Les erreurs de reconstruction : une explication simple et efficace pour les auto-encodeurs de graphe utilisés pour la détection d'anomalies

Bastien Giles*,**, Baptiste Jeudy*
Christine Largeron*, Damien Saboul**

*Université Jean Monnet Saint-Etienne, CNRS,
Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France
**Be-ys Research

Résumé. Les auto-encodeurs de graphes (Graph Auto-Encoders ou GAEs) ont fait preuve d'une efficacité remarquable pour la détection d'anomalies dans des graphes. Cependant, leur nature de "boîte noire" ne permet pas de comprendre les raisons qui les ont conduits à classer un nœud comme anormal. De plus, alors qu'avec le développement de l'XAI, de nombreuses méthodes ont été proposées pour fournir des explications pour différents modèles d'apprentissage profond, il y a une absence notable de cadre d'évaluation dédié à la détection d'anomalies dans les graphes. Notre contribution comble cette lacune en adaptant des cadres d'évaluation existants aux défis spécifiques de la détection d'anomalies à l'aide de GAEs. De plus, elle introduit une technique d'explication simple mais efficace basée sur les erreurs de reconstruction des GAEs. En utilisant ce nouveau cadre, nous évaluons l'efficacité de différents explainers et montrons expérimentalement que la méthode que nous proposons, basée sur les erreurs de reconstruction, surpasse les autres explainers pour les GAEs.

1 Introduction

L'identification des anomalies dans des données tabulaires a été bien étudié dans la littérature (Aggarwal (2017); Akoglu (2021); Chalapathy et al. (2018); Grubbs (1969); Pang et al. (2021)) et plus récemment sur des graphes (Ma et al. (2021)) ou des graphes attribués qui incluent un vecteur d'attributs décrivant chaque nœud (Interdonato et al. (2019)). Parmi les approches non supervisées de détection des anomalies, les auto-encodeurs de graphes (GAE) ont émergé comme des modèles performants (Ding et al. (2019); Fan et al. (2020); Bandyopadhyay et al. (2020)). Ils utilisent des réseaux de neurones pour graphes (GNN) pour produire des plongements (embeddings) des noeuds dans un espace réel de faible dimension, puis reconstruisent le graphe ; ce qui permet d'identifier les nœuds mal reconstruits comme étant des anomalies. Ces GAEs sont particulièrement intéressants en raison de leur caractère non supervisé et du manque fréquent de données étiquetées, mais leur nature de "boîte noire" limite leur utilisation pratique. En effet, lorsqu'un nœud est classé comme anormal, les raisons de ce classement ne sont pas fournies ; ce qui n'est pas acceptable dans certains contextes comme

la détection de fraudes. Plusieurs méthodes dites "explainers" ont été développées pour expliquer les décisions des GNNs, mais elles n'ont pas encore été testées sur les GAEs, et aucun benchmark n'existe pour comparer ces techniques dans le cadre de la détection d'anomalies.

Pour pallier cette lacune, nous avons adapté un cadre d'évaluation initialement conçu pour la classification de nœuds (Agarwal et al. (2023)) afin d'évaluer l'explicabilité des anomalies détectées par les GAEs. Nous montrons également que l'erreur de reconstruction, utilisée pour détecter les anomalies, peut aussi servir d'explication, surpassant souvent les explainers existants. En résumé, nos contributions sont les suivantes :

1. Nous introduisons une méthode simple mais efficace, basée sur les erreurs de reconstruction, pour expliquer les décisions des auto-encodeurs de graphes. Son objectif est de mettre en lumière les attributs et arêtes contribuant à la classification d'un nœud comme anormal.

2. Nous adaptons un cadre d'évaluation initialement conçu pour la classification de nœuds aux besoins spécifiques de la tâche de détection d'anomalies. En effet un modèle de classification fournit en sortie la classe du noeud ou sa probabilité d'appartenance aux classes alors que dans le cadre de la détection d'anomalie, il s'agit d'un score d'anormalité ; ce qui nécessite de modifier les métriques d'évaluation. À l'aide de ce cadre, nous évaluons expérimentalement notre méthode d'explication et montrons qu'elle obtient des performances supérieures aux techniques d'explication existantes.

