Une rencontre entre les noyaux de graphes et la détection d'anomalies dans les réseaux

Hicham Lesfari, Frédéric Giroire

Une rencontre entre les noyaux de graphes et la détection d'anomalies dans les réseaux

Hicham Lesfari et Frédéric Giroire

Université Côte d'Azur, Inria, CNRS, France

La détection d'anomalies demeure une tâche cruciale pour assurer une gestion efficace et flexible d'un réseau. Récemment, les noyaux de graphes ont connu un grand succès dans de nombreux domaines, notamment en bio-informatique et vision artificielle. Notre travail vise à étudier leur pouvoir de discrimination dans le domaine des réseaux afin de détecter les vulnérabilités et catégoriser le trafic. Dans cet article, nous présentons *Nadege*, un système d'apprentissage à l'intérieur duquel nous concevons un nouveau noyau de graphe adapté au profilage de réseaux. De surcroît, nous proposons des algorithmes avec des garanties d'approximation théoriques ainsi qu'une politique de détection hybride. Finalement, nous évaluons les performances de *Nadege* en menant des expériences approfondies sur une variété d'environnements réseaux. Pour différents scénarios, nous montrons son efficacité à empêcher les anomalies de perturber le réseau tout en fournissant une assistance pour la surveillance du trafic.

Mots-clefs : Noyaux de graphe, Détection d'anomalies, Apprentissage automatique, Réseaux informatiques

1 Introduction

L'apprentissage à partir de graphes de données a pris un essor notable grâce aux récents progrès dans le domaine de l'Intelligence Artificielle. Toutefois, combiner graphes et algorithmes d'apprentissage automatique pose plusieurs défis au coeur desquels se situe la recherche de méthodes efficaces pour comparer entre graphes. Parmi ces méthodes, les noyaux de graphe sont apparus comme un outil prometteur et connaissent un large succès dans plusieurs domaines, notamment en bio-informatique, vision artificielle et NLP.

Noyaux de graphes. Un noyau de graphe est une fonction de noyau [2] qui permet de mesurer la similarité entre des graphes. Elle correspond à un produit interne défini sur des graphes qui équivaut à les plonger implicitement dans un espace d'Hilbert. Un avantage important de ce plongement est de permettre aux méthodes d'apprentissage automatique d'être utilisées directement sur les graphes en étendant l'applicabilité de l'arsenal des méthodes à noyaux (e.g., machines à vecteurs de support) aux graphes. Un tel avantage exploite la capacité des noyaux à opérer sur des espaces de dimensions arbitrairement larges sans difficultés majeures de calcul, à condition que les algorithmes d'apprentissage mettant en œuvre le modèle linéaire de choix soient réécrits exclusivement en termes de produits internes. Cette observation (voir Fig. 1) est dite "astuce du noyau" [2] : étant donné une fonction de noyau K définie sur un espace vectoriel U, il existe un espace vectoriel V et une fonction ϕ qui fait correspondre U à V de telle sorte que : $K(x,y) = \phi(x).\phi(y)$ pour chaque x, y dans U. La valeur du noyau calculée entre deux points x, y dans l'espace original U peut donc être utilisée comme substitut du produit interne des points correspondants $\phi(x), \phi(y)$ dans l'espace transformé V.

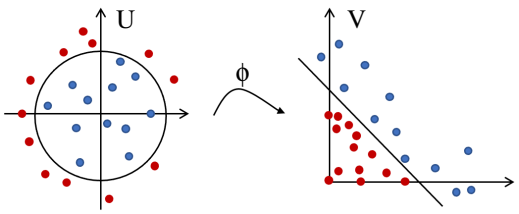
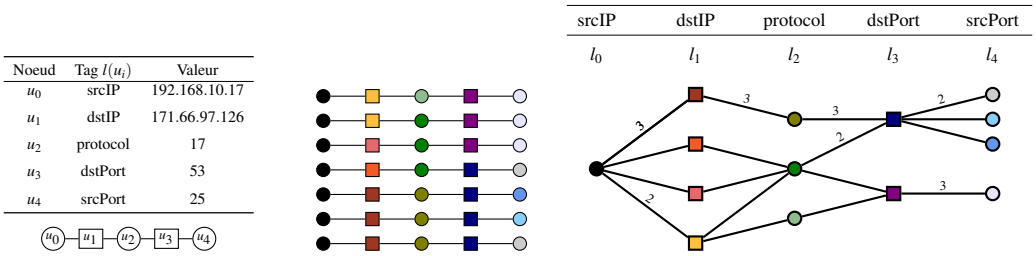


FIGURE 1: La fonction ϕ projette les données dans U vers un espace V, où ils sont linéairement séparables.



