نیاز است که دو ایده متفاوت برای انجام آنها نیز پیشنهاد شده است.
- ایده‌ای برای استخراج زیرگراف که در SGAE پیشنهاد شده است با روش پرکاربرد h-hop [۱۱] رقابت می‌کند.

[1] Embedding
[2] Representation
[3] Graph Representation Learning
[4] Graph Neural Network
[5] Graph Auto Encoder
[6] Isomorph
[7] Sub-Graph Auto Encoder

- ایده‌ای برای ایجاد بردار ویژگی زیرگراف پیشنهاد شده است که در چارچوب پیشنهادی از روش شبکه عصبی کانولوشنال[1] [۱۲] کاراتر است.

ساختار مقاله در ادامه به شرح زیر است: بخش دوم به مرور ادبیات و کارهای مرتبط می‌پردازد. بخش سوم روشی را که چارچوب پیشنهادی، مبتنی بر آن ایجاد شده است را معرفی می‌کند. در بخش چهارم به معرفی روش‌های پیشنهادی پرداخته شده است. در بخش پنجم، آزمایش‌های انجام شده و نتایج آن‌ها گفته می‌شود و در آخر، در بخش ششم، نتیجه‌گیری بیان می‌شود.

۲. مرور ادبیات و کارهای مرتبط

در این بخش به مرور ادبیات و کارهای مرتبط پرداخته می‌شود. با توجه به این که در این مقاله از زیرگراف، برچسب گذاری، بردار ویژگی و شبکه عصبی گرافی برای حل مسئله پیش‌بینی پیوند استفاده شده است، در ادامه این مفاهیم شرح داده می‌شوند.

۱,۲. زیرگراف[2]

گراف $G(V,E)$ با مجموعه گره V و مجموعه پیوند E را در نظر بگیرید. زیرگراف S_G، زیرمجموعه‌ای از گره‌ها و پیوندهای گراف است که به‌صورت $S_G(\acute{V},\acute{E})$ نشان داده می‌شود؛ به‌طوری که $\acute{V} \subseteq V$ و $\acute{E} \subseteq E$ باشد.

استخراج یک زیرگراف به تعیین مجموعه‌های $\acute{V}$ و $\acute{E}$ گفته می‌شود به گونه‌ای که شامل المان یا المان‌های مشخصی باشد. به‌طور مثال برای پیش‌بینی پیوند میان دو گره مشخص (هدف)، زیرگراف استخراج شده از گراف اصلی باید شامل جفت گره هدف باشد. در ادامه دو دسته از روش‌های استخراج زیرگراف مرور می‌شوند:

- **مبتنی بر همسایگی**: روش‌های استخراج زیرگراف مبتنی بر همسایگی، گره‌های همسایه‌ی سطح‌های متفاوت گره یا پیوند هدف را به‌عنوان گره‌های زیرگراف استخراج می‌کنند؛ به‌طوری‌که اگر A گره هدف باشد و $\mathcal{N}_h(A)$ نشان‌دهنده‌ی همسایه‌های گره A در سطح h باشد، $\acute{\boldsymbol{V}} \subseteq \mathcal{N}_h(A)$ است و پیوندهای بین گره‌های زیرگراف در مجموعه $\acute{\boldsymbol{E}}$ قرار می‌گیرد. یکی از پرکاربردترین روش‌های ایجاد زیرگراف مبتنی بر همسایگی، روش h-hop [۱۱] است که در بخش ۲,۱,۴. به معرفی آن پرداخته شده است.

- **مبتنی بر جامعه**[3]: جامعه یک نوع زیرگراف است؛ که به‌طور گسترده‌ای در شبکه‌ها وجود دارد و به‌عنوان یکی از ضروری‌ترین و مفیدترین استراتژی‌های شبکه شناخته شده‌است. جامعه مجموعه‌ای از گره‌هایی است که احتمالا ویژگی‌های مشترکی دارند و یا نقش‌های مشابهی در شبکه ایفا می‌کنند. پیوندهای بین گره‌ها در یک جامعه یکسان متراکم و در بین جوامع متفاوت پراکنده است [۱۳, ۱۴].

۲,۲. برچسب‌گذاری[4]

برچسب‌گذاری به معنی نسبت دادن برچسب به گره‌ها یا پیوندهای گراف است. برچسب‌ها می‌توانند نشان دهنده اولویت، ترتیب، اهمیت و ... گره‌ها یا پیوندها باشند. هدف اصلی برچسب‌گذاری، نشان دادن نقش‌های متفاوت گره‌ها یا پیوندهای مختلف در گراف است [۴]. روش‌های برچسب‌گذاری گره استفاده شده در پیش‌بینی پیوند باید توانایی متمایز کردن گره‌های

[1] Convolutional Neural Network

[2] Subgraph

[3] Community

[4] Labelling

هدف از گره‌های دیگر و حفظ ترتیب گره‌ها در صورت تکرار را داشته باشند [۱۰].

۳،۲. بردار ویژگی[1]

بردار ویژگی، برداری تک بعدی و یا چند بعدی است؛ که می‌تواند شامل ویژگی‌های المان‌های گراف باشد؛ که به آن بردار بازنمایی نیز گفته می‌شود. در این مقاله بردار ویژگی گره (A)،گراف (G) و زیرگراف (S_G)، به‌ترتیب با $h(A)$، $h(G)$ و $h(S_G)$ نشان داده می‌شود. روش‌های متفاوتی برای ایجاد بردار ویژگی معرفی شده است که در این‌جا به دو دسته‌ی سنتی و ایجاد تعبیه دسته‌بندی شده است.

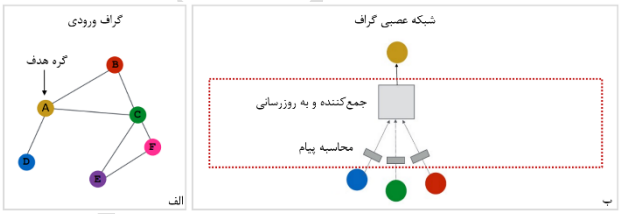
- **روش‌های سنتی**: روش‌های سنتی ایجاد بردار ویژگی، روش‌هایی هستند که در آن‌ها به‌صورت دستی اعداد درون بردار ویژگی مشخص می‌شوند و می‌توان دلیل و روش ایجاد این اعداد را توضیح داد. برای مثال ماتریس همسایگی یک بردار ویژگی دو بعدی ساده است که هر درایه درون آن نشان دهنده وجود و یا عدم وجود پیوند است.

- **روش‌های ایجاد تعبیه**: تعبیه نوعی بردار ویژگی با ابعاد پایین است که بوسیله مدل‌های یادگیری ایجاد می‌شود و مقادیر آن قابل تحلیل نیست. یکی از روش‌های ایجاد تعبیه استفاده از شبکه عصبی گرافی می‌باشد.

۴،۲. شبکه عصبی گرافی

شبکه عصبی گرافی یا GNN یک چارچوب یادگیری عمیق در گراف است که در سال‌های اخیر بسیار مورد توجه قرار گرفته است. شکل‌های اولیه GNN در [۱۵-۱۷] آمده است. GNN برای هر گره، بوسیله پیام‌ها و اطلاعات ارسال شده توسط گره‌های همسایه آن گره، تعبیه ایجاد می‌کند. GNN در هر لایه از سه بخش اساسی تشکیل شده است؛ ۱) محاسبه پیام[2] ۲) جمع کننده[3] و ۳) به‌روزرسانی[4]؛ که در بخش محاسبه پیام، بردار ویژگی گره‌ها به پیام تبدیل می‌شود. در بخش جمع کننده، پیام همسایه‌های گره با یکدیگر جمع و تبدیل به یک بردار ویژگی می‌شود و در بخش به‌روزرسانی، به‌منظور از بین نرفتن اطلاعات گره مورد نظر، بردار ویژگی تولید شده در بخش دوم با پیام آن گره بوسیله توابعی مانند جمع و الصاق ترکیب شده و در نهایت بردار ویژگی جدید ایجاد می‌شود.

در شکل (۱) ساختار کلی یک شبکه عصبی گرافی تک لایه نشان داده شده است. در قسمت الف شکل، گراف ورودی نشان داده شده که گره A در آن گره هدفی است که تعبیه آن در قسمت ب با توجه به همسایه‌های آن محاسبه می‌شود. همانطور که نشان داده شده است، مجموعه همسایه‌های گره A شامل گره‌های B، C و D است؛ که در قسمت ب، بردارهای ویژگی آن‌ها ابتدا در قسمت محاسبه پیام، به پیام تبدیل شده، سپس در قسمت جمع‌کننده و به‌روزرسانی، پیام همسایه‌ها جمع شده و بردار ویژگی جدیدی را ایجاد می‌کند، سپس بردار ویژگی ایجاد شده با بردار ویژگی گره A ترکیب شده و در نهایت تعبیه جدید گره A به‌دست می‌آید.



شکل (۱): تصویر ساختار کلی شبکه عصبی گرافی تک لایه.

۵،۲. کارهای مرتبط

[1] Feature Vector

[2] Message Computation

[3] Aggregation

[4] Update

پیش‌بینی پیوند یک مسئله کلیدی در تحلیل شبکه است. در پیش‌بینی پیوند تلاش می‌شود که احتمال ایجاد پیوند بین دو گره در آینده، براساس پیوندهای مشاهده شده و ویژگی‌های شبکه تخمین زده شود. در بسیاری از کارها برای بهبود مسئله پیش‌بینی پیوند از بردار ویژگی‌های زیرگراف استفاده شده است. روش‌های ایجاد بردار ویژگی زیرگراف را همانطور که در بخش ۳,۲ گفته‌شد می‌توان به دو دسته‌ی ۱) روش‌های سنتی و ۲) روش‌های ایجاد تعبیه تقسیم‌بندی کرد. در دسته اول بردار ویژگی به‌صورت دستی و براساس ویژگی‌های قابل مشاهده زیرگراف ایجاد می‌شود در صورتی که در دسته دوم بردار ویژگی بوسیله شبکه‌های عصبی و یا روش‌های قدم‌زدن تصادفی ایجاد می‌شوند. در روش ایجاد تعبیه بوسیله شبکه‌های عصبی، اکثراً از روش‌های سنتی برای ایجاد بردار ویژگی اولیه برای ورودی شبکه عصبی استفاده می‌شود.

