Détection d'anomalies dans les flux de graphes et attaques d'empoisonnement

Fatma Zohra Khaoula Saadi*, Abd Errahmane Kiouche*,**
Karima Amrouche* Hamida Seba** Mohamed-Lamine Messai***

*Ecole nationale Supérieure d'Informatique (ESI) Oued Smar Alger Algérie
{gf_saadi, k_amrouche}@esi.dz,
https://www.esi.dz/
**Université de Lyon, Université Lyon 1, LIRIS UMR 5205 F-69622 France
{abd-errahmane.kiouche, hamida.seba}@univ-lyon1.fr
***Université de Lyon, Lyon 2, ERIC UR 3083, France
mohamed-lamine.messai@univ-lyon2.fr

Résumé. Le problème de détection d'anomalies dans les flux de graphes se pose dans de nombreuses applications comme la cyber-sécurité et la finance. Plusieurs méthodes sont proposées dans la littérature pour répondre à cette problématique. Cependant, la plupart de ces méthodes sont vulnérables aux attaques par empoisonnement qui consistent à compromettre le processus d'apprentissage en injectant des données corrompues lors de la phase d'initialisation ou d'entraînement afin d'altérer le modèle représentant le comportement normal du système. Dans ce travail, nous étendons une des méthodes, les plus récentes et les plus effectives, de détection d'anomalies pour résister à cette attaque. Nous procédons par hybridation