(a) Un flot réseau. (b) Collection de flots. (c) Graphe reflétant des schémas de communication.

FIGURE 2: Instance de génération d'un graphe d'activité lorsque $f = 4$.

De ce point de vue, les noyaux de graphes fournissent une riche connexion entre la théorie des graphes et l'apprentissage automatique.

Motivations. La détection d'anomalies est une tâche cruciale en raison de son impact direct sur la croissance économique et le progrès technologique. Selon des études récentes, le coût des dommages causés par la cybercriminalité devrait atteindre 10,5 milliards de dollars par an d'ici 2025. Cette situation alarmante est démontrée par l'adoption de récentes réglementations européennes telle que le Cybersecurity Act.

La plupart des noyaux de graphes existants proviennent de domaines biologiques sans une perspective réseau. Ils sont définis au sein du cadre R-convolutionnel [5], où les graphes sont décomposés en sous-structures (e.g., graphlets, cycles), puis comparés en évaluant le nombre de leurs sous-structures communes. Par exemple, le noyau de Weisfeiler-Lehman (resp. du plus court chemin) compte les occurrences des sous-arbres d'une hauteur donnée (resp. paires de plus courts chemin) dans le graphe. Comme les sous-arbres (resp. courts chemins) ne peuvent considérer que les (resp. ne tiennent pas comptes des) structures de voisinage des sommets, la similarité entre graphes n'est saisie qu'à des granularités grossières (resp. fines). Par conséquent, motivés par la popularité des noyaux de graphes dans d'autres domaines, notre travail vise à combler le fossé entre les deux communautés en adaptant ces méthodes pour détecter les anomalies.

Contributions. Les composantes de notre système d'apprentissage sont présentées en section 2. A partir d'un modèle graphique de profilage réseau défini en section 2.1, nous développons des algorithmes avec des garanties théoriques en sections 2.2 et 2.3 afin d'assurer une extraction rapide et précise des caractéristiques. Ceci nous permet d'introduire en section 2.4 un nouveau noyau de graphe adapté au profilage réseau. Enfin, nous évaluons l'efficacité de *Nadege* en section 4 à détecter les anomalies et classifier le trafic.

2 Système d'apprentissage

2.1 Graphes d'activités

Nous modélisons le profil d'activité d'un hôte [†] par un graphe dit *graphe d'activité* construit comme suit. L'observation du trafic sortant et entrant d'un hôte donne lieu à une collection de flots $\mathcal{B}$, où chaque flot est un chemin $P_f = (u_0, ..., u_f)$ attribué par des tags réseau $l(u_i)$ $(0 \leq i \leq f)$ choisis selon le protocole informatique considéré. Ensuite, nous disposons un graphe défini par des couches $l_0, ..., l_f$ telle que chaque couche l_i contient l'union disjointe des noeuds de la collection $\mathcal{B}$ ayant le tag $l(u_i)$. Enfin, une arrête pondéré (e_{uv}, p_e) est attachée entre deux sommets u, v du graphe s'ils sont précédemment voisins dans un flot dans $\mathcal{B}$, avec le poids p_e correspondant à la fréquence d'apparition de chaque arrête e_{uv} à travers $\mathcal{B}$.

Par conséquent, chaque flot crée un chemin partant toujours de l'adresse IP source de l'hôte sur la première couche l_0 et traversant les identifiants de chaque couche du graphe. Nous remarquons que tout graphe d'activité possède une structure bipartie représentée par une partition (S, T). Un exemple est fourni en Fig. 2 où $S = \bigcup_{i \equiv 0[2]} l_{i+1}$ (resp. $T = \bigcup_{i \equiv 1[2]} l_{i+1}$) est illustrée par des noeuds circulaires (resp. carrés). En outre, ces graphes peuvent être utilisés en conjonction avec plusieurs protocoles fonctionnant sur TCP/IP pour divers environnements comme le protocole MQTT pour l'Internet des objets (IoT), augmentant ainsi leur diamètre tout en restant expressifs.