در این بخش به بررسی کارهایی پرداخته شده که: ۱) از رده‌بند[1] برای حل مسئله استفاده کرده‌اند و ۲) در حل مسئله پیش‌بینی پیوند از ویژگی‌ها و اطلاعات زیرگراف برای ایجاد بردار ورودی رده‌بند استفاده کرده‌اند.

در پژوهش‌های [۱۸-۲۰] از روش‌های سنتی برای ایجاد بردار ویژگی زیرگراف استفاده شده است. در این پژوهش‌ها ابتدا برای هر جفت گره هدف بوسیله روش h-hop زیرگراف استخراج شده و سپس گره‌های زیرگراف بوسیله روش‌های برچسب‌گذاری، برچسب‌گذاری می‌شوند. در این مرحله برای هر زیرگراف برچسب‌گذاری شده به‌صورت دستی بردار ویژگی ایجاد می‌شود. در [۱۸] از وجود ویا عدم وجود پیوند بین هر جفت گره زیرگراف و در [۱۹, ۲۰] از ۵ معیار شباهت و معیار وزن برای شبکه‌های وزن‌دار برای ایجاد بردار ویژگی زیرگراف استفاده می‌شود. در آخر برای پیش‌بینی پیوند جفت گره هدف، بردار ویژگی زیرگراف به رده‌بند داده می‌شود.

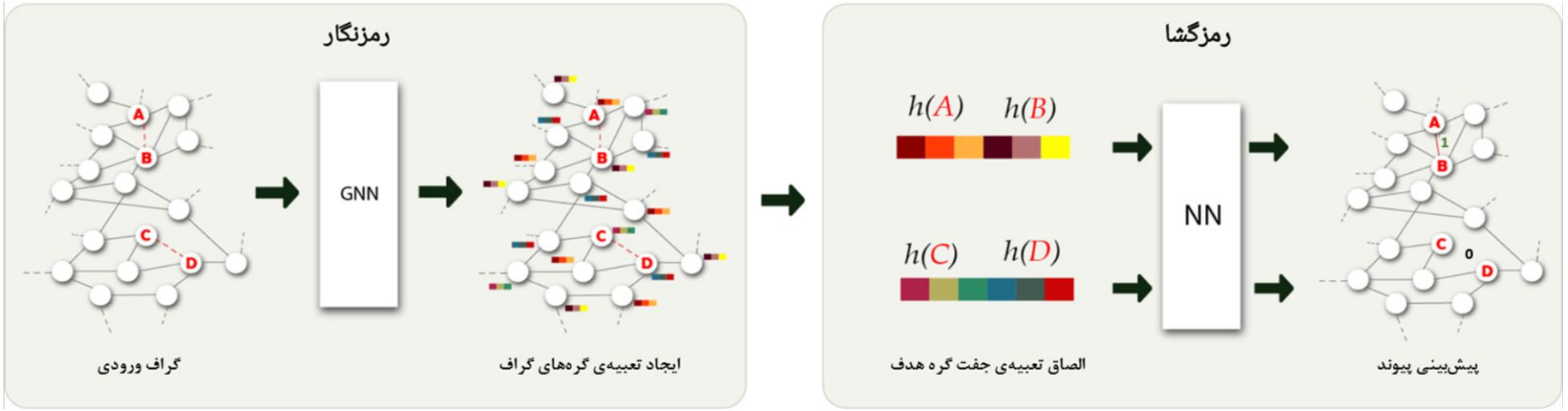
در پژوهش‌های [۴-۷] از روش‌های ایجاد تعبیه برای ایجاد بردار ویژگی زیرگراف استفاده شده است. در [۴, ۵] ابتدا برای هر جفت گره هدف با روش h-hop یک زیرگراف استخراج و برچسب‌گذاری می‌شود. در پژوهش اول بوسیله روش‌های قدم‌زدن تصادفی و در پژوهش دوم بوسیله شبکه‌های عصبی گرافی، تعبیه گره‌های زیرگراف ایجاد می‌شود. این تعبیه‌ها به‌منظور پیش‌بینی پیوند به رده‌بند شبکه‌های عصبی داده می‌شود. در پژوهش [۷] ابتدا برای هر گره هدف، بوسیله روش h-hop، زیرگراف‌هایی با عمق‌های متفاوت استخراج و سپس گراف‌لت‌های[2] زیرگراف‌های ایجاد شده برای هر گره شمرده و بردار ویژگی آن ایجاد می‌شود. این بردار ویژگی به شبکه رمزنگار خودکار گراف داده برای هر گره یک تعبیه ایجاد می‌کند. تعبیه جفت گره‌های هدف به یکدیگر متصل شده و برای پیش‌بینی پیوند به شبکه عصبی داده می‌شود. در پژوهش [۶] یک روش برای ایجاد تعبیه گره‌های شبکه مبتنی‌بر قدم‌زدن تصادفی در زیرگراف جامعه پیشنهاد و برای حل مسئله پیش‌بینی پیوند استفاده شده است.

۳. روش GAE

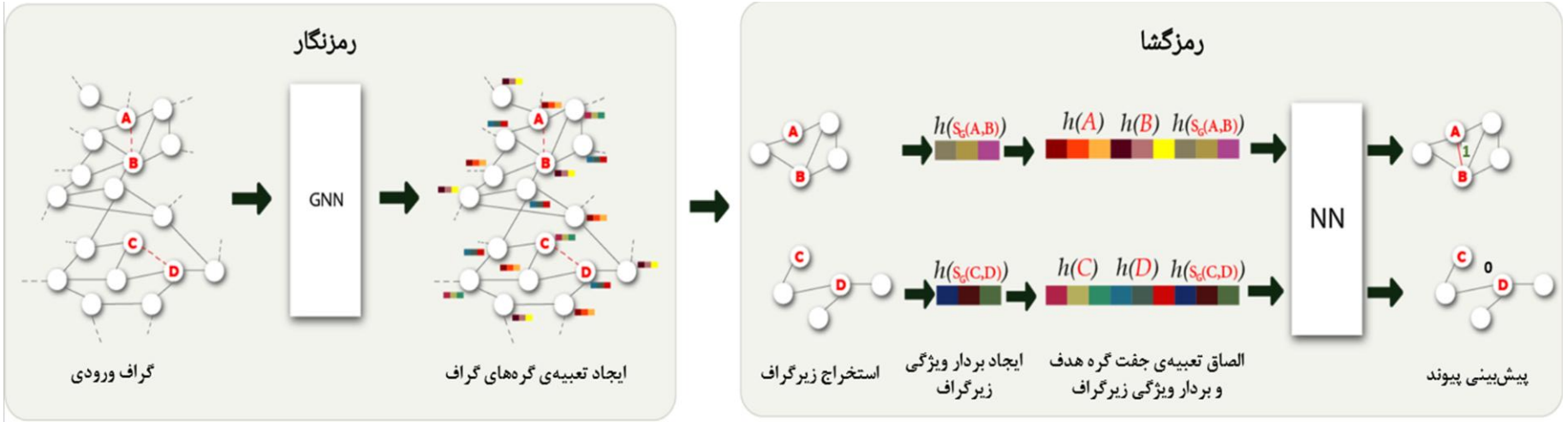
یکی از روش‌هایی که از تعبیه برای پیش‌بینی پیوند استفاده می‌کند، روش GAE [۹] است. این روش به دو بخش رمزنگار و رمزگشا تقسیم می‌شود. در بخش رمزنگار تعبیه‌های گره‌های گراف ایجاد می‌شود و در بخش رمزگشا پیش‌بینی پیوند انجام می‌شود.

[1] Classifier

[2] Graphlet



شکل (۲): چارچوب **GAE**.



شکل (۳): چارچوب **SGAE**.

- **رمزنگار:** در بخش رمزنگار، بوسیله شبکه عصبی گرافی تعبیه گره‌های گراف ایجاد می‌شود. شبکه عصبی گرافی ماتریس همسایگی و ماتریس ویژگی گراف را به‌عنوان ورودی گرفته و تعبیه گره‌های گراف را ایجاد می‌کند.
- **رمزگشا:** در بخش رمزگشا، تعبیه جفت گره هدف به یکدیگر الصاق شده و بردار ویژگی پیوند هدف را ایجاد می‌کند. بردار ویژگی پیوند هدف به‌منظور پیش‌بینی وجود و یا عدم وجود پیوند به رده‌بند شبکه عصبی ساده داده شده و پیش‌بینی پیوند انجام می‌شود.

در شکل (۲) مثالی از روش GAE نشان داده شده است. در سمت چپ شکل، بخش رمزنگار قرار دارد که در آن ابتدا گراف ورودی با جفت گره‌های هدف (A, B) و (C, D) که به‌ترتیب در رده‌های ۱ و ۰ قرار دارند، به شبکه عصبی گرافی داده می‌شوند. شبکه عصبی گراف، برای هر گره‌ی گراف یک تعبیه ایجاد می‌کند. تعبیه‌های ایجاد شده در بخش رمزنگار به بخش رمز گشا داده می‌شود. همانطور که در ابتدای بخش رمزگشا نشان داده شده است، تعبیه جفت گره‌های هدف (A, B) و (C, D) به یکدیگر الصاق شده و برای پیش‌بینی پیوند به رده‌بند شبکه عصبی داده می‌شود.

