GNNExplainer: Visualisation des Explications d'un GNNExplainer sur un Réseau de Neurones Graphique

July 29, 2024

Abstract

Ce projet vise à explorer et à visualiser les explications fournies par GNNExplainer, une méthode innovante permettant d'interpréter les prédictions des modèles basés sur les réseaux neuronaux graphiques (GNN) appliqués à des ensembles de données structurées en graphe. Les GNN sont devenus des outils essentiels pour analyser et prédire sur des données complexes telles que les réseaux sociaux, les molécules chimiques et les réseaux de transport. Cependant, ces modèles souffrent souvent d'un manque d'interprétabilité, ce qui limite leur adoption dans des domaines où la transparence et la compréhension des décisions sont cruciales.

Dans ce travail, nous utilisons GNNExplainer pour identifier les structures de sous-graphe et les caractéristiques de nœuds les plus importantes qui contribuent aux prédictions des modèles GNN. En appliquant cette méthode à un ensemble de données spécifique, nous générons des explications interprétables pour chaque instance de graphe, mettant en évidence les motifs et les attributs clés utilisés par le modèle pour prendre des décisions. Nous formulons ce processus comme une tâche d'optimisation visant à maximiser l'information mutuelle entre les prédictions du modèle GNN et les distributions des structures de sous-graphe possibles.

À l'aide d'expériences sur des ensembles de données réels, nous démontrons l'efficacité de GNNExplainer dans l'identification des structures de graphe importantes et des caractéristiques de nœuds influentes. De plus, nous proposons une méthode de visualisation pour rendre ces explications compréhensibles, permettant aux utilisateurs de mieux comprendre le fonctionnement des modèles GNN et les décisions qu'ils prennent. Ces résultats ouvrent de nouvelles perspectives pour l'interprétabilité des modèles GNN et leur application dans des domaines où la transparence est primordiale.

1 Introduction

Dans de nombreuses applications du monde réel, notamment dans les domaines sociaux, de l'information, chimiques et biologiques, les données peuvent être na-

turellement modélisées sous forme de graphes. Les graphes sont des représentations de données puissantes mais difficiles à manipuler car elles nécessitent la modélisation d'informations relationnelles riches ainsi que d'informations sur les caractéristiques des nœuds. Pour relever ce défi, les réseaux neuronaux graphiques (GNN) ont émergé comme étant à la pointe de la technologie pour l'apprentissage automatique sur les graphes, en raison de leur capacité à incorporer de manière récursive des informations provenant des nœuds voisins dans le graphe, capturant ainsi naturellement à la fois la structure du graphe et les caractéristiques des nœuds. Malgré leurs forces, les GNN manquent de transparence car ils ne permettent pas facilement une explication intelligible par l'homme de leurs prédictions. Pourtant, la capacité à comprendre les prédictions des GNN est importante et utile pour plusieurs raisons : (i) elle peut accroître la confiance dans le modèle GNN, (ii) elle améliore la transparence du modèle dans un nombre croissant d'applications décisionnelles critiques en matière de justice, de confidentialité et d'autres défis de sécurité, et (iii) elle permet aux praticiens de comprendre les caractéristiques du réseau, d'identifier et de corriger les modèles systématiques d'erreurs commises par les modèles avant de les déployer dans le monde réel.

Bien qu'il existait déjà quelques méthodes pour expliquer les GNN, des approches récentes pour expliquer d'autres types de réseaux neuronaux ont suivi l'une des deux principales voies. Une ligne de travail approxime localement les modèles avec des modèles de substitution plus simples, qui sont ensuite sondés pour des explications. D'autres méthodes examinent soigneusement les modèles pour trouver des caractéristiques pertinentes et fournir de bonnes interprétations qualitatives des caractéristiques de haut niveau ou identifient des instances d'entrée influentes. Cependant, ces approches sont limitées dans leur capacité à incorporer des informations relationnelles, l'essence même des graphes. Comme cet aspect est crucial pour le succès de l'apprentissage automatique sur les graphes, toute explication des prédictions des GNN devrait tirer parti des informations relationnelles riches fournies par le graphe ainsi que des caractéristiques des nœuds. Dans ce travail, nous utilisons GNNEXPLAINER, une approche pour expliquer les prédictions faites par les GNN. GNNEXPLAINER prend un GNN entraîné et sa ou ses prédictions, et renvoie une explication sous la forme d'un petit sous-graphe du graphe d'entrée ainsi qu'un petit sous-ensemble des caractéristiques des nœuds qui sont les plus influentes pour la ou les prédictions. L'approche est agnostique au modèle et peut expliquer les prédictions de n'importe quel GNN sur n'importe quelle tâche d'apprentissage automatique pour les graphes, y compris la classification des nœuds, la prédiction des liens et la classification des graphes. Elle gère les explications à une seule instance ainsi que les explications à plusieurs instances. Dans le cas des explications à une seule instance, GNNEXPLAINER explique la prédiction d'un GNN pour une instance particulière (c'est-à-dire, une étiquette de nœud, un nouveau lien, une étiquette de niveau de graphe). Dans le cas des explications à plusieurs instances, GNNEXPLAINER fournit une explication qui explique de manière cohérente un ensemble d'instances (par exemple, les nœuds d'une classe donnée). GNNEXPLAINER spécifie une explication