†. Le terme "hôte" désigne un équipement réseau de communication, tandis que le terme "nœud" indique les sommets d'un graphe d'activité.

Notations. Notons dorénavant un graphe d'activité par $\mathcal{A} = (V_\mathcal{A}, E_\mathcal{A})$ et par s (resp. t) la taille de l'ensemble S (resp. T) définissant la partition de $V_\mathcal{A}$, tel que $n = |V_\mathcal{A}| = s+t$. Pour $\mathcal{A}$, notons par D sa matrice des degrés, L_{st} sa matrice de biadjacence induite par (S,T), J sa matrice d'adjacence pondérée, et $\widetilde{J} = D^{-\frac{1}{2}} J D^{-\frac{1}{2}}$ sa matrice d'adjacence normalisée. D_s (resp. D_t) dénote les premières s (resp. dernières t) lignes de D et $\mathcal{F}$ un ensemble de graphes d'activités. Enfin, $\odot$ désigne le produit de Hadamar et $\oplus$ l'opérateur de concaténation.

2.2 Empreinte intrinsèque

Afin d'avoir une caractérisation locale-globale sur l'environnement topologique de chaque nœud de $\mathcal{A}$, nous capitalisons sur les méthodes de marche aléatoires pour extraire des caractéristiques basées sur la propriété d'isomorphisme-invariance des probabilités de retour des marches aléatoires. L'efficacité de ces caractéristiques repose sur leur convergence à une certaine valeur connue sous le nom de probabilité stationnaire π dans la théorie des chaînes de Markov [4]. Cependant, une distribution stable n'existe pas pour les graphes d'activité en raison de leur bipartité. Pour contourner cela, nous explorons le graphe avec une marche aléatoire ergodique $\widehat{W}$ où à chaque instant, avec une probabilité de 1/2, la marche reste sur son noeud actuel, sinon elle saute uniformément vers l'un de ses voisins.

Definition 1. *Soit $\tau > 0$. L'empreinte intrinsèque de $\mathcal{A}$ est l'ensemble de vecteurs $i_\mathcal{A} = \{i_v, v \in V_\mathcal{A}\}$ où $i_v = \left[R^1_\mathcal{A}[v,v], ..., R^{\tau+1}_\mathcal{A}[v,v]\right]^T$ avec $R_\mathcal{A} = \frac{1}{2}(I + JD^{-1})$ la matrice de transition induite par $\widehat{W}$ pour $\mathcal{A}$.*

L'empreinte intrinsèque fournit une riche information sur la structure du graphe et peut être vue comme une mesure de centralité de sous-graphes. En focalisant sur les marches fermées, elle capture l'interaction d'un nœud avec les sous-graphes qui l'impliquent. Par exemple, à la première couche l_0, l'empreinte permet de saisir la popularité d'un hôte en termes de nombre des autres hôtes avec lesquels il communique.

Algorithme. Un calcul direct de $i_\mathcal{A}$ coûte $O(\tau(s+t)^3)$ en calculant successivement les puissances de $R_\mathcal{A}$. Puisque seules les diagonales des matrices de transition $R^k_\mathcal{A}$ sont nécessaires, nous pouvons faire mieux en observant que : (i) $R^k_\mathcal{A} = \frac{1}{2^k}\sum_{i=0}^k \binom{k}{i}(JD^{-1})^i = \frac{1}{2^k}\sum_{i=0}^k \binom{k}{i} D^{\frac{1}{2}} \widetilde{J}^i D^{-\frac{1}{2}}$, (ii) $\widetilde{J} \triangleq \begin{bmatrix} 0 & N \\ N^T & 0 \end{bmatrix}$ où $N = D_s^{-\frac{1}{2}} L_{st} D_t^{-\frac{1}{2}}$. En utilisant une décomposition SVD : $N = U\Sigma V^T$ avec U, V des matrices réelles orthogonales et Σ une matrice diagonale avec des coefficients diagonaux réels non-négatives $\sigma_1 \geq ... \geq \sigma_{\mu:=\min(s,t)}$. Il suffit alors d'exprimer, pour chaque $k \in \{1, ..., \tau+1\}$: $i^{(k)} \triangleq \left[R^k[v_1,v_1], ..., R^k[v_n,v_n]\right]^T = \frac{1}{2^k}\sum_{j\equiv 0[2]}^k \binom{k}{j} Y^j$ où $Y^j = \left((U \odot U)W^{2j}\right) \oplus \left((V \odot V)W^{2j}\right)$ avec $W^j = (\sigma_1^j, ..., \sigma_\mu^j)^T$. $i_\mathcal{A}$ est calculée ainsi en $O(st\min(s,t) + (\tau+1)(s^2+t^2) + \tau(\tau+1)(s+t))$. Notons que la SVD peut être approximée en temps linéaire au nombre d'entrées non nulles d'une matrice, ce qui rend le calcul pratique très rapide en raison de la sparsité de L_{st}.