۴. چارچوب پیشنهادی SGAE

چارچوب پیشنهادی SGAE بر اساس روش پایه GAE است که در حل مسئله پیش‌بینی پیوند ویژگی زیرگراف شامل دو جفت گره هدف را نیز در نظر می‌گیرد. این چارچوب مانند روش پایه، از دو قسمت رمزنگار و رمزگشا تشکیل شده است که در قسمت رمزگشای آن به‌منظور استفاده از زیرگراف، یک الگوریتم دو مرحله‌ای اضافه شده است (شکل (۳))

در قسمت رمزنگار چارچوب SGAE، همانند روش پایه، گراف ورودی دریافت و برای تمام گره‌های آن بوسیله شبکه عصبی گرافی تعبیه ایجاد می‌شود. از تعبیه‌های استخراج شده، در قسمت رمزگشا استفاده می‌شود.

تغییرهای چارچوب پیشنهادی نسبت به روش پایه در قسمت رمزگشا است. در قسمت رمزگشا، یک الگوریتم دو مرحله‌ای برای تزریق زیرگراف به روش پایه اضافه شده است. در مرحله اول این الگوریتم، برای هر جفت گره هدف، یک زیرگراف استخراج می‌شود. سپس در مرحله دوم، برای هر زیرگراف یک بردار ویژگی زیرگراف ایجاد می‌شود. در انتها و در پیش‌بینی کننده، تعبیه گره‌های هدف و بردار ویژگی زیرگراف به‌صورتی به یکدیگر الصاق می‌شوند که بردار ویژگی زیرگراف در انتها قرار گیرد. بردار ویژگی ایجاد شده را بردار ویژگی پیوند هدف مبتنی بر زیرگراف می‌نامیم؛ که برای پیوند هدف (A, B) با $h(A, B)_{S_G}$ نشان داده می‌شود. در نهایت برای پیش‌بینی پیوند، بردار ویژگی پیوند مبتنی بر زیرگراف به‌عنوان ورودی به رده‌بند شبکه عصبی داده می‌شود.

شکل (۳) چارچوب پیشنهادی SGAE را نشان می‌دهد. گراف G به‌عنوان ورودی به چارچوب داده می‌شود. همانطور که در این شکل نشان داده شده است، در بخش رمزنگار بوسیله شبکه عصبی گرافی برای تمام گره‌های گراف G، تعبیه ایجاد می‌شود (به‌عنوان مثال برای گره‌های Aو B، به‌ترتیب تعبیه‌های $h(A)$ و $h(B)$ تولید می‌شود). در بخش رمزگشا برای هر جفت گره هدف (A , B)و (C, D) زیرگراف‌های $S_G(A,B)$ و $S_G(C, D)$ ایجاد می‌شود. در ادامه با استفاده از روش ایجاد بردار ویژگی زیرگراف، بردار ویژگی زیرگراف‌های $h(S_G(A,B))$ و $h(S_G(C, D))$ ایجاد می‌شود. بعد از ایجاد بردار ویژگی زیرگراف، تعبیه گره‌های هدف (به‌عنوان مثال $h(B)$و $h(A)$) و

بردار ویژگی زیرگراف (به‌عنوان مثال $h(S_G(A,B))$) به یکدیگر الصاق شده و بردار ویژگی پیوند هدف مبتنی بر زیرگراف (به‌عنوان مثال $h(A,B)_{S_G}$) را ایجاد می‌کند؛ که برای پیش‌بینی پیوند به ورودی رده‌بند شبکه عصبی داده می‌شود. در نهایت رده‌ی هر جفت گره پیش‌بینی می‌شود؛ به‌طوری‌که اگر بین جفت گره هدف پیوندی پیش‌بینی شود، در رده ۱ و در غیر این‌صورت در رده ۰ رده‌بندی می‌شود.

همان‌طور که گفته شد، در چارچوب پیشنهادی، یک الگوریتم دو مرحله‌ای وجود دارد که شامل استخراج زیرگراف و ساخت بردار ویژگی زیرگراف است. در ادامه، روش‌های پیشنهادی برای این دو مرحله ارائه می‌شود. در انتهای این قسمت نیز یک تحلیل برای SGAE ارائه خواهد شد.

۱،۴. روش پیشنهادی استخراج زیرگراف

در این بخش دو روش برای استخراج زیرگراف معرفی می‌شود؛ که یکی از آن‌ها روش پیشنهادی DIS و دیگری روش h-hop است.

۱،۱،۴. روش DIS

در روش پیشنهادی DIS، فرض می‌شود که گره‌های گراف دارای بردار ویژگی هستند[1]. در استخراج زیرگراف مبتنی بر جفت گره هدف، ابتدا گره‌ای فرضی به نام $DummyTarget$ فرض می‌شود و برای این گره فرضی، با استفاده از رابطه (۱) بردار ویژگی محاسبه می‌شود ($h(DummyTarget)$این بردار را نشان می‌دهد). چنانچه زیرگراف تنها مبتنی بر یک گره، استخراج شود آنگاه بدیهی است که تک گره هدف همان گره $DummyTarget$ است و بردار ویژگی گره جدید همان بردار ویژگی گره هدف است.

$$h(DummyTarget) = \frac{(h(A) + h(B))}{2} \qquad (۱)$$

هر گره مانند z فاصله‌ای با گره $DummyTarget$ دارد که با $d(z)$ نشان داده می‌شود. بعد از به‌دست آوردن فاصله گره‌های گراف با گره $DummyTarget$، گره‌های گراف براساس فاصله از کوچک به بزرگ مرتب شده و k نزدیک‌ترین گره به گره $DummyTarget$ استخراج شده و زیرگراف را می‌سازند. تعداد گره‌های زیرگراف، طبق رابطه (۲) که در مقاله [۱۹] معرفی شده است، به‌دست می‌آید.

$$k = \left\lfloor \frac{2|E|}{|V|} . (1 + \frac{2|E|}{|V|.(|V|-1)}) \right\rfloor \qquad (۲)$$

Algorithm1: *Extract Subgraph* DIS

Input: $G(V,E)$: *input graph*,
$\{A,B\}$: *target nodes pair*,
h: *feature vector of nodes in graph* $\{h(z) \mid z \in V\}$,
k: *the size of subgraph.*

Output: $S_G(A,B)$: *subgraph.*

1: $h(DummyTarget) \leftarrow (h(A) + h(B))/2$
2: **for each** z **in** $V \backslash \{A,B\}$:
3: $\quad d(z) \leftarrow Distance(\mathrm{h}(z), \mathrm{h}(DummyTarget))$
4: $sort\ nodes \leftarrow Sort\ graph\ nodes\ by\ d$
5: $S_G(A,B) \leftarrow sort\ nodes[:k] \cup \{A,B\}$
6: $return\ (S_G(A,B))$

شکل (۴): شبه‌کد روش پیشنهادی استخراج زیرگراف DIS.

در شکل (۴)، شبه‌کد روش استخراج زیرگراف DIS برای جفت گره هدف نشان داده شده است. در این شبه‌کد، ابتدا ورودی‌ها و خروجی‌های تابع نشان داده شده است. ورودی‌های تابع شامل گراف ورودی، جفت گره‌های هدف که با A,B نشان داده شده است، بردار ویژگی گره‌های گراف و تعداد گره‌های زیرگراف (k) و خروجی آن زیرگراف $S_G(A,B)$ است. در خط اول شبه‌کد، بردار ویژگی $DummyTarget$ همانطور که نشان داده شده است (رابطه (۱))، به‌صورت میانگین بردار ویژگی

[1] این بردار ویژگی می‌تواند بردار ویژگی ذاتی گره‌ها، بردار ویژگی ساخته شده برای آن‌ها، بردار امبدینگ و یا هر ترکیبی از بردارهای مختلف باشد.

گره‌های هدف محاسبه می‌شود. در خط سوم، فاصله بین گره‌های گراف و گره $DummyTarget$ برای همه گره‌های گراف به‌جز جفت گره هدف محاسبه و در خط چهارم فاصله گره‌ها به‌صورت صعودی مرتب می‌شود. در خط پنجم، k تا از گره‌های نزدیک‌تر به گره $DummyTarget$ انتخاب و با جفت گره‌های هدف اجتماع شده و زیرگراف S_G استخراج می‌شود.

۲،۱،۴. روش h-hop

روش h-hop، یک روش استخراج زیرگراف مبتنی‌بر همسایگی است. در این روش برای هر گره هدف، گره‌های همسایه تا سطح h استخراج می‌شود. گره‌های استخراج شده (یا k تا از گره‌ها)، گره‌های زیرگراف ($\acute{V}$) و پیوندهای بین آن‌ها پیوندهای زیرگراف ($\acute{E}$) را می‌سازند. چنانچه زیرگراف مبتنی بر دو گره هدف باشد؛ از اجتماع گره‌های همسایه آن‌ها استفاده می‌شود.

Algorithm2: *Extract Subgraph h − hop*

Input: $G(V,E)$: *input graph*,
$\{A,B\}$: *target nodes pair*,
hops: *the size of hops*,
k: *the size of subgraph*.

Output: $S_G(A,B)$: *subgraph*.

1: ***for*** *h* ***in*** *hops*:
2: $subgraph\ nodes = subgraph\ nodes \cup \mathcal{N}_h(A) \cup \mathcal{N}_h(B)$
3: $S_G(A,B) \leftarrow subgraph\ nodes[:k] \cup \{A,B\}$
4: $return\ (S_G(A,B))$

شکل (۵): شبه‌کد روش استخراج زیرگراف h-hop.

در شکل (۵)، شبه‌کد روش استخراج زیرگراف h-hop برای جفت گره هدف نشان داده شده است. در این شبه‌کد، ابتدا ورودی‌ها و خروجی‌های تابع نشان داده شده است. ورودی‌های تابع شامل گراف ورودی، جفت گره‌های هدف برای مثال A,B، تعداد سطح همسایگی (hops) و تعداد گره‌های زیرگراف (k) و خروجی آن زیرگراف $S_G(A,B)$ است. در خط دوم شبه‌کد، اجتماع گره‌های همسایه‌ی گره‌های هدف تا سطح h استخراج می‌شود. در خط سوم، k تا از آن‌ها انتخاب و با جفت گره‌های هدف اجتماع شده و زیرگراف S_G استخراج می‌شود.