2 Travaux connexes

Détection d'anomalies. La détection d'anomalies a été largement étudiée avec des méthodes classiques basées sur la distance, la densité ou le clustering, mais ces approches ne tiennent pas compte des structures de graphe (Aggarwal (2017); Chandola et al. (2009)). Récemment, les réseaux de neurones pour graphes (GNN) ont été proposés pour exploiter à la fois les attributs des nœuds et la structure du graphe. Les modèles comme GCN (Kipf et Welling (2017)), GraphSage (Veličković et al. (2018)), GIN (Xu et al. (2019)), et SGC (Wu et al. (2019)) sont désormais des standards dans les tâches de détection d'anomalies car ils permettent de produire des plongements de nœuds combinant les matrices d'attributs et d'adjacence (Liu et al. (2022)). Les détecteurs d'anomalies traditionnels peuvent ensuite être appliqués à ces plongements, comme celui de Kumagai et al. (2021) où la distance du nœud au centre d'une hypersphère détermine son score d'anomalie. Il existe aussi des GNN spécifiques, comme BWGNN proposé par Tang et al. (2022), adaptés à cette tâche. Mais ce sont surtout les méthodes basées sur la reconstruction, telles que les auto-encodeurs de graphes (GAE, Ding et al. (2019); Fan et al. (2020)) qui sont les plus utilisées. Leur principe consiste à compresser le graphe avec un encodeur, à le reconstruire à l'aide d'un décodeur et à calculer ensuite un score d'anomalie basé sur l'erreur de reconstruction. Dominant introduit par (Ding et al. (2019)), un des premiers GAEs, utilise un GCN comme encodeur et deux décodeurs distincts pour reconstruire les matrices d'adjacence et d'attributs. Depuis, plusieurs variantes ont été développées, comme AnomalyDAE (Fan et al. (2020)) ou Suspicious (Giles et al. (2023)), faisant des GAEs l'état de l'art pour la détection d'anomalies non supervisée.

Expliquer les GNN. Les techniques qui ont été proposées pour expliquer les modèles d'apprentissage automatique peuvent être classées en plusieurs familles selon qu'elles fournissent une explication locale (prédiction individuelle) ou globale (comportement du modèle) (Molnar (2022)). L'explication peut être post hoc si une méthode externe génère l'explication après la classification, ou ante hoc si le modèle produit l'explication pendant la classification. Les explainers pour les GNN sont principalement locaux et post hoc (Ying et al. (2019); Simonyan et al. (2014); Yuan et al. (2021); Baldassarre et Azizpour (2019)). Ils diffèrent aussi selon les éléments du graphe pris en compte (nœuds $\mathcal{V}$, arêtes $\mathcal{E}$, attributs $\mathbf{X}$) et selon la méthode utilisée pour générer l'explication. Les méthodes basées sur les gradients, comme Gradients (GradEx, Simonyan et al. (2014)), Integrated Gradients (IGEx, Simonyan et al. (2014)) ou GuidedBP (Baldassarre et Azizpour (2019)), calculent les gradients de la prédiction par rapport aux attributs d'entrée, révélant comment les modifications des attributs influencent la sortie du modèle. Ces méthodes sont dites "dépendantes du modèle" (model-aware) car elles nécessitent l'accès aux paramètres internes pour calculer les gradients. En revanche, les méthodes basées sur la perturbation, telles que SubgraphX (SubGx, Yuan et al. (2021)) et GNNExplainer (GNNEx, Ying et al. (2019)), modifient légèrement les données d'entrée et observent leur impact sur la sortie. Ces approches, "indépendantes du modèle" (model-agnostique), peuvent être appliquées à tout modèle de classification.