comme un sous-graphe riche de l'ensemble du graphe sur lequel le GNN a été entraîné, de sorte que le sous-graphe maximise l'information mutuelle avec la ou les prédictions du GNN. Cela est réalisé en formulant une approximation variationnelle en champ moyen et en apprenant un masque de graphe à valeurs réelles qui sélectionne le sous-graphe important du graphe de calcul du GNN. Simultanément, GNNEXPLAINER apprend également un masque de caractéristiques qui masque les caractéristiques de nœuds non importantes

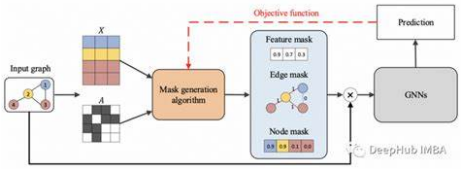

Figure 1: Le pipeline général des méthodes basées sur les perturbations. Ils utilisent différents algorithmes de génération de masques pour obtenir différents types de masques. Notez que le masque peut correspondre à des nœuds, des arêtes ou des caractéristiques de nœud. Dans cet exemple, nous montrons un masque souple pour les caractéristiques des nœuds, un masque discret pour les arêtes et un masque discret approché pour les nœuds. Ensuite, le masque est combiné avec le graphique d'entrée pour capturer des informations d'entrée importantes. Enfin, les GNN formés évaluent si la nouvelle prédiction est similaire à la prédiction originale et peuvent fournir des conseils pour améliorer les algorithmes de génération de masques.

2 Etat de L'art

Récemment, plusieurs approches ont été proposées pour expliquer les prédictions des modèles graphiques profonds. Ces méthodes se concentrent sur différents aspects des modèles graphiques et fournissent différentes perspectives pour comprendre ces modèles. Elles répondent généralement à quelques questions; certaines d'entre elles sont, quelles arêtes d'entrée sont les plus importantes? quels nœuds d'entrée sont les plus importants? quelles caractéristiques des nœuds sont les plus importantes? quels motifs de graphe maximiseront la prédiction d'une certaine classe? Pour mieux comprendre ces méthodes, nous proposons une taxonomie des différentes techniques d'explication pour les GNN. La structure de notre taxonomie est illustrée dans la Figure 1. En fonction des types d'explications fournies, différentes techniques sont classées en deux classes principales: les méthodes au niveau de l'instance et les méthodes au niveau du modèle.

2.1 Méthodes

GNNExplainer utilise des masques souples pour les arêtes et les caractéristiques des nœuds afin d'expliquer les prédictions via une optimisation de masque. Pour obtenir des masques, il initialise aléatoirement des masques souples et les traite comme des variables entraînables. Ensuite, GNNExplainer combine les masques avec le graphe original via des multiplications élémentaires. Ensuite, les masques sont optimisés en maximisant l'information mutuelle entre les prédictions du graphe original et les prédictions du graphe nouvellement obtenu. Même si différents termes de régularisation, tels que l'entropie élémentaire, sont utilisés pour encourager les masques optimisés à être discrets, les masques obtenus sont toujours des masques souples, de sorte que GNNExplainer ne peut pas éviter le problème d'évidence introduite. De plus, les masques sont optimisés pour chaque graphe d'entrée individuellement et donc les explications peuvent manquer d'une vue globale.

PGExplainer apprend des masques discrets approximatifs pour les arêtes afin d'expliquer les prédictions. Pour obtenir des masques d'arête, il entraîne un prédicteur de masque paramétré pour prédire des masques d'arête. Étant donné un graphe d'entrée, il obtient d'abord les plongements pour chaque arête en concaténant les plongements de nœuds correspondants. Ensuite, le prédicteur utilise les plongements d'arête pour prédire la probabilité de chaque arête d'être sélectionnée, ce qui peut être traité comme le score d'importance. Ensuite, les masques discrets approximatifs sont échantillonnés via le tour de ré-échantillonnage. Enfin, le prédicteur de masque est entraîné en maximisant l'information mutuelle entre les prédictions originales et les nouvelles prédictions. Notez que même si le tour de ré-échantillonnage est utilisé, les masques obtenus ne sont pas strictement discrets mais peuvent largement atténuer le problème d'évidence introduite. De plus, puisque toutes les arêtes dans l'ensemble de données partagent le même prédicteur, les explications peuvent fournir une compréhension globale des GNN entraînés.