۲،۴. روش پیشنهادی ایجاد بردار ویژگی زیرگراف

در این بخش دو روش برای ایجاد بردار ویژگی زیرگراف پیشنهاد می‌شود. در این روش‌ها مانند روش استخراج زیرگراف DIS فرض می‌شود که گره‌های گراف دارای بردار ویژگی هستند. روش‌های پیشنهادی ایجاد بردار ویژگی زیرگراف بر روی زیرگراف‌هایی که دارای تک و یا جفت گره هدف می‌باشند قابل استفاده می‌باشند. در این روش‌ها ابتدا گره $DummyTarget$ و بردار ویژگی آن همانطور که در قسمت قبل توضیح داده شد، مشخص می‌شود؛ سپس با استفاده از روش‌های پیشنهادی بردار ویژگی زیرگراف ایجاد می‌شود. این روش‌ها در ادامه توضیح داده شده است.

۱،۲،۴. روش NDP

در این روش گره‌ها بر اساس فاصله‌ی بردارهای ویژگی آن‌ها با گره‌های هدف جریمه می‌شوند؛ به‌همین دلیل این روش جریمه‌ی فاصله‌ی نرمال یا NDP[1] نام‌گذاری شده است. در این روش به هر گره زیرگراف، براساس فاصله بردار ویژگی آن‌ها با بردار ویژگی گره $DummyTarget$، یک وزن نسبت داده می‌شود؛ که آن را برای گره z به‌صورت $w(z)$ نشان می‌دهیم. روش محاسبه وزن گره‌ها از رابطه خطا AdBoost [۲۱] الهام گرفته شده است. در این روش ابتدا به‌وسیله رابطه (۳)، وزن اولیه هر گره به دست می‌آید؛ به‌طوری که $\acute{w}(z)$ وزن اولیه گره z را نشان می‌دهد.

$$\acute{w}(z) = log((1-d(z))/d(z)) \qquad (۳)$$

بعد از محاسبه وزن اولیه گره‌ها، با استفاده از رابطه (۴) مقادیر وزن‌ها نرمال می‌شوند.

[1] Normal Distance Penalty

$$w(z) = \frac{\acute{w}(z)}{\sum_{i=0}^{k+2} \acute{w}(i)} \quad (۴)$$

بعد از تعیین وزن گره‌های زیرگراف، بردار ویژگی هر گره i در وزن $w(i)$ ضرب شده و در نهایت بردارهای به‌دست آمده با یکدیگر جمع می‌شوند (رابطه (۵)). بردار ویژگی $h(\square_{\square})$ به‌دست آمده، بردار ویژگی زیرگراف است. باید توجه داشته باشیم که k بیانگر تعداد گره‌های زیرگراف بدون در نظر گرفتن گره‌های دوسر پیوند هدف است؛ به‌همین دلیل تعداد کل گره‌های زیرگراف برابر با $k+2$ می‌باشد.

$$h(S_G) = \sum_{i=1}^{k+2} w(i) * h(i) \quad (۵)$$

Algorithm3: *Create a Subgraph Feature Vector NDP*

Input: $S_G(A,B)(\acute{V}, \acute{E})$: *input subgraph*, $\{A,B\}$: *target nodes pair in subgraph*, h: *feature vector of nodes in subgraph* $\{h(z) | z \in \acute{V}\}$.

Output: $h(S_G(A,B))$: *subgraph feature vector.*

$h(DummyTarget) \leftarrow (h(A) + h(B))/2$
for each z in $\acute{V}$:
    $d(z) \leftarrow Distance(h(z), h(DummyTarget))$
$sorted\ nodes \leftarrow Sort\ subgraph\ nodes\ by\ d$
for each z **in** $sorted\ nodes$:
    $\acute{w} \leftarrow \log((1 - d(z))/d(z))$
for each z **in** $sorted\ nodes$:
    $w \leftarrow \frac{\acute{w}(z)}{\sum_{i=0}^{k+2} \acute{w}(i)}$
$h(S_G(A,B)) \leftarrow w.h(sorted\ nodes)$
$return\ (h(S_G(A,B)))$

شکل (۶): شبه‌کد روش پیشنهادی اول ایجاد بردار ویژگی زیرگراف.

در شکل (۶) شبه‌کد روش پیشنهادی ایجاد بردار ویژگی زیرگراف جریمه‌ی فاصله‌ی نرمال (NDP) برای جفت گره نشان داده شده است. در ابتدا ورودی‌ها و خروجی آن را مشخص می‌کند؛ که ورودی‌های آن شامل زیرگراف، جفت گره‌های هدف زیرگراف و بردار ویژگی گره‌های زیرگراف است و خروجی آن بردار ویژگی زیرگراف است که با $h(S_G(A,B))$ نشان داده شده است. در خط اول این شبه‌کد، بردار ویژگی گره $DummyTarget$ محاسبه می‌شود و سپس در خط سوم، فاصله بردار ویژگی تمام گره‌های زیرگراف با گره $DummyTarget$ به‌دست می‌آید. در خط چهارم، گره‌های زیرگراف براساس فاصله با گره $DummyTarget$ به‌صورت صعودی مرتب می‌شوند. در خط ششم، برای هر گره زیرگراف بوسیله رابطه (۳) یک وزن اولیه ایجاد شده و در خط هشتم وزن‌های اولیه با استفاده از رابطه (۴) نرمال می‌شوند. در خط نهم بردار وزن با ابعاد $1 * n$ و ماتریس ویژگی مرتب شده زیرگراف با ابعاد $n * d$ که d بعد بردار ویژگی هر گره را نشان می‌دهد، ضرب شده و بردار ویژگی زیرگراف را با ابعاد $1 * d$ را محاسبه می‌کند و در خط آخر این بردار ویژگی زیرگراف را برمی‌گرداند.

۲،۲،۴. روش CNN

در روش دوم، از شبکه عصبی کانولوشنال (CNN) برای ایجاد بردار ویژگی زیرگراف استفاده می‌شود؛ به‌همین دلیل این روش، روش ایجاد بردار ویژگی زیرگراف مبتنی بر شبکه کانولوشنال (و به اختصار روش CNN) نامیده شده است. در این روش ابتدا گره‌های زیرگراف بر اساس فاصله بردار ویژگی گره‌های زیرگراف با بردار ویژگی گره $DummyTarget$ برچسب‌گذاری می‌شوند؛ این کار برای مشخص شدن ترتیب گره‌های زیرگراف در ماتریس ویژگی زیرگراف که ورودی شبکه عصبی کانولوشنال است بسیار اهمیت دارد. در این روش جفت گره هدف همیشه برچسب‌های ۱ و ۲ می‌گیرد و گره‌های دیگر هرچقدر دورتر باشند، برچسب بزرگتری می‌گیرند. بعد از برچسب‌گذاری، بردار ویژگی گره‌ها با توجه به برچسب‌های مشخص شده، مرتب می‌شوند و ماتریس ویژگی زیرگراف ایجاد می‌شود. این ماتریس به‌عنوان ورودی به شبکه عصبی

کانولوشنال داده می‌شود و خروجی آن یک بردار ویژگی است که بردار ویژگی زیرگراف نامیده می‌شود.

در شکل (۷) شبه‌کد روش ایجاد بردار ویژگی زیرگراف مبتنی بر شبکه کانولوشنال برای جفت گره هدف A, B نشان داده شده است. در ابتدا ورودی‌ها و خروجی آن را مشخص می‌کند؛ که ورودی‌های آن شامل زیرگراف، جفت گره‌های هدف زیرگراف و بردار ویژگی گره‌های زیرگراف است و خروجی آن بردار ویژگی زیرگراف است که با $h(S_G(A,B))$ نشان داده شده است. در خط اول این شبه‌کد، بردار ویژگی گره $DummyTarget$ محاسبه می‌شود و سپس در خط سوم، فاصله بردار ویژگی تمام گره‌های زیرگراف با گره $DummyTarget$ به‌دست می‌آید. در خط چهارم، گره‌های زیرگراف براساس فاصله به‌صورت صعودی مرتب می‌شوند. در خط پنجم ماتریس ویژگی مرتب شده زیرگراف به شبکه عصبی کانولوشنال داده می‌شود، خروجی این شبکه یک بردار ویژگی زیرگراف با ابعاد $1 * d$ است که در خط آخر این بردار ویژگی زیرگراف را برمی‌گرداند.

Algorithm4: *Create a Subgraph Feature Vector based on CNN*

Input: $S_G(A,B)(\acute{V}, \acute{E})$: *input subgraph*, $\{A,B\}$: *target nodes pair in subgraph*, h: *feature vector of nodes in subgraph* $\{h(z) | z \in \acute{V}\}$.

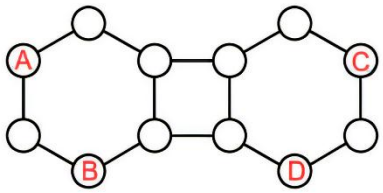
Output: $h(S_G(A,B))$: *subgraph feature vector.*

1: $h(DummyTarget) \leftarrow (h(A) + h(B))/2$
2: ***for each*** z ***in*** $\acute{V}$:
3: $d(z) \leftarrow Distance(\mathrm{h}(z), \mathrm{h}(DummyTarget))$
4: $sorted\ nodes \leftarrow Sort\ subgraph\ nodes\ by\ d$
5: $h(S_G(A,B)) \leftarrow CNN(h(sorted\ nodes))$
6: $return\ (h(S_G(A,B)))$

شکل (۷): شبه‌کد روش پیشنهادی دوم ایجاد بردار ویژگی زیرگراف.