Explainer	**Model-aware/ agnostique**	**Explication**	**Méthode**
IGEx	Model-aware	$\mathcal{V}/\mathbf{X}$	Gradient
GradEx	Model-aware	$\mathcal{V}/\mathbf{X}$	Gradient
GuidedBP	Model-aware	$\mathcal{V}/\mathbf{X}$	Gradient
GNNEx	Agnostique	$\mathcal{V}/\mathcal{E}/\mathbf{X}$	Perturbation
SubGx	Agnostique	$\mathcal{V}/\mathcal{E}$	Perturbation

TAB. 1 – *Caractéristiques de certains explainers.*

Expliquer les auto-encodeurs. Dans les auto-encodeurs tabulaires classiques, la détection des anomalies repose souvent sur les erreurs de reconstruction. Cette erreur, qui reflète la capacité d'un auto-encodeur à reproduire les données d'entrée, sert intuitivement de base pour identifier les anomalies : les points de données avec des erreurs de reconstruction élevées sont considérés comme des anomalies potentielles. Les études sur l'utilisation de l'erreur de reconstruction comme technique d'explication sont encore rares (Ravi et al. (2021); Charte et al. (2020)). Quelques travaux industriels (Assaf et al. (2021); Gorman et al. (2023)) ont utilisé intuitivement les erreurs de reconstruction pour fournir des explications mais ces études rapportent souvent les erreurs de reconstruction comme des explications fournies aux experts humains pour guider leur analyse. Ainsi, ces approches n'ont jamais été évaluées dans le cadre d'une expérimentation rigoureuse conduite avec des métriques établies et une analyse comparative avec d'autres méthodes d'explication.

Évaluation des explications. Récemment, Amara et al. (2022) ont défini des protocoles d'évaluation pour comparer les explainers dans la classification de nœuds sur des graphes non

attribués. Ils ont également proposé de nouvelles métriques permettant de comparer les explainers sans nécessité de vérité terrain synthétique. Agarwal et al. (2023) ont ensuite étendu ce protocole à la classification de nœuds sur des graphes attribués en introduisant ShapeGGen, un générateur de graphes synthétiques qui produit des graphes avec des anomalies ainsi que des vérités terrain pour les explications de ces anomalies. Mais à notre connaissance, il n'existe pas de cadre d'évaluation pour l'explication de la détection d'anomalies dans des graphes. Cette contribution vise à combler ce manque.

3 Définitions et formalisation du problème

Soit $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ un graphe attribué, défini par un ensemble de nœuds $\mathcal{V} = \{v_1, \ldots, v_n\}$, un ensemble d'arêtes $\mathcal{E}$ représenté par une matrice d'adjacence symétrique $\mathbf{A} = (a_{i,j}) \in \{0,1\}^{n \times n}$, où $a_{i,j} = 1$ s'il existe une arête entre les nœuds i et j, et $a_{i,j} = 0$ sinon, ainsi qu'une matrice d'attributs $\mathbf{X} = (x_{i,j}) \in \mathbb{R}^{n \times d}$ dont la i-ème ligne, $\mathbf{x}_i$, représente le vecteur d'attributs du nœud v_i. De la même manière, $\mathbf{a}_i$ désigne la i-ème ligne de $\mathbf{A}$. Dans la suite, les lettres majuscules en gras désignent des matrices, et les lettres minuscules en gras désignent des vecteurs. Étant donné un modèle f qui produit une prédiction $f(v_i)$ pour un nœud v_i, la sortie peut prendre différentes formes (classe ou probabilité) selon le type de modèle f. De manière générale, une explication de cette prédiction est un vecteur $\text{Imp}(f, v_i)$ qui attribue un poids d'importance à chaque nœud, arête et attribut du graphe. Les éléments ayant les poids les plus élevés sont considérés comme les plus importants pour la prédiction $f(v_i)$. Cependant, ces poids d'importance pour $f(v_i)$ sont souvent limités aux éléments (arêtes, nœuds et leurs attributs) situés dans le sous-graphe de voisinage à h sauts du nœud v_i. Ce sous-graphe, noté :