GraphMask est une méthode post-hoc pour expliquer l'importance des arêtes dans chaque couche GNN. Similaire à PGExplainer, il entraîne un classificateur pour prédire si une arête peut être supprimée sans affecter les prédictions originales. Cependant, GraphMask obtient un masque d'arête pour chaque couche GNN tandis que PGExplainer se concentre uniquement sur l'espace d'entrée. De plus, pour éviter de modifier les structures de graphe, les arêtes supprimées sont remplacées par des connexions de base apprenables, qui sont des vecteurs avec les mêmes dimensions que les plongements de nœuds. Notez que la distribution concrète binaire [68] et le tour de ré-échantillonnage sont utilisés pour approximer des masques discrets. De plus, le classificateur est entraîné en utilisant l'ensemble de données entier en minimisant un terme de divergence, qui mesure la différence entre les prédictions du réseau. Tout comme PGExplainer, il peut largement atténuer le problème d'évidence introduite et fournir une compréhension globale des GNN entraînés.

2.1.1 Formulation des explications d'un Graph Neural Network

Soit G un graphe sur les arêtes E et les nœuds V qui sont associés à des caractéristiques de nœuds de dimension d $X = \{x_1, ..., x_n\}, x_i \in R^d$. Sans perte de généralité, nous considérons le problème d'explication d'une tâche de classification des nœuds. Soit f une fonction d'étiquetage sur les nœuds $f : V \to \{1, ..., C\}$ qui mappe chaque nœud de V à l'une des C classes. Le modèle GNN Φ est optimisé sur tous les nœuds de l'ensemble d'entraînement puis est utilisé pour la prédiction, c'est-à-dire pour approximer f sur de nouveaux nœuds.

2.1.2 Formulation des explications d'un Graph Neural Network

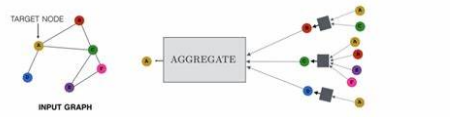

Figure 2: Vue d'ensemble de la manière dont un seul nœud agrège les messages de son voisinage local. Le modèle agrège les messages des voisins du graphe local de A (c'est-à-dire B, C et D), et à leur tour, les messages provenant de ces voisins sont basés sur les informations agrégées de leurs voisinages respectifs, et ainsi de suite. Cette visualisation montre une version à deux couches d'un modèle de passage de messages. Remarquez que le graphe de calcul du GNN forme une structure arborescente en dépliant le voisinage autour du nœud cible.

À la couche l, la mise à jour du modèle GNN Φ implique trois calculs clés. (1) Tout d'abord, le modèle calcule des messages neuronaux entre chaque paire de nœuds. Le message pour la paire de nœuds (v_i, v_j) est une fonction MSG des représentations h^i_{l-1} et h^j_{l-1} des nœuds dans la couche précédente et de la relation r_{ij} entre les nœuds : $m^l_{ij} = MSG(h^i_{l-1}, h^j_{l-1}, r_{ij})$. (2) Ensuite, pour chaque nœud v_i, le GNN agrège les informations de tous les messages entrants en utilisant une fonction d'agrégation AGG : $a^l_i = AGG(\{m^l_{ji} : j \in N_i\})$, où N_i est l'ensemble des nœuds voisins de v_i. (3) Enfin, le modèle met à jour la représentation du nœud en combinant l'incorporation précédente avec les informations agrégées : $h^l_i = update(h^{l-1}_i, a^l_i)$.

Notre observation clé est que le graphe de calcul du nœud v, qui est défini par l'agrégation basée sur le voisinage du GNN (voir Figure 2), détermine entièrement toutes les informations que le GNN utilise pour générer la prédiction $\hat{y}$ au nœud v. En particulier, le graphe de calcul de v indique au GNN comment générer l'incorporation de v. Notons ce graphe de calcul par $G_c(v)$, la matrice d'adjacence binaire associée par $A_c(v) \in \{0, 1\}^{n \times n}$, et l'ensemble de fonctionnalités associé par $X_c(v) = \{x_j | v_j \in G_c(v)\}$. Le modèle GNN Φ apprend une distribution conditionnelle $P_\Phi(Y|G_c, X_c)$, où Y est une variable aléatoire représentant les étiquettes $\{1, \ldots, C\}$, indiquant la probabilité que les nœuds

appartiennent à chacune des C classes. Une prédiction du GNN est donnée par $\hat{y} = \Phi(G_c(v), X_c(v))$, ce qui signifie qu'elle est entièrement déterminée par le modèle Φ, les informations structurales du graphe $G_c(v)$, et les informations de fonctionnalités des nœuds $X_c(v)$. En effet, cette observation implique que nous devons seulement considérer la structure du graphe $G_c(v)$ et les fonctionnalités des nœuds $X_c(v)$ pour expliquer $\hat{y}$. Formellement, GNNEXPLAINER génère une explication pour la prédiction $\hat{y}$ comme (GS, XF_S), où GS est un petit sous-graphe du graphe de calcul. XS est la fonctionnalité associée de GS, et XF_S est un petit sous-ensemble de fonctionnalités des nœuds (masqué par le masque F, c'est-à-dire $XF_S = \{x_j^F | v_j \in GS\}$) qui sont les plus importantes pour expliquer $\hat{y}$.