۳،۴. بحث روی روش پیشنهادی

دلیل استفاده از زیرگراف در SGAE، به ضعف روش پایه GAE بازمی‌گردد؛ اگر گره‌های دو سر دو پیوند غیرهمریخت، به‌صورت دو به دو همریخت باشند، بردار ویژگی پیوندها یکسان خواهد شد. در روش GAE تعبیه‌ی گره‌ها یادگرفته می‌شود ولی بردار ویژگی پیوند یاد گرفته نمی‌شود. برای مثال در گراف نشان‌داده شده در شکل (۸) گره‌های C و A و گره‌های B و D با یکدیگر همریخت‌اند به‌همین دلیل تعبیه‌ی این گره‌ها یکسان به‌دست می‌آید. بردار ویژگی پیوند (A,B) از ترکیب تعبیه‌های گره‌های B و A بردار ویژگی پیوند (C,B) از ترکیب تعبیه‌های گره‌های B و C به‌دست می‌آید. در نتیجه به‌دلیل یکسان بودن تعبیه‌های C و A بردار ویژگی پیوندهای (A,B) و (C,B)نیز یکسان خواهد بود؛ در حالی‌که همانطور که در شکل نشان داده شده است، این دو پیوند همریخت نیستند و نباید بردار ویژگی یکسانی داشته باشند. این یکی از ضعف‌های این روش است [۱۰].



شکل (۸): بازنمایی یکسان برای جفت گره‌های (A,B) و (C,B) بدست می‌آید که ضعف روش است [۱۰].

در چارچوب پیشنهادی SGAE برای مرتفع ساختن این ضعف از بردار ویژگی زیرگراف استفاده می‌شود. در این چارچوب همانطور که در شکل (۳) در قسمت رمزگشا نشان داده شده است، برای هر جفت گره هدف (پیوند هدف) یک بردار ویژگی زیرگراف ایجاد می‌شود؛ که به انتهای بردار ویژگی جفت گره هدف (پیوند هدف) الصاق می‌شود. از آنجایی که زیرگراف برای پیوندهای غیرهمریخت متفاوت است، بردار ویژگی آن نیز متفاوت خواهد بود؛ به‌همین دلیل بردار ویژگی

پیوند مبتنی بر زیرگراف ایجاد شده برای پیوندهای غیرهمریخت متفاوت خواهد شد.

پیچیدگی زمانی روش‌های پیشنهادی برای استخراج زیرگراف و ایجاد بردار ویژگی زیرگراف از مرتبه‌ی $O(n)$ است. همانطور که در شکل (۴) و شکل (۵) نشان داده شده است، روش‌های استخراج زیرگراف DIS و h-hop دارای یک حلقه به اندازه‌ی گره‌های گراف است به‌همین دلیل پیچیدگی زمانی این دو روش از مرتبه‌ی $O(n)$ خواهد بود. در روش‌های ایجاد بردار ویژگی زیرگراف NDP و مبتنی بر CNN همانطور که در الگوریتم‌های شکل (۶) و شکل (۷) نشان داده شده است، حلقه‌های تودرتو در این روش‌ها استفاده نشده و پیچیدگی زمانی این دو روش از مرتبه $O(n)$ است.

در چارچوب پیشنهادی به‌دلیل این که برای استخراج زیرگراف و ایجاد بردار ویژگی زیرگراف از روش‌های متفاوتی می‌تواند استفاده شود، پیچیدگی زمانی آن بر اساس روش‌های متفاوت، متغیر خواهد بود. در صورت استفاده از هر کدام از روش‌های پیشنهاد شده برای استخراج زیرگراف و ایجاد بردار ویژگی زیرگراف در چارچوب پیشنهادی SGAE، به این دلیل که پیچیدگی زمانی روش پایه GAE از مرتبه‌ی $O(n)$ است و فرآیند استخراج زیرگراف و ایجاد بردار ویژگی زیرگراف به‌صورت جداگانه و متوالی انجام می‌شود، پیچیدگی زمانی از مرتبه‌ی $O(n)$ خواهد بود.

۵. آزمایش و ارزیابی

در این بخش، سه آزمایش برای ارزیابی کارایی چارچوب پیشنهادی در پیش‌بینی پیوند بر اساس معیارهای رده‌بندی طراحی شده است. این بخش علاوه بر مقایسه میان روش پایه GAE و چارچوب پیشنهادی SGAE، روش‌های پیشنهادی استخراج زیرگراف و ایجاد بردار ویژگی زیرگراف را نیز در چارچوب پیشنهادی مورد آزمایش قرار می‌دهد. قبل از ارائه آزمایش‌ها لازم است تا نامگذاری مناسبی برای روش‌های مورد مقایسه انجام شود. در جدول (۱) این نامگذاری مشخص شده است.

جدول (۱): نامگذاری روش‌های مورد مقایسه.*: روش پیشنهادی مقاله.

نام روش	روش استخراج زیرگراف	روش ایجاد بردار ویژگی
GAE	ندارد	ندارد
SGAE-DIS-NDP	DIS*	NDP*
SGAE-DIS-CNN	DIS*	CNN
SGAE-hhop-NDP	h-hop	NDP*
SGAE-hhop-CNN	h-hop	CNN

۱،۵. مجموعه داده‌ها

در آزمایش‌ها از مجموعه داده گراف‌های همگن Karate، USAir، Power و Yeast استفاده شده است؛ که در بسیاری از پژوهش‌های مرتبط با مسئله‌ی پیش‌بینی پیوند مورد تحقیق قرار گرفته‌اند [۴, ۱۸, ۱۹, ۲۲-۲۴]. تمام گراف‌های مورد آزمایش در این مقاله بدون جهت و بدون وزن در نظر گرفته می‌شوند. مشخصات این گراف‌ها در جدول (۲) نشان داده شده است. تراکم پیوندهای یک گراف به‌صورت نسبت تعداد پیوندهای موجود گراف ($|E|$) به تعداد کل پیوندهایی که در یک گراف بدون جهت می‌تواند ایجاد شود، محاسبه می‌شود. در ادامه به بررسی چگونگی جداسازی مجموعه داده پرداخته شده اشت.

جدول (۲): مشخصات مجموعه داده‌ها

مجموعه داده	گره	پیوند	تعداد گره‌های زیرگراف	تراکم پیوندها
Karate	۳۴	۷۷	۶	۰،۱۳۷۲
USAir	۳۳۲	۲۱۲۶	۱۴	۰،۰۳۸۶
Power	۴۹۴۱	۶۵۹۳	۳	۰،۰۰۰۵
Yeast	۲۳۷۵	۱۱۶۹۳	۱۰	۰،۰۰۴۱

۱،۱،۵. جداسازی مجموعه داده

در آزمایش‌های این مقاله از روش اعتبارسنجی hold out استفاده شده است. برای این کار مجموعه داده به دو دسته‌ی آموزشی و آزمایشی تقسیم می‌شود. در پیش‌بینی با استفاده از شبکه عصبی

گرافی، در هر دسته‌ی آموزشی و آزمایشی، پیوندهای گراف به دو دسته ۱)پیوند پیام[1] ۲) پیوند نظارت[2] تقسیم می‌شوند. پیوند پیام، پیوندهایی است که برای ارسال پیام به شبکه عصبی گرافی نشان داده می‌شوند؛ به این معنی که در زمان آموزش و یا آزمایش مدل پیش‌بینی کننده‌ی پیوند، مدل تنها این پیوندها را مشاهده می‌کند و براساس اطلاعات این پیوندها پیش‌بینی پیوند انجام می‌شود. پیوندهای پیام شامل دو نوع هستند:

- پیوندهای پیام آموزشی: در زمان آموزش مدل پیش‌بینی کننده‌ی پیوند تنها پیوندهای پیام آموزشی را مشاهده می‌کند و با استفاده از آن‌ها سعی می‌کند پیوندهای نظارت آموزشی را پیش‌بینی کند.
- پیوندهای پیام آزمایشی: در زمان آزمایش مدل پیش‌بینی کننده‌ی پیوند تنها پیوندهای پیام آموزشی را مشاهده می‌کند و با استفاده از آن‌ها سعی می‌کند پیوندهای نظارت آموزشی را پیش‌بینی کند. از اجتماع پیوندهای پیام و نظارت آموزشی ایجاد می‌شود.

پیوند نظارت، پیوندهایی است که برای نظارت بر عملکرد مدل پیش‌بینی کننده‌ی پیوند استفاده می‌شوند و به شبکه عصبی گرافی نشان داده نمی‌شوند؛ به این معنی که در زمان آموزش و یا آزمایش مدل پیوندهای نظارت حذف شده و مدل باید آن‌ها را پیش‌بینی کند. این پیوند نیز شامل دو نوع است:

- پیوندهای نظارت آموزشی: در زمان آموزش این پیوندها حذف شده و مدل پیش‌بینی کننده‌ی پیوند آن‌ها را پیش‌بینی می‌کند.
- پیوندهای نظارت آزمایشی: در زمان آزمایش این پیوندها حذف شده و مدل پیش‌بینی کننده‌ی پیوند آن‌ها را پیش‌بینی می‌کند.