$$\text{SUB}(v_i, h) = (\mathcal{V}_{\text{SUB}}(v_i, h), \mathcal{E}_{\text{SUB}}(v_i, h), X_{\text{SUB}}(v_i, h)), \tag{1}$$

désigne le sous-graphe induit par tous les nœuds situés à une distance inférieure ou égale à h de v_i. De plus, les poids d'importance sont divisés en trois vecteurs : $\text{Imp}_{\mathcal{V}}(f, v_i)$ contient un poids d'importance pour chaque nœud de $\text{SUB}(v_i, h)$, $\text{Imp}_{\mathcal{E}}(f, v_i)$ pour les arêtes de $\text{SUB}(v_i, h)$, et $\text{Imp}_{\mathbf{X}}(f, v_i)$ pour les attributs du nœud v_i. Pour simplifier les notations, nous omettons h et f dans le reste de cet article.

Un auto-encodeur de graphe (GAE) prend en entrée les matrices d'attributs et d'adjacence $\mathbf{X}$ et $\mathbf{A}$ d'un graphe attribué G, les compresse en un embedding, puis tente de les reconstruire, produisant les matrices reconstruites $\hat{\mathbf{X}}$ et $\hat{\mathbf{A}}$. L'erreur de reconstruction de chaque nœud v_i est calculée par :

$$error(v_i) = (1 - \alpha)\|\mathbf{a}_i - \hat{\mathbf{a}}_i\|^2 + \alpha\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2. \tag{2}$$

Les nœuds présentant une erreur de reconstruction élevée sont considérés comme des anomalies. Le problème que nous cherchons à résoudre est de produire une explication pour chaque nœud identifié comme anormal par un auto-encodeur de graphe f donné.

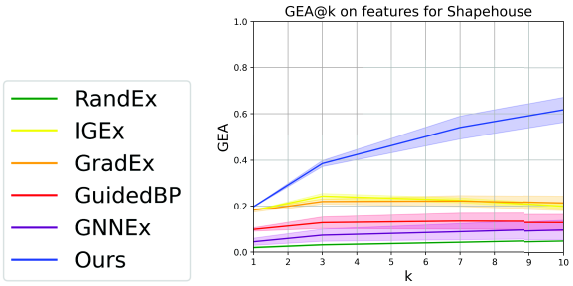

FIG. 1 – *Moyenne (+/- écart-type) de GEA@k sur les attributs de ShapeHouse.*

4 Générer des explications à partir des erreurs de reconstruction des auto-encodeurs de graphe

Nous faisons l'hypothèse que l'erreur de reconstruction commise sur chaque élément (noeud, arête, et attribut) est déjà une bonne approximation de sa contribution à la classification et fournit ainsi une explication pour les anomalies.

Vecteurs d'importance : Nous proposons une méthode pour extraire cette explication à partir de l'erreur de reconstruction. A partir des matrices d'attributs et d'adjacence initiales $\mathbf{X}$ et $\mathbf{A}$ et celles reconstruites $\hat{\mathbf{X}}$ et $\hat{\mathbf{A}}$, nous générons d'abord les matrices d'erreurs $\mathbf{A}' = (a'_{i,j})$ et $\mathbf{X}' = (x'_{i,j})$, définies par :

$$a'_{i,j} = (a_{i,j} - \hat{a}_{i,j})^2,\ x'_{i,j} = (x_{i,j} - \hat{x}_{i,j})^2. \tag{3}$$

Les vecteurs de reconstruction $\mathbf{x}'_i$ et $\mathbf{a}'_i$ peuvent alors être utilisés pour créer des vecteurs d'importance, car ils représentent la contribution de chaque composant à la classification finale. Nous calculons ainsi le vecteur d'importance des attributs $\text{Imp}_{\mathbf{X}}$ et celui des arêtes $\text{Imp}_{\mathcal{E}}$ comme suit :

$$\text{Imp}_{\mathbf{X}}(v_i) = \mathbf{x}'_i,\ \text{Imp}_{\mathcal{E}}(v_i) = (a'_{j,k}, (v_j, v_k) \in \mathcal{E}_{\text{SUB}}(v_i)), \tag{4}$$

avec $\text{Imp}_{\mathbf{X}}(v_i) \in \mathbb{R}^d$ et $\text{Imp}_{\mathcal{E}}(v_i) \in \mathbb{R}^{|\mathcal{E}_{\text{SUB}}(v_i)|}$.