2.1.3 GNNExplainer

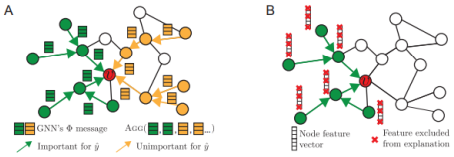

Figure 3: A. Graphe de calcul GNN G_c (en vert et orange) pour effectuer la prédiction $\hat{y}$ au nœud v. Certaines arêtes dans G_c forment des chemins importants de passage de message neuronal (en vert), qui permettent à des informations utiles sur les nœuds d'être propagées à travers G_c et agrégées au nœud v pour la prédiction, tandis que d'autres arêtes ne le font pas (en orange). Cependant, le GNN doit agréger à la fois des messages importants et non importants pour former une prédiction au nœud v. L'objectif de GNNEXPLAINER est d'identifier un petit ensemble de caractéristiques et de chemins importants (en vert) qui sont cruciaux pour la prédiction. B. En plus de GS (en vert), GNNEXPLAINER identifie quelles dimensions de caractéristiques des nœuds de GS sont importantes pour la prédiction en apprenant un masque de caractéristiques de nœud.

Étant donné un nœud v, notre objectif est d'identifier un sous-graphe $G_S \subseteq G_c$ et les caractéristiques associées $X_S = \{x_j | v_j \in G_S\}$ qui sont importantes pour la prédiction du GNN $\hat{y}$. Pour l'instant, nous supposons que X_S est un petit sous-ensemble des caractéristiques des nœuds en dimension d. Nous formalisons la notion d'importance en utilisant l'information mutuelle MI et formulons GNNExplainer comme le cadre d'optimisation suivant :

$$\max_{G_S} MI(Y, (G_S, X_S)) = H(Y) - H(Y|G = G_S, X = X_S)(1).$$

Pour le nœud v, MI quantifie le changement de probabilité de prédiction $\hat{y} = \Phi(G_c, X_c)$ lorsque le graphe de calcul de v est limité au sous-graphe d'explication

G_S et que ses caractéristiques de nœud sont limitées à X_S. Par exemple, considérons la situation où $v_j \in G_c(v_i)$, $v_j \neq v_i$. Ensuite, si la suppression de v_j de $G_c(v_i)$ diminue fortement la probabilité de prédiction $\hat{y}_i$, le nœud v_j est une bonne explication contrefactuelle pour la prédiction en v_i. De même, considérons la situation où $(v_j, v_k) \in G_c(v_i)$, $v_j \neq v_i$, $v_k \neq v_i$. Ensuite, si la suppression d'un bord entre v_j et v_k diminue fortement la probabilité de prédiction $\hat{y}_i$, alors l'absence de ce bord est une bonne explication contrefactuelle pour la prédiction en v_i.

En examinant l'équation (1), nous voyons que le terme d'entropie $H(Y)$ est constant car Φ est fixé pour un GNN entraîné. En conséquence, la maximisation de l'information mutuelle entre la distribution des étiquettes prédites Y et l'explication (G_S, X_S) équivaut à minimiser l'entropie conditionnelle $H(Y|G = G_S, X = X_S)$, ce qui peut être exprimé comme suit :

$$H(Y|G = G_S, X = X_S) = -E_{Y|G_S,X_S}[\log P_\Phi(Y|G = G_S, X = X_S)]$$

L'explication pour la prédiction $\hat{y}$ est donc un sous-graphe G_S qui minimise l'incertitude de Φ lorsque le calcul GNN est limité à G_S. En effet, G_S maximise la probabilité de $\hat{y}$. Pour obtenir une explication compacte, nous imposons une contrainte sur la taille de G_S : $|G_S| \leq KM$, de sorte que G_S ait au plus KM nœuds. En effet, cela implique que GNNEXPLAINER vise à débruiter G_c en prenant KM arêtes qui donnent la plus haute information mutuelle avec la prédiction.

2.1.4 Métriques d'évaluation

Pour évaluer l'explicabilité des GNN, la fidélité est communément évaluée, ce qui renvoie à la précision avec laquelle l'explication reflète le véritable processus de raisonnement du modèle. Un autre critère populaire est la plausibilité , qui fait référence à la convaincance de l'explication pour les humains. Bien qu'il n'existe pas de métrique d'évaluation standard dans ce domaine, toutes les métriques actuellement utilisées reflètent généralement l'un de ces deux aspects.

Évaluation de la plausibilité

Précision : La précision est utilisée lorsque les ensembles de données contiennent des modèles d'explication de vérité terrain définis par l'humain. La précision mesure la pertinence de l'explication générée par rapport à la réalité terrain. Formellement, elle est définie comme $\frac{|GT \cap E|}{|E|}$, où GT est l'ensemble des arêtes dans l'explication de la réalité terrain et E est l'ensemble des arêtes dans l'explication générée. Remarquez que la précision est liée à la taille de l'explication générée ; ainsi, elle nécessite souvent que les utilisateurs définissent un seuil pour l'explication dense en fonction de la taille des explications de vérité terrain. Comme cela nécessite une vérité terrain définie par l'humain pour les ensembles de données, la précision n'est pas applicable à la plupart des ensembles de données du monde réel.