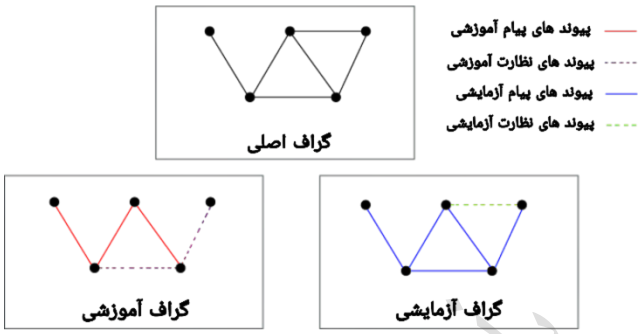
شکل (۹) جداسازی یک گراف را نشان می‌دهد، در بالای این شکل گراف اصلی و در پایین سمت چپ گراف آموزشی با پیوندهای پیام آموزشی (خط قرمز) و پیوندهای نظارت آموزشی (نقطه‌چین بنفش) نشان داده شده است. قسمت سمت راست شکل نیز گراف آزمایشی را با پیوندهای پیام آزمایشی (خط آبی) که از اجتماع پیوندهای پیام و نظارت آموزشی ایجاد شده است و پیوندهای نظارت آزمایشی (نقطه‌چین سبز) نشان می‌دهد. در زمان آموزش مدل، از گراف آموزشی استفاده می‌شود، به‌صورتی که پیوندهای پیام آموزشی (خط قرمز) به مدل نشان داده شده و مدل پیوندهای نظارت آموزشی (نقطه‌چین بنفش) را پیش‌بینی می‌کند. در زمان آزمایش، پیوندهای پیام آزمایشی (خط آبی) که از اجتماع پیوندهای پیام آموزشی و پیوندهای نظارت آموزشی ایجاد می‌شوند، به مدل نشان داده شده و مدل پیوندهای نظارت آزمایشی (نقطه‌چین سبز) را پیش‌بینی می‌کند.

برای جداسازی گراف، ابتدا ۴۶ درصد (این نسبت براساس شبکه ogbl-collab [3] تعیین شده است) پیوندهای گراف را برای پیوندهای پیام آموزشی جدا کرده‌ایم. ۱۰ درصد پیوندهای باقی‌مانده را برای پیوندهای آزمایشی و باقی آن‌ها را برای پیوندهای نظارت آموزشی جدا کرده‌ایم. پیوندهای پیام آزمایشی از اجتماع پیوندهای پیام آموزشی و پیوندهای نظارت آموزشی ایجاد شده است.

[1] Message Edge

[2] Supervision Edge

[3] https://ogb.stanford.edu/



شکل (۹): جداسازی مجموعه داده.

۲,۵. برپاسازی

در پیاده‌سازی روش‌هایی که در جدول (۱) معرفی شده‌اند، شبکه عصبی گرافی استفاده شده دارای یک لایه مخفی با ابعاد ۲۵۶ است. بردارهای ویژگی اولیه گره‌های گراف با روش one-hot ایجاد می‌شود. نرخ یادگیری در این روش‌ها برای گراف Karate، 10^{-4} و در دیگر گراف‌ها 10^{-5} تنظیم شده است. همه مدل‌های یادگیری برای تمام گراف‌ها در ۲۰۰ دوره و با بهینه‌ساز آدام[1] آموزش داده می‌شوند. هر مدل ۱۰ بار اجراشده و نتایج معیارهای ارزیابی آن در نهایت میانگین گرفته می‌شود. در هربار اجرا تقسیم‌بندی مجموعه داده آموزشی و آزمایشی یکسان است اما ترتیب ارائه آن‌ها به مدل به‌صورت تصادفی است. بعضی پارامترهای الگوریتم‌ها براساس پژوهش‌های [۸, ۹] تنظیم شده‌است.

روش‌های پیشنهادی استخراج زیرگراف و ایجاد بردار ویژگی زیرگراف بر پایه فاصله بین بردار ویژگی گره‌های گراف و گره $DummyTarget$ عمل می‌کند؛ در این آزمایش‌ها فاصله از روش شباهت کسینوسی[2] استفاده شده است. در شباهت کسینوسی، زاویه بین دو بردار ویژگی محاسبه می‌شود. در رابطه (۶) روش محاسبه آن به‌وسیله بردار ویژگی گره $DummyTarget$ و بردار ویژگی گره z نشان داده شده است.

$$d_{cos}(z) = \frac{h(DummyTarget).h(z)}{\|h(DummyTarget)\|.\|h(z)\|} \quad (۶)$$

در این آزمایش‌ها از معیارهای ارزیابی F1-score [۲۵]، متوسط صحت[3] [۲۶]، دقت[4] [۲۵] و مساحت زیر منحنی صحت – فراخوانی[5] [۲۵] برای سنجش عملکرد روش‌ها استفاده شده است.

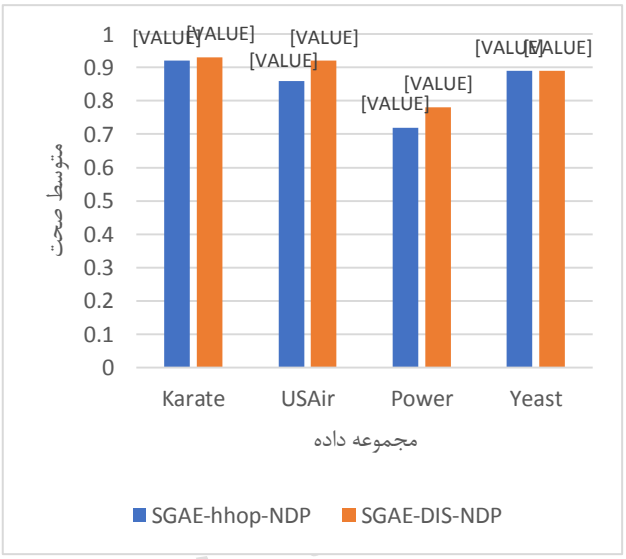
۳,۵. آزمایش اول: استخراج زیرگراف

این آزمایش با هدف مقایسه عملکرد روش استخراج زیرگراف پیشنهاد با روش h-hop در چارچوب پیشنهادی SGAE طراحی شده است. در این آزمایش به‌منظور ارزیابی عملکرد روش‌ها از معیار ارزیابی متوسط صحت استفاده شده است.

برای مقایسه عملکرد این دو روش استخراج زیرگراف، از دو روش SGAE-hhop-NDP و SGAE-DIS-NDP استفاده شده است. این مدل روشی برای پیش‌بینی پیوند بر پایه‌ی چارچوب SGAE است که از روش ایجاد بردار ویژگی جریمه‌ی فاصله‌ی نرمال در آن استفاده شده است. به‌منظور مقایسه دو روش استخراج زیرگراف، استخراج زیرگراف یک بار از روش NDP و بار دیگر از روش h-hop استفاده شده است.

نتایج این آزمایش به‌صورت نمودار میله‌ای در شکل (۱۰) نشان داده شده است. همانطور که مشاهده می‌شود، میانگین صحت روش SGAE-DIS-NDP که از روش پیشنهادی استخراج زیرگراف DIS استفاده می‌کند، در سه مجموعه داده‌ی Karate، USAir و Power بیشتر است. به‌طور میانگین، روش SGAE-DIS-NDP ۳,۲۵٪ بهتر از روش SGAE-hhop-NDP عمل کرده است.

[1] Adam
[2] Cosine Similarity
[3] Average Precision
[4] Accuracy
[5] Precision-Recall Area Under Curve



شکل (۱۰): نمودار متوسط صحت برای دو روش استخراج زیرگراف DIS و h-hop در چارچوب پیشنهادی SGAE.

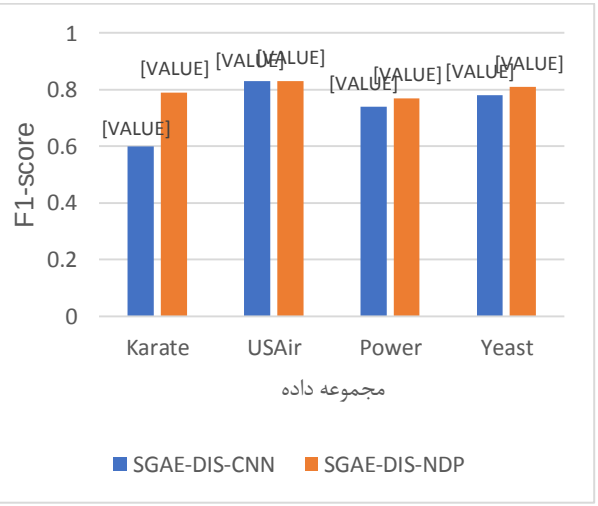
۴.۵. آزمایش دوم: ایجاد بردار ویژگی

این آزمایش با هدف ارزیابی دو روش پیشنهادی ایجاد بردار ویژگی زیرگراف در چارچوب پیشنهادی SGAE طراحی شده است. در این آزمایش از معیار ارزیابی F1-Score برای ارزیابی روش‌ها استفاده شده است.

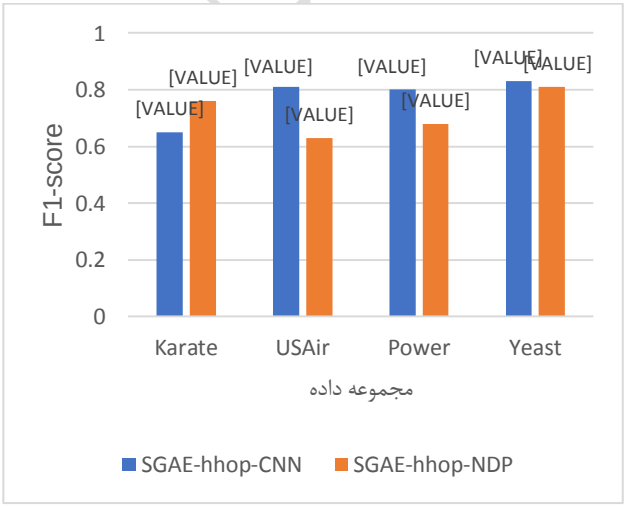
برای مقایسه عملکرد این دو روش استخراج بردار ویژگی زیرگراف، دو دسته مقایسه صورت گرفته است. در دسته اول روش SGAE-DIS-NDP با روش SGAE-DIS-CNN (شکل (۱۱)) و در دسته دوم روش SGAE-hhop-NDP با روش SGAE-hhop-CNN (شکل (۱۲)) مقایسه می‌شود. در دسته اول مقایسه‌ها برای استخراج زیرگراف از روش DIS و در دسته دوم از روش h-hop استفاده شده است. نتایج این آزمایش به‌صورت نمودار میله‌ای در شکل (۱۱) و شکل (۱۲) نشان داده شده است.