Des vecteurs d'importance aux explications : Pour calculer la métrique d'évaluation GEA (Graph Explanation Accuracy, Agarwal et al. (2023)) qui sera définie ultérieurement, il est nécessaire de transformer les vecteurs d'importance en vecteurs d'explication binaires, tels que $\text{Exp}_{\mathcal{V}}(v_i) \in \{0,1\}^{|\mathcal{V}_{\text{SUB}}(v_i)|}$, $\text{Exp}_{\mathcal{E}}(v_i) \in \{0,1\}^{|\mathcal{E}_{\text{SUB}}(v_i)|}$, et $\text{Exp}_{\mathbf{X}}(v_i) \in \{0,1\}^d$.

Nous proposons d'adopter la fonction top$k$, comme suggéré par Amara et al. (2022). Cette fonction sélectionne les indices des k plus grandes valeurs de son vecteur d'entrée. Ceci permet d'uniformiser la taille (k) des explications fournies par les différentes méthodes et donc de faciliter leur comparaison. Ainsi on obtient :

$$\text{Exp}(v_i)[j] = \begin{cases} 0, \text{si } j \notin \text{top}k(\text{Imp}(v_i)) \\ 1, \text{si } j \in \text{top}k(\text{Imp}(v_i)) \end{cases}. \tag{5}$$

5 Protocole expérimental et resultats

Dans cette section, nous décrivons le protocole expérimental que nous avons suivi pour évaluer l'intérêt de l'approche proposée dans la section 4.

Pour évaluer l'explicabilité des GAEs dans la détection des anomalies, nous utilisons la GEA (Graph Explanation Accuracy). La métrique GEA (Agarwal et al. (2023)) est utilisée pour comparer le masque d'explication et le masque de vérité terrain lorsque ce dernier est disponible. Elle est définie comme l'indice de Jaccard entre le masque de vérité terrain et le masque d'explication. Elle prend une valeur comprise entre 0 et 1, où 1 représente le meilleur résultat possible et 0 le pire.

Nous comparons notre méthode, **Reconstruction error (ours)**, et les méthodes de l'état de l'art de la table 1 décrites dans la section 2. Nous considérons également une baseline triviale notée (RandEx) où les vecteurs d'importance sont générés aléatoirement. Dominant est utilisé comme modèle de détection d'anomalies (f). Le jeu de données utilisé ici est ShapeHouse décrit dans Agarwal et al. (2023).

La Figure 1 (gauche) présente les GEA@k moyens, avec écarts-types (calculés sur dix exécutions) en fonction de la longueur k de l'explication, pour les explications des attributs obtenues sur ShapeHouse. Cela montre que la méthode proposée (Ours), malgré sa simplicité, surpasse significativement toutes les baselines pour expliquer les prédictions du modèle Dominant. Integrated Gradients (IGEx) et GradEx ont des performances similaires à notre méthode pour des valeurs de k faibles, mais très inférieures pour des valeurs de k plus élevées, fournissant des explications partielles. Enfin, GNNExplainer fait à peine mieux que l'explainer aléatoire (RandEx) en termes de GEA.