Évaluation de la fidélité

Sparsité et fidélité : La sparsité est généralement combinée à la fidélité, où la sparsité mesure combien d'arêtes redondantes sont supprimées du graphe original. Une sparsité plus élevée est préférée, car intuitivement le sous-graphe d'explication devrait être le plus petit sous-graphe contenant toutes les informations nécessaires pour prendre la décision. La fidélité mesure à quel point le modèle d'explication est fidèle au modèle GNN d'origine. Formellement, la sparsité est définie comme :

Sparsité = 1 - m / M (2)

où m est la taille du sous-graphe important (c'est-à-dire le nombre d'arêtes) et M est la taille du graphe d'origine.La fidélité est définie comme : **Fidelity** = $P(Y = c|G) - P(Y = c|G \setminus GS)$, $c = \arg\max_{c \in C} P(Y = c|G)$ (3) où P est la distribution probabiliste générée par le modèle, Y est la prédiction, G est le graphe d'origine, GS est l'explication et C est l'ensemble de toutes les classes. Cette équation évalue la fidélité de l'explication au modèle en mesurant la différence entre les prédictions des graphes en retirant les sous-graphes importants et les graphes d'entrée originaux. En pratique, la fidélité est calculée et moyennée sur des explications avec différentes sparsités.

Fidélité inverse : Cette métrique d'évaluation compare l'exactitude du modèle en utilisant le sous-graphe important et l'exactitude du modèle en utilisant le graphe d'origine par rapport à la tâche sur l'ensemble de test. Elle est définie comme :

Fidélité inverse = 1 / N * (1(ˆyi' = yi) - 1(ˆyi = yi)) (4)

où yi est l'étiquette, N est le nombre d'échantillons, ˆyi et ˆyi' sont les prédictions du modèle d'origine et des explications. Les métriques d'évaluation de la fidélité sont motivées par la compréhension du processus de raisonnement sous-jacent des modèles, différentiées du processus de raisonnement produit par les méthodes d'explicabilité, et ne nécessitent pas d'explications de vérité terrain ; par conséquent, elles peuvent être appliquées à toutes les tâches. Les études qui utilisent la précision sélectionnent généralement le même nombre d'arêtes ayant le score d'importance le plus élevé que le nombre d'arêtes dans la vérité terrain. Comme discuté ci-dessus, premièrement, la vérité terrain n'est pas disponible dans la plupart des ensembles de données du monde réel ; deuxièmement, la réalité terrain définie par l'humain n'est pas garantie d'être correcte car nous n'aurions pas besoin d'expliquer les modèles si nous savions déjà comment ils raisonnent, ce qui rend la métrique moins fiable. La fidélité et la fidélité inverse sont généralement calculées sur différentes sparsités. Le problème avec cela est qu'une certaine sparsité peut avoir un impact différent sur différents points de données. Par exemple, une sparsité de 50 pourcent pourrait ne pas affecter la prédiction du modèle pour un nœud avec un voisinage important, tandis qu'elle pourrait changer significativement la prédiction du modèle pour un nœud avec un petit voisinage. Par conséquent, il n'est pas juste de comparer directement la fidélité entre différents points de données basés sur la même sparsité.

3 Experimentation

La section expérimentale débute par une présentation de dataset utilisé, en plus d'une comparaison avec des méthodes de référence alternatives et la mise en place des expériences. Ensuite, nous nous plongeons dans les expériences visant à élucider les GNN pour des tâches de classification de nœuds. À travers une analyse qualitative et quantitative, nos résultats soulignent la précision et l'efficacité de GNNEXPLAINER dans la discernement des explications. Ces explications, englobant à la fois les attributs structurels des graphes et les caractéristiques des nœuds individuels, contribuent de manière significative à améliorer notre compréhension du comportement des GNN et de leur processus décisionnel.

TheMondialDatabase: Le dataset utilisé dans ce projet est structuré autour de données géopolitiques, démographiques et économiques extraites de la base de données 'Mondial', une base de données publique contenant des informations détaillées sur les pays du monde. Les données sont initialement structurées en Prolog, et notre traitement les convertit en un format adapté aux réseaux de neurones graphiques (GNN) grâce à PyTorch Geometric (PyG). Le processus de transformation commence par la lecture des enregistrements Prolog qui décrivent les attributs des pays tels que la population, la croissance démographique, la mortalité infantile, le PIB, la répartition sectorielle de l'économie, et d'autres indicateurs clés. Chaque pays est identifié par un code unique, et des relations telles que les frontières partagées entre les pays sont également enregistrées pour créer les arêtes du graphe. Pour chaque pays, nous avons extrait des caractéristiques telles que l'aire géographique, la population, la croissance démographique, et la mortalité infantile, ainsi que des données économiques détaillées incluant le PIB et la répartition sectorielle (agriculture, services, industrie). Les relations politiques (comme l'indépendance et les dépendances), religieuses (incluant le pourcentage de la population adhérant à une religion donnée) et géographiques (appartenance continentale) sont également modélisées. Enfin Les pays et leurs caractéristiques sont convertis en nœuds du graphique, où chaque nœud représente un pays et ses caractéristiques multidimensionnelles (voir Figure 4). Les relations entre pays, telles que les frontières communes, sont transformées en arêtes, créant ainsi un réseau interconnecté qui reflète la complexité et l'interdépendance des données géopolitiques. Les identifiants numériques uniques sont attribués aux pays et aux continents pour faciliter les calculs dans le modèle GNN.

3.1 Méthodologie

Répartition du dataset : La répartition des données pour les ensembles d'entraînement, de test et de validation est faite ainsi dans notre projet : sur le total des 246 noeuds,Pour l'entraînement, nous utilisons 180 nœuds, ce qui représente environ 73.2% de l'ensemble des données. Cet ensemble d'entraînement aide notre modèle à apprendre en ajustant ses poids pour réduire l'erreur de prédiction.

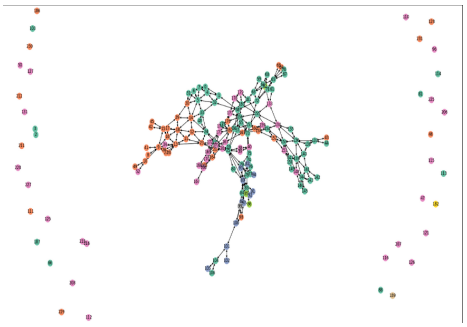

Figure 4: Visualisation du graphe .

Pour la validation, nous avons alloué 30 nœuds, soit à peu près 12.2% du total. L'ensemble de validation est utilisé pour peaufiner les réglages de notre modèle et pour décider quand arrêter l'entraînement afin d'éviter que notre modèle ne mémorise les données d'entraînement plutôt que d'apprendre à généraliser à partir de celles-ci. Pour les tests, nous avons réservé 36 nœuds, ce qui fait environ 14.6% des nœuds disponibles. Cet ensemble de test est essentiel pour évaluer comment notre modèle performe sur des données qu'il n'a jamais rencontrées, nous donnant ainsi une idée de sa capacité à généraliser et à être appliqué à de nouvelles situations.

Paramètre du modèle : Nous utilisons un modèle de graphe neuronal, le Graph Convolutional Network (GCN), équipé de 9 canaux cachés pour traiter les données de chaque nœud en vecteurs de dimension 9. L'optimiseur Adam est employé pour ajuster les poids du réseau, avec un taux d'apprentissage fixé à 0.01, afin de minimiser efficacement la fonction de perte, CrossEntropyLoss, qui est idéale pour les tâches de classification. Le processus d'entraînement consiste en des passes avant où les données sont traitées pour calculer la perte, suivies de rétropropagation pour la mise à jour des poids. Un mécanisme d'arrêt anticipé est en place, déclenché si la réduction de la perte entre deux époques consécutives est inférieure à 0.0001, ce qui prévient le surapprentissage. Après l'entraînement, le modèle est évalué sur un ensemble de test pour vérifier sa capacité à généraliser, avec la précision mesurée par le ratio de prédictions correctes. Nous visualisons également la progression de l'apprentissage à travers des graphiques de la perte d'entraînement, fournissant une représentation claire de l'efficacité de l'entraînement au fil des époques.

3.1.1 Résultats

Dans le cadre de notre étude visant à évaluer l'efficacité du modèle Graph Convolutional Network (GCN) dans la prédiction des orientations religieuses majeures des pays, nous avons réalisé une analyse quantitative complète. Les résultats obtenus fournissent des informations essentielles sur les performances du modèle GCN ainsi que des insights pertinents sur la nature de ses prédictions.

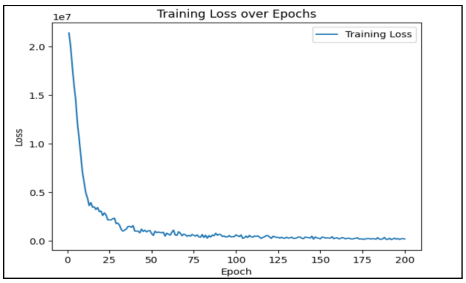

Figure 5: Visualisation l'évolution de la loss function .

L'exactitude du modèle GCN sur l'ensemble d'entraînement est de 40%, tandis que sur l'ensemble de test, cette exactitude s'élève légèrement à 38.89%. Ces performances, bien que modestes, doivent être interprétées en tenant compte du contexte du dataset utilisé, qui comprend seulement 246 nœuds représentant des pays. Cette limitation en termes de volume de données peut affecter la capacité du modèle à généraliser efficacement, ce qui explique en partie les performances observées.

Il est important de souligner que l'objectif principal de notre étude n'est pas seulement d'obtenir un modèle performant, mais surtout de comprendre les mécanismes sous-jacents des prédictions du modèle, même en présence de limitations de données. Dans cette optique, nous avons comparé les performances du modèle GCN à celles d'un modèle de référence, le MLP (Perceptron Multi-Couches).

Les résultats de cette comparaison sont résumés dans le tableau 1. On constate que le GCN affiche une accuracy de 39%, un rappel de 38.89%, et un F1-score de 37.02%. En revanche, le MLP présente des performances légèrement inférieures, avec une accuracy de 30%, un rappel de 29.89%, et un F1-score de 29.97%.

Il convient de noter que, malgré la simplicité apparente des mesures de performance telles que l'accuracy, le rappel et le F1-score, l'interprétation de ces indicateurs dans le contexte des prédictions du modèle peut être complexe. Ainsi, une analyse qualitative approfondie des prédictions, associée à des méthodes d'interprétation telles que l'analyse des masques (node_mask et edge_mask) ou l'utilisation de modèles de langage naturel pour générer des explications textuelles, peut fournir des informations précieuses sur les facteurs qui influencent les prédictions du modèle.

Modèles	Accuracy	Rappel	F1-score
GCN	39%	38.89%	37.02%
MLP	30%	29.89%	29.97%

Table 1: Résultats des modèles sur le dataset.

3.1.2 GNNExplainer

Utilisation de GNNExplainer : Nous utilisons GNNExplainer pour fournir des explications sur les prédictions de notre modèle GCN. GNNExplainer génère des masques pour chaque nœud et chaque arête du graphe, indiquant leur importance respective dans la prédiction du modèle.

Une fois les masques obtenus, nous les utilisons pour comprendre les facteurs sous-jacents qui influencent les prédictions du modèle. Ces explications nous aident à interpréter les résultats du modèle en mettant en évidence les caractéristiques et les relations les plus pertinentes pour chaque prédiction. De plus il genere le sous graphe induit du grphe de départ pour chaque noeud , ceci pour montrer l'importance des voisins d'un noeud dans sa prediction En visualisant les masques générés par GNNExplainer, nous sommes en mesure de représenter graphiquement l'importance de chaque feature de chaque nœud et chaque arête dans le graphe. Cette visualisation nous permet de mieux comprendre les déterminants clés des prédictions du modèle et facilite l'interprétation des résultats.

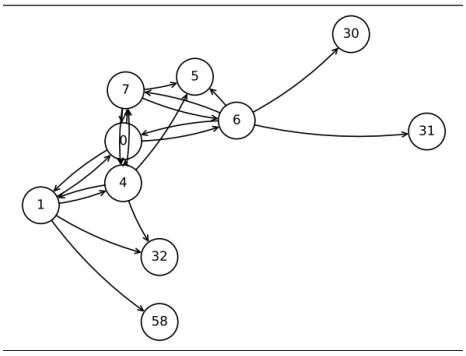

Figure 6: sous graphe induit pour explication du noeud 1.

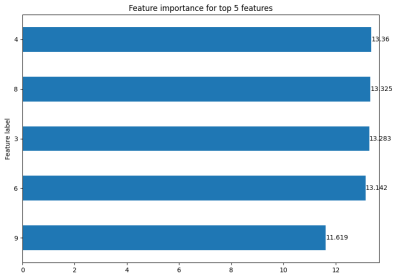

Figure 7: histogramme des features les plus importants pourun noeud donné.

3.1.3 Visualisation des explications

La visualisation des explications a été réalisée en utilisant les masques générés par le modèle GNNExplainer. Ces masques ont été utilisés pour attribuer une couleur à chaque pays sur une carte, en fonction de ses caractéristiques les plus importantes pour sa prédiction de religion. Cette approche permet de présenter de manière intuitive les facteurs qui influencent les prédictions du modèle, en fournissant une représentation visuelle des caractéristiques saillantes de chaque pays.

- Cette visualisation offre plusieurs avantages :
  - Elle facilite la compréhension des prédictions du modèle en mettant en évidence les caractéristiques les plus influentes pour chaque pays.
  - Elle permet d'identifier les similitudes entre les profils de différents pays, en mettant en évidence ceux qui partagent les mêmes caractéristiques importantes.
  - Elle fournit une interface conviviale pour les utilisateurs non spécialistes du domaine, en remplaçant les masques bruts par une représentation visuelle intuitive.

La figure ci-dessous illustre cette visualisation, où chaque pays est coloré en fonction de sa caractéristique la plus importante pour la prédiction de sa religion(numéro du pays: numéro du feature).

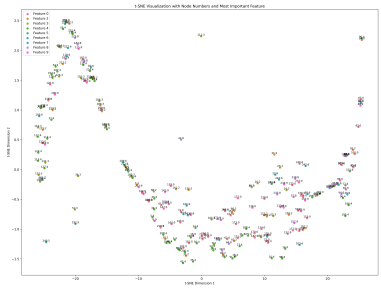

Figure 8: Visualisation des explications .

Cette approche de visualisation des explications contribue à rendre les résultats du modèle plus accessibles et compréhensibles pour un large éventail d'utilisateurs, favorisant ainsi une meilleure interprétation et utilisation des prédictions du modèle.

4 Directions Futures

Bien que notre étude ait permis d'explorer les prédictions de notre modèle GCN et de visualiser les caractéristiques les plus importantes pour chaque pays, plusieurs pistes de recherche demeurent pour améliorer notre compréhension des orientations religieuses des pays à l'aide des techniques de deep learning.

Tout d'abord, l'utilisation de modèles plus complexes et de données supplémentaires pourrait permettre d'améliorer les performances de prédiction du modèle. Les modèles de graphes plus avancés, tels que les graphes attentionnels, pourraient être explorés pour capturer des relations plus complexes entre les pays. De plus, l'incorporation de données supplémentaires telles que des informations géographiques, économiques ou historiques pourrait enrichir l'analyse et fournir des insights plus approfondis sur les déterminants des orientations religieuses.

Ensuite, une analyse plus détaillée des clusters de pays similaires, basée sur les caractéristiques les plus importantes identifiées, pourrait permettre de mieux comprendre les dynamiques régionales et les influences culturelles sur les orientations religieuses. Des méthodes d'apprentissage non supervisé telles que le clustering pourraient être utilisées à cette fin.

Par ailleurs, une exploration approfondie des explications fournies par le modèle, en particulier à l'aide de techniques de langage naturel comme les modèles de langage pré-entraînés (LLM), pourrait permettre de générer des explications textuelles plus détaillées et compréhensibles des prédictions du modèle. Ces explications pourraient être précieuses pour les décideurs politiques, les chercheurs et d'autres parties prenantes cherchant à comprendre les prédictions du modèle et à prendre des décisions éclairées.

Enfin, une validation approfondie des résultats du modèle à l'aide de données externes et une évaluation de l'impact des caractéristiques sélectionnées sur les prédictions pourraient renforcer la robustesse et la fiabilité de notre approche. Des études comparatives avec d'autres méthodes de prédiction des orientations religieuses pourraient également fournir des insights supplémentaires sur l'efficacité de notre approche.

En résumé, notre étude constitue une première étape prometteuse dans l'utilisation des techniques de deep learning pour comprendre les orientations religieuses des pays. Les directions futures évoquées ici ouvrent la voie à des recherches plus approfondies et à des applications potentielles dans divers domaines.

5 Conclusion

Nous avons présenté GNNEXPLAINER, une méthode novatrice pour expliquer les prédictions de n'importe quel GNN sur n'importe quelle tâche d'apprentissage automatique basée sur un graphe sans nécessiter de modification de l'architecture GNN sous-jacente ou de re-entraînement. Notre étude démontre que la visualisation des explications fournies par GNNEXPLAINER offre une perspective

précieuse pour mieux comprendre les facteurs influençant les prédictions des modèles graphiques. En intégrant cette visualisation dans notre analyse, nous avons pu identifier des profils de pays similaires en termes de caractéristiques importantes pour leur prédiction religieuse, offrant ainsi un moyen plus accessible pour les utilisateurs de saisir les décisions du modèle. Ce travail représente une avancée significative dans la recherche sur l'explicabilité des prédictions des modèles d'apprentissage automatique, offrant une méthode efficace pour analyser et interpréter les résultats des GNN sur des graphes avec des caractéristiques de nœuds riches.

References

[1] Yuan, Hao, et al. "Explainability in graph neural networks: A taxonomic survey." *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[2] Ying, Zhitao, et al. "Gnnexplainer: Generating explanations for graph neural networks." *Advances in neural information processing systems 32* (2019).

[3] Huang, Qiang, et al. "Graphlime: Local interpretable model explanations for graph neural networks." *IEEE Transactions on Knowledge and Data Engineering* (2022).

[4] Khan, A., Mobaraki, E. B. (2023). Interpretability Methods for Graph Neural Networks. In *2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA)* (pp. 1-4). Thessaloniki, Greece. doi: 10.1109/DSAA60987.2023.10302600.

[5] Adadi, A., Berrada, M. (2018). Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6, 52138–52160.

[6] Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B. (2018). Sanity checks for saliency maps. In *NeurIPS*.

[7] Gethsiyal Augasta, M., Kathirvalavakumar, T. (2012). Reverse Engineering the Neural Networks for Rule Extraction in Classification Problems. *Neural Processing Letters*, 35(2), 131–150.

[8] Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., ... et al. (2018). Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

[9] Chen, J., Zhu, J., Song, L. (2018). Stochastic training of graph convolutional networks with variance reduction. In *ICML*.

[10] Chen, J., Song, L., Wainwright, M. J., Jordan, M. I. (2018). Learning to explain: An information-theoretic perspective on model interpretation. *arXiv preprint arXiv:1802.07814*.

[11] Chen, J., Ma, T., Xiao, C. (2018). Fastgcn: fast learning with graph convolutional networks via importance sampling. In *ICLR*.

[12] Chen, Z., Li, L., Bruna, J. (2019). Supervised community detection with line graph neural networks. In *ICLR*.

[13] Cho, E., Myers, S., Leskovec, J. (2011). Friendship and mobility: user movement in location-based social networks. In *KDD*.