همانطور که در شکل (۱۱) مشاهده می‌شود، معیار F1-score برای روش SGAE-DIS-NDP در تمام مجموعه داده‌های مورد آزمایش نتیجه‌ی بهتری را نشان می‌دهد؛ اما اختلاف نتایج آن با روش SGAE-DIS-CNN در تمام مجموعه داده‌ها به‌جز Karate کم است. به‌طور میانگین، روش SGAE-DIS-NDP ۶,۲۵٪ بهتر از روش SGAE-DIS-CNN عمل کرده است.

در شکل (۱۲) معیار F1-score در اکثر مجموعه داده‌ها روش SGAE-hhop-CNN مقدار بیشتری را نشان می‌دهد و اختلاف آن با روش SGAE-hhop-NDP تقریبا زیاد است. به‌طور میانگین، روش SGAE-hhop-CNN ۵,۲۵٪ بهتر از روش SGAE-hhop-NDP عمل کرده است.



شکل (۱۱): نتایج مقایسه‌ی اول در آزمایش دوم.



شکل (۱۲): نتایج مقایسه‌ی دوم در آزمایش دوم.

از تحلیل این دو دسته مقایسه می‌توان نتیجه گرفت که روش ایجاد بردار ویژگی NDP در صورتی که در چارچوب برای استخراج زیرگراف از یک روش متناسب با این روش مانند روش DIS استفاده شود، بهتر از شبکه عصبی کانولوشنال عمل می‌کند. اما شبکه عصبی کانولوشنال چون یک روش یادگیری است، به روش استخراج زیرگراف وابسته نیست و در اکثر حالت‌ها عملکرد قابل قبولی دارد.

۵،۵. آزمایش سوم

این آزمایش با هدف ارزیابی چارچوب پیشنهادی SGAE با روش پایه GAE طراحی شده است. در این آزمایش دو دسته مقایسه صورت می‌گیرد. در دسته‌ی اول مقایسه‌ها از معیارهای ارزیابی دقت، F1-score و متوسط صحت و در دسته‌ی دوم از معیار ارزیابی مساحت زیر منحنی صحت – فراخوانی استفاده شده است.

همانطور که در آزمایش دوم مشاهده شد، روش‌های SGAE-DIS-NDP و SGAE-hhop-CNN طبق معیار F1-score عملکرد بهتری نشان داده‌اند؛ به‌همین دلیل در این آزمایش برای مقایسه چارچوب پیشنهادی SGAE با روش پایه GAE، از این دو روش به نمایندگی چارچوب SGAE استفاده شده است.

نتیجه‌ی مقایسه اول در جدول (۳) نشان داده شده است. به‌دلیل این که آزمایش‌ها ۱۰ بار انجام شده و نتایج میانگین گرفته شده‌اند، در این جدول انحراف معیار (به‌دلیل این که واریانس نتایج بسیار کوچک است از انحراف معیار استفاده شده است.) نتایج به‌صورت ± نیز نشان داده شده‌اند. همانطور که مشاهده می‌شود در هر چهار مجموعه داده، دو روش SGAE-hhop-CNN و SGAE-DIS-NDP عملکرد بهتری از روش پایه دارند.

در مقایسه دوم، روش‌های SGAE-hhop-CNN و SGAE-DIS-NDP بر اساس معیار ارزیابی مساحت زیر منحنی صحت–فراخوانی با روش GAE مقایسه می‌شوند. شکل (۱۳) منحنی‌های این سه روش را برای بهترین خروجی در ۱۰ اجرا، به تفکیک مجموعه داده نشان داده است.

در **Error! Reference source not found.** مقدار مساحت زیر منحنی صحت–فراخوانی روش‌ها، برای هر گراف نشان داده شده است. همانطور که در این جدول مشاهده می‌شود، مساحت زیر منحنی در همه‌ی مجموعه داده‌ها برای روش‌های مبتنی بر چارچوب پیشنهادی SGAE بیشتر از روش پایه GAE است. در مجموعه داده‌های Karate، Yeast و Power روش SGAE-hhop-CNN و در مجموعه داده USAir روش SGAE-DIS-NDP مقدار بیشتری را نشان می‌دهند. این نتایج نشان دهنده‌ی برتری چارچوب پیشنهادی بر روش پایه و تاثیر مثبت مفهوم زیرگراف بر روش پایه است.

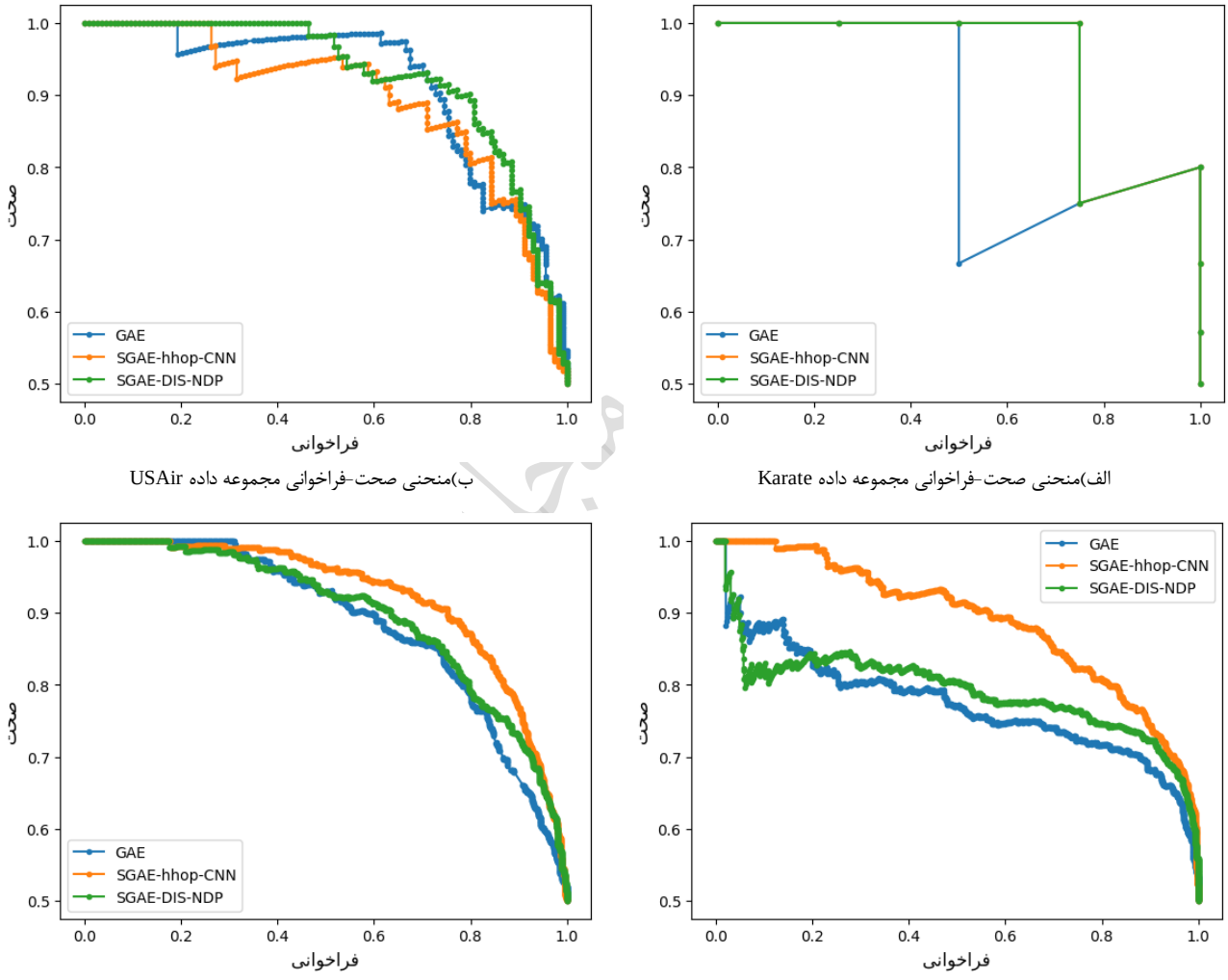
جدول (۳): نتایج مقایسه‌ی اول در آزمایش سوم.

مجموعه داده	مدل	دقت	F1-Score	متوسط صحت
Karate	GAE	۰±۰،۷۵	۰±۰،۷۵	۰،۰۲۷±۰،۸۵
	SGAE-hhop-CNN	۰،۱۰۵±۰،۷	۰،۱۱۷±۰،۶۵	۰،۰۴۴±۰،۹
	SGAE-DIS-NDP	**۰،۱۲۰±۰،۸۲**	**۰،۱۶±۰،۷۹**	**۰،۰۴۶±۰،۹۳**
USAir	GAE	۰،۰۱۶±۰،۷۹	۰،۰۱۹±۰،۸۰	۰،۰۱۲±۰،۹۱
	SGAE-hhop-CNN	۰،۰۱۱±۰،۸۱	۰،۰۱۶±۰،۸۱	۰،۰۰۲±۰،۸۹
	SGAE-DIS-NDP	**۰،۰۰۷±۰،۸۴**	**۰،۰۰۹±۰،۸۳**	**۰،۰۰۱±۰،۹۲**
Power	GAE	۰،۰۰۶±۰،۷۳	۰،۰۱۰±۰،۷۳	۰،۰۰۵±۰،۷۶
	SGAE-hhop-CNN	**۰،۰۱۲±۰،۷۸**	**۰،۰۱۰±۰،۸۰**	**۰،۰۲۷±۰،۸۵**
	SGAE-DIS-NDP	۰،۰۵۴±۰،۷۶	۰،۰۵۸±۰،۷۷	۰،۰۳۴±۰،۷۸

	GAE	۰,۰۰۹±۰,۷۸	۰,۰۱۳±۰,۷۷	۰,۰۰۹±۰,۸۷
Yeast	SGAE-hhop-CNN	**۰,۰۰۴±۰,۸۳**	۰,۰۰۴±**۰۸۳**	۰,۰۰۲±**۰۹۲**
	SGAE-DIS-NDP	۰,۰۰۴±۰,۸۰	۰,۰۰۴±۰,۸۱	۰,۰۰۲±۰,۸۹

جدول (۴): مساحت زیر منحنی صحت-فراخوانی.