Du fait de la limitation de place, nous ne présentons qu'un extrait des expériences réalisées. En complément des jeux de données générés (ShapeHouse, ShapeFlag, ShapeCircle), nos travaux incluent des tests sur des graphes réels avec anomalies injectées, comme proposé par Liu et al. (2022) (Cora, Citeseer, Amazon Photo). Pour évaluer la qualité des explications, nous avons utilisé la GEA@k, la précision@k et le rappel@k lorsque la vérité terrain est disponible. En son absence, les métriques de nécessité ($fid+$) et de suffisance ($fid-$), introduites par Amara et al. (2022), ont été employées pour quantifier l'impact des explications sur les prédictions. L'efficacité temporelle a également été mesurée afin de comparer les différentes méthodes. L'ensemble de ces expérimentations est décrite dans Giles (2024).

6 Conclusion

Dans cet article, nous avons introduit un cadre d'évaluation des méthodes d'explicabilité dans le contexte de la détection d'anomalies basée dans des graphes via GAE. Pour ce faire, nous avons adapté divers explainers disponibles pour les GNNs à l'architecture des GAEs. Nous avons également proposé de dériver directement les explications à partir des vecteurs d'erreurs de reconstruction du GAE. Les expériences montrent que l'utilisation de ces erreurs de reconstruction fournit les meilleures explications pour les attributs pour les quatre jeux de données. Pour les explications des arêtes, cette approche simple fonctionne aussi bien que les autres explainers dans tout les cas. En conclusion, la méthode proposée utilisant les erreurs de reconstruction, s'avère le meilleur explainer disponible pour GAE.

Références

Agarwal, C., O. Queen, H. Lakkaraju, et M. Zitnik (2023). Evaluating explainability for graph neural networks. *Scientific Data 10*(1), 144.

Aggarwal, C. C. (2017). *Outlier Analysis*. Springer Int. Publishing.

Akoglu, L. (2021). Anomaly mining - past, present and future. In *IJCAI*, pp. 4932–4936.

Amara, K., Z. Ying, Z. Zhang, Z. Han, Y. Zhao, Y. Shan, U. Brandes, S. Schemm, et C. Zhang (2022). Graphframex : Towards systematic evaluation of explainability methods for graph neural networks. In B. Rieck et R. Pascanu (Eds.), *Learning on Graphs Conference, LoG 2022*, Volume 198 of *Proceedings of Machine Learning Research*, pp. 44. PMLR.

Assaf, R., I. Giurgiu, J. Pfefferle, S. Monney, H. Pozidis, et A. Schumann (2021). An anomaly detection and explainability framework using convolutional autoencoders for data storage systems. In *IJCAI*, pp. 5228–5230.

Baldassarre, F. et H. Azizpour (2019). Explainability techniques for graph convolutional networks. *CoRR abs/1905.13686*.

Bandyopadhyay, S., L. N, S. V. Vivek, et M. N. Murty (2020). Outlier resistant unsupervised deep architectures for attributed network embedding. In *WSDN*, New York, NY, USA, pp. 25–33.

Chalapathy, R., A. K. Menon, et S. Chawla (2018). Anomaly detection using one-class neural networks. *ArXiv 1802.06360*.

Chandola, V., A. Banerjee, et V. Kumar (2009). Anomaly detection : A survey. *ACM Comput. Surv. 41*(3), 1–58.

Charte, D., F. Charte, M. J. del Jesus, et F. Herrera (2020). An analysis on the use of autoencoders for representation learning : Fundamentals, learning task case studies, explainability and challenges. *Neurocomputing 404*, 93–107.

Ding, K., J. Li, R. Bhanushali, et H. Liu (2019). Deep anomaly detection on attributed networks. In *SIAM ICDM*, pp. 594–602.

Fan, H., F. Zhang, et Z. Li (2020). Anomalydae : Dual autoencoder for anomaly detection on attributed networks. In *ICASSP*, pp. 5685–5689.

Giles, B. (Phd thesis, 2024). Detecting health care frauds in attributed graphs using interpretable or explainable methods. *Université Jean Monnet, https ://hal.science/tel-04808841*.

Giles, B., B. Jeudy, C. Largeron, et D. Saboul (2023). Suspicious : a resilient semi-supervised framework for graph fraud detection. In *ICTAI*, pp. 212–220.

Gorman, M., X. Ding, L. Maguire, et D. Coyle (2023). Anomaly detection in batch manufacturing processes using localized reconstruction errors from 1-d convolutional autoencoders. *IEEE TSM 36*(1), 147–150.

Grubbs, F. E. (1969). Procedures for detecting outlying observations in samples. *Technometrics 11*(1), 1–21.

Interdonato, R., M. Atzmueller, S. Gaito, R. Kanawati, C. Largeron, et A. Sala (2019). Feature-rich networks : going beyond complex network topologies. *Appl. Netw. Sci.*, 4 :1–4 :13.

Kipf, T. N. et M. Welling (2017). Semi-supervised classification with graph convolutional networks. In *ICLR*.

Kumagai, A., T. Iwata, et Y. Fujiwara (2021). Semi-supervised Anomaly Detection on Attributed Graphs. In *IJCNN*, pp. 1–8.

Liu, K., Y. Dou, Y. Zhao, X. Ding, X. Hu, R. Zhang, K. Ding, C. Chen, H. Peng, K. Shu, L. Sun, J. Li, G. H. Chen, Z. Jia, et P. S. Yu (2022). Benchmarking node outlier detection on graphs. In *NeurIPS*.

Ma, X., J. Wu, S. Xue, J. Yang, C. Zhou, Q. Z. Sheng, H. Xiong, et L. Akoglu (2021). A comprehensive survey on graph anomaly detection with deep learning. *IEEE TKDE*.

Molnar, C. (2022). *Interpretable Machine Learning* (2 ed.).

Pang, G., C. Shen, L. Cao, et A. v. d. Hengel (2021). Deep learning for anomaly detection : A review. *ACM Computing Surveys*, 1–38.

Ravi, A., X. Yu, I. Santelices, F. Karray, et B. Fidan (2021). General frameworks for anomaly detection explainability : comparative study. In *ICAS*, pp. 1–5. IEEE.

Simonyan, K., A. Vedaldi, et A. Zisserman (2014). Deep inside convolutional networks : Visualising image classification models and saliency maps. In *ICLR*.

Tang, J., J. Li, Z. Gao, et J. Li (2022). Rethinking graph neural networks for anomaly detection. In *ICML*, Volume 162, pp. 21076–21089.

Veličković, P., G. Cucurull, A. Casanova, A. Romero, P. Liò, et Y. Bengio (2018). Graph attention networks. *ICLR (Poster)*.

Wu, F., A. Souza, T. Zhang, C. Fifty, T. Yu, et K. Weinberger (2019). Simplifying graph convolutional networks. In *ICML*, pp. 6861–6871. PMLR.

Xu, K., W. Hu, J. Leskovec, et S. Jegelka (2019). How powerful are graph neural networks ? In *ICLR*.

Ying, Z., D. Bourgeois, J. You, M. Zitnik, et J. Leskovec (2019). Gnnexplainer : Generating explanations for graph neural networks. In *Advances in Neural Information Processing Systems*, Volume 32.

Yuan, H., H. Yu, J. Wang, K. Li, et S. Ji (2021). On explainability of graph neural networks via subgraph explorations. In *ICML*.

Summary

Graph Auto-Encoders (GAEs) have demonstrated remarkable effectiveness in detecting anomalies in graphs. However, their "black-box" nature makes it difficult to understand why they classify a node as anomalous. Moreover, despite the development of XAI, where many methods have been proposed to provide explanations for various deep learning models, there is a notable lack of an evaluation framework dedicated to anomaly detection in graphs. Our contribution addresses this gap by adapting existing evaluation frameworks to the specific challenges of anomaly detection using GAEs. Additionally, it introduces a simple yet effective explanation technique based on GAE reconstruction errors. Using this new framework, we evaluate the effectiveness of different explainers and experimentally show that the method we propose, based on reconstruction errors, outperforms other explainers for GAEs.