Clairement, les empreintes intrinsèques sont sensibles au choix de τ qui, intuitivement, correspond à la profondeur des explorations du graphe initiées à partir des nœuds. Nous avons besoin ainsi d'un compromis entre intensité d'exploration et informativité des caractéristiques extraites. A cette effet, le Théorème 1 nous indique à quelle distance explorer, tout en assurant une vue contrôlée sur la convergence vers la distribution stationnaire. Pour $\delta = \frac{1}{4}$, la portée τ équivaut au temps de mélange d'une chaine de Markov [4].

Theorem 1. *Soit $\mathcal{A}$ un graphe d'activité, $\delta \in (0,1]$. Si $\tau_\delta \geq \frac{2}{1+\frac{1}{\sqrt{st}}} \log(\frac{n}{\delta})$ alors $\nu(p_t, \pi) \leq \delta$ où $\nu(p,q) = \|p-q\|_1$ indique la variation totale et p_t la distribution de probabilité de la position de la marche aléatoire.*

2.3 Empreinte contextuelle

Il est primordial de considérer le contexte dans lequel les empreintes intrinsèques apparaissent. En effet, des hôtes distincts peuvent induire des graphes isomorphes ayant la même empreinte $i_\mathcal{A}$. Nous associons ainsi à chaque graphe une empreinte dite contextuelle $c_\mathcal{A}$, définie au sein d'un groupe d'hôtes et, qui correspond à un vecteur de fréquence de motifs particuliers pour discriminer l'activité d'un hôte donné.

Algorithme. L'empreinte $c_\mathcal{A}$ est calculée en exécutant une variante du test d'isomorphisme de Weisfeiler-Lehman (2-WL) [1] sur un graphe dit de fusion avec un voisinage adaptatif. Un graphe de fusion $\mathcal{M}_\mathcal{F} = (V_\mathcal{M}, E_\mathcal{M})$ est construit à partir de $\mathcal{F}$ en ajoutant deux noeuds s_u et t_u tel que $V_\mathcal{M} = \bigcup_{\mathcal{A}\in\mathcal{F}} V_\mathcal{A} \cup \{s_u, t_u\}$

Méthode	Dataset											
	CSE-CIC-IDS-18		**UGR'16**		**IoT-Bot**		**N-BaIoT**		**Tor-nonTor**		**VPN-nonVPN**		**CICAndMal-17**	
SP	0.7290	C_5	0.6476	C_6	0.7854	C_5	0.7872	C_6	0.6893	C_4	0.6516	C_4	0.8447	C_5
RW	0.6932	C_5	0.5780	$C_{6,5}$	0.5461	C_5	0.6018	$C_{6,5}$	0.7261	C_4	0.7082	$C_{4,6}$	0.7593	C_5
GR	0.9571	$C_{5,4}$	0.7804	$C_{6,5}$	0.6574	C_5	0.7630	C_6	0.7517	C_4	0.6490	C_4	0.7928	C_6
WL	0.9631	$C_{5,4}$	0.8505	C_6	0.8283	C_5	0.8500	$C_{6,5}$	0.8906	C_4	0.8391	$C_{4,5}$	0.8997	C_5
Nadege	**0.9989**	C_5	**0.8937**	C_6	**0.9986**	C_5	**0.9704**	$C_{6,5}$	**0.9860**	C_4	**0.9823**	$C_{4,5}$	**0.9791**	$C_{5,6}$

TABLE 1: Comparaison de scores F1. La seconde colonne pour chaque trace indique les meilleurs classifieurs.

et $E_{\mathcal{M}} = \bigcup_{\mathcal{A}\in\mathcal{F}} E_{\mathcal{A}} \cup \{(s_u, v), (w, t_u), v \in l_0(\mathcal{A}), w \in l_4(\mathcal{A})\}$. Ceci nous permet d'exploiter la capacité de 2-WL à décomposer implicitement un graphe en ses composantes triconnectées [3]. Un potentiel avantage réside dans l'identification de communautés de nœuds, correspondant par exemple à des clients avec des intérêts similaires ou des membres d'une application collaborative. **Approximation.** Comme le schéma de propagation de 2-WL coûte $\Omega(n^2)$, nous équipons *Nadege* d'un algorithme d'approximation des empreintes contextuelles basé sur un échantillage de graphes d'activités parmi ceux dans $\mathcal{F}$.

2.4 Noyau de graphe à base d'empreintes

Chaque graphe $\mathcal{A}$ dans $\mathcal{F}$ est representé par son ensemble d'empreintes $r_{\mathcal{A}} = \{(i_v, c_{\mathcal{A}})\}_{v\in V_{\mathcal{A}}}$. Le noyau de graphe à base d'empreintes K est défini alors comme suit : $K(\mathcal{A}_1, \mathcal{A}_2) = \frac{1}{n_1 n_2} \sum_{v,u} (\frac{\|i_v - i_u\|^2}{2\sigma^2}) \exp(\frac{-\|c_{\mathcal{A}_1} - c_{\mathcal{A}_2}\|^2}{2\sigma^2})$, avec un paramètre de lissage $\sigma > 0$ et $\mathcal{A}_1, \mathcal{A}_2$ deux graphes d'activités.

3 Evaluation

Setup. Nous évaluons les performances de *Nadege* sur sept datasets récents couvrant différents types d'environnements réseau. Nous comparons avec quatre méthodes de noyaux de graphes standards, à savoir les noyaux Shortest-Path (SP), Random-Walk (RW), Graphlet (GR) ainsi que Weisfeiler-Lehman (WL). **Apprentissage.** Nous surveillons le trafic en utilisant des fenêtres d'observation séquentielles et chevauchées pour imiter un modèle de fenêtre temporelle glissante. Chaque fenêtre donne lieu à un ensemble $\mathcal{F}$ de graphes caractérisés par une matrice de similarité $K_{\mathcal{F}}$ associée à un noyau de graphe donné. Nous introduisons ensuite K_S dans différents classifieurs à base de machines à vecteurs de support (SVM) selon deux modes d'exécution : *en ligne* $(C_1, \ldots, C_4)$ et *hors-ligne* (C_5, C_6). **Paramètres.** Nous avons fixé $\delta = 0.1$ $\varepsilon = 0.05$, et σ à $1/d_{att}$ avec d_{att} la dimension des attributs, en normalisant toutes les matrices de noyaux.
Résultats. Le noyau de graphe à base d'empreintes est puissant, comme le démontre *Nadege*, qui surpasse toutes les méthodes de base dans sept ensembles de données par des marges significatives. En outre, *Nadege* a donné les variances les plus stables (c'est-à-dire les écarts entre le maximum et le minimum) parmi toutes les méthodes. Nous signalons la supériorité de la version exacte de *Nadege* (sans approximation) au détriment, toutefois, des coûts élevés en termes de temps et de mémoire. La Table 1 implique que les modèles hors ligne (C_4) sont adaptés à la classification du trafic tandis que ceux en ligne (C_5, C_6) sont adaptés à la détection d'attaques avec C_6 adapté aux attaques lentes à basse fréquence comme les botnets. Ces observations nous permettent de préconiser *Nadege* comme un cadre d'apprentissage efficace et polyvalent.

4 Conclusion

Dans ce papier, nous avons proposé *Nadege*, un système d'apprentissage pour la détection d'anomalies et la classification du trafic basé sur un nouveau noyau de graphe adapté pour les réseaux. Une évaluation sur divers environnements montre que *Nadege* atteint un score de détection surpassant les noyaux existants.

Références

[1] V. Arvind et al. On weisfeiler-leman invariance. *Journal of Computer and System Sciences*, 113 :42–59, 2020.

[2] T. Hofmann, B. Schölkopf, and A. J. Smola. Kernel methods in machine learning. *The annals of statistics*, 2008.

[3] S. Kiefer and D. Neuen. The power of the weisfeiler-leman algorithm to decompose graphs. In *MFCS*, 2019.

[4] D. A. Levin and Y. Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

[5] S. V. N. Vishwanathan et al. Graph kernels. *The Journal of Machine Learning Research*, 11 :1201–1242, 2010.