مدل	Karate	USAir	Power	Yeast
GAE	۰,۸۷۰۸	۰,۹۱۶۰	۰,۷۷۴۳	۰,۸۸۵۷
SGAE-hhop-CNN	**۰,۹۴۳۸**	۰,۸۹۴۵	**۰,۸۸۹۷**	**۰,۹۲۲۳**
SGAE-DIS-NDP	**۰,۹۴۳۸**	**۰,۹۲۵۸**	۰,۷۹۱۸	۰,۹۰۴۱



الف)منحنی صحت-فراخوانی مجموعه داده Karate

ب)منحنی صحت-فراخوانی مجموعه داده USAir

ج)منحنی صحت-فراخوانی مجموعه داده Power

د)منحنی صحت-فراخوانی مجموعه داده Yeast

شکل (۱۳): منحنی صحت-فراخوانی.

۶,۵. آزمایش چهارم

در این آزمایش به بررسی چند روش دیگر پیش‌بینی پیوند مبتنی بر GNN و مقایسه عملکرد آن‌ها با چارچوب پیشنهادی SGAE پرداخته شده است. روش‌های مورد بررسی شامل GraphLP [۲۷] و GAT [۲۸] است.

روش GraphLP، از توانایی یادگیری ویژگی شبکه عصبی گرافی برای استخراج خودکار الگوهای ساختاری گراف‌ها و

پیش‌بینی پیوند استفاده می‌کند [۲۷]. روش GAT، برای هر گره را با ترکیب بردارهای همسایگی آن به روشی تطبیقی با وزن توجه قابل تنظیم برای همسایگی‌های مختلف یک تعبیه استخراج می‌کند و باتوجه به تعبیه‌ی استخراج شده برای هر جفت گره هدف، پیش‌بینی پیوند انجام می‌شود [۲۸].

این آزمایش روی دو مجموعه داده‌ی USAir و Yeast براساس معیار صحت انجام شده و نتایج آن در جدول (۵) نشان داده شده است. همانطور که مشاهده می‌شود در مجموعه داده USAir روش SGAE-hhop-CNN و در مجموعه داده Yeast روش GAT، صحت بیشتری دارد.

جدول (۵): نتایج آزمایش چهارم.

مدل	USAir	Yeast
GraphLP	۰,۸۲	۰,۷۴
GAT	۰,۸۸	**۰,۹۱**
GAE	۰,۸۱	۰,۸۱
SGAE-hhop-CNN	**۰,۹۱**	۰,۸۹

۶. نتیجه‌گیری

در این مقاله یک چارچوب برای حل مسئله پیش‌بینی پیوند با نام SGAE پیشنهاد شد؛ که از شبکه عصبی گرافی، زیرگراف و بردار ویژگی زیرگراف در آن استفاده شده است. علاوه بر چارچوب SGAE روش‌هایی برای استخراج زیرگراف و ایجاد بردار ویژگی زیرگراف نیز پیشنهاد شده است که در چارچوب SGAE قابل استفاده است. روش‌های استخراج زیرگراف و ایجاد بردار ویژگی زیرگراف در چارچوب پیشنهادی مورد بررسی قرار گرفت. نتایج بررسی نشان داد که چارچوب پیشنهادی SGAE به‌طور میانگین، در معیارهای ارزیابی دقت، F1-score، متوسط صحت و مساحت زیر منحنی صحت-فراخوانی، به‌ترتیب ۵,۵، ۵، ۵,۷۵ و ۵,۸۷ درصد بهتر از روش پایه GAE عمل می‌کند.

مراجع

[1] A. Kumar, S. S. Singh, K. Singh, and B. Biswas, "Link prediction techniques, applications, and performance: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 553, p. 124289, 2020, doi: 10.1016/j.physa.2020.124289.

[2] V. Martínez, F. Berzal, and J.-C. Cubero, "A survey of link prediction in complex networks," *ACM computing surveys (CSUR)*, vol. 49, no. 4, pp. 1-33, 2016, doi: 10.1145/3012704.

[3] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: statistical mechanics and its applications*, vol. 390, no. 6, pp. 1150-1170, 2011, doi: 10.1016/j.physa.2010.11.027.

[4] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," *Advances in neural information processing systems*, vol. 31, pp. 5171-5181, 2018, doi: 10.48550/arXiv.1802.09691.

[5] Y. Han, D. Guan, and W. Yuan, "Learning Subgraph Structure with LSTM for Complex Network Link Prediction," in *Advanced Data Mining and Applications: 15th International Conference, ADMA 2019, Dalian, China, November 21–23, 2019, Proceedings 15*, 2019, pp. 34-47, doi: 10.1007/978-3-030-35231-8_3.

[6] A. Saxena, G. Fletcher, and M. Pechenizkiy, "NodeSim: node similarity based network embedding for diverse link prediction," *EPJ Data Science*, vol. 11, no. 1, p. 24, 2022, doi: 10.1140/epjds/s13688-022-00336-8.

[7] J. Feng and S. Chen, "Link prediction based on orbit counting and graph auto-encoder," *IEEE Access*, vol. 8, pp. 226773-226783, 2020, doi: 10.1109/ACCESS.2020.3045529.

[8] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016, doi: 10.48550/arXiv.1609.02907.

[9] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016, doi: 10.48550/arXiv.1611.07308.

[10] M. Zhang, P. Li, Y. Xia, K. Wang, and L. Jin, "Labeling trick: A theory of using graph neural networks for multi-node representation learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 9061-9073, 2021.

[11] K. Teru, E. Denis, and W. Hamilton, "Inductive relation prediction by subgraph reasoning," in *International Conference on Machine Learning*, 2020: PMLR, pp. 9448-9457, doi: 10.48550/arXiv.1911.06962.

[12] 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, 1998, doi: 10.1109/5.726791.

[13] L. Li, S. Fang, S. Bai, S. Xu, J. Cheng, and X. Chen, "Effective Link Prediction Based on Community Relationship Strength," *IEEE Access*, vol. 7, pp. 43233-43248, 2019, doi: 10.1109/access.2019.2908208.

[14] D. Jin *et al.*, "A survey of community detection approaches: From statistical modeling to deep learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, pp. 1149 - 1170, 2021, doi: 10.1109/TKDE.2021.3104155.

[15] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings. 2005 IEEE international joint conference on neural networks*, 2005, vol. 2: IEEE pp. 729-734, doi: 10.1109/IJCNN.2005.1555942.

[16] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans Neural Netw*, vol. 20, no. 1, pp. 61-80, Jan 2009, doi: 10.1109/TNN.2008.2005605.

[17] C. Gallicchio and A. Micheli, "Graph echo state networks," in *The 2010 international joint conference on neural networks (IJCNN)*, 2010: IEEE, pp. 1-8, doi: 10.1109/IJCNN.2010.5596796.

[18] M. Zhang and Y. Chen, "Weisfeiler-Lehman Neural Machine for Link Prediction," presented at the Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017.

[19] K. Ragunathan, K. Selvarajah, and Z. Kobti, "Link prediction by analyzing common neighbors based subgraphs using convolutional neural network," in *ECAI 2020*: IOS Press, 2020, pp. 1906-1913.

[20] K. Selvarajah, K. Ragunathan, Z. Kobti, and M. Kargar, "Dynamic Network Link Prediction by Learning Effective Subgraphs using CNN-LSTM," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1-8, doi: 10.1109/IJCNN48605.2020.9207301.

[21] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119-139, 1997, doi: 10.1006/jcss.1997.1504.

[22] Y. Wang and L. Ming, "Global Path Link Prediction Method Based on Improved Resource Allocation," in *Journal of Physics: Conference Series*, 2023, vol. 2522, no. 1: IOP Publishing, p. 012023, doi: 10.1088/1742-6596/2522/1/012023.

[23] J. Zhu *et al.*, "SpotTarget: Rethinking the Effect of Target Edges for Link Prediction in Graph Neural Networks," *arXiv preprint arXiv:2306.00899*, 2023, doi: 10.48550/arXiv.2306.00899.

[24] N. K. Ahmed, J. Neville, R. A. Rossi, and N. Duffield, "Efficient graphlet counting for large networks," in *2015 IEEE International Conference on Data Mining*, 2015: IEEE, pp. 1-10, doi: 10.1109/ICDM.2015.141.

[25] K. Abbas *et al.*, "Application of network link prediction in drug discovery," *BMC Bioinformatics*, vol. 22, no. 1, p. 187, Apr 12 2021, doi: 10.1186/s12859-021-04082-y.

[26] S. Scellato, A. Noulas, and C. Mascolo, "Exploiting place features in link prediction on location-based social networks," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 1046-1054, doi: 10.1145/2020408.2020575.

[27] X. Xian *et al.*, "Generative Graph Neural Networks for Link Prediction," *arXiv preprint arXiv:2301.00169*, 2022, doi: 10.48550/arXiv.2301.00169.

[28] W. Gu, F. Gao, X. Lou, and J. Zhang, "Link prediction via graph attention network," *arXiv preprint arXiv:1910.04807*, 2019, doi: 10.48550/arXiv.1910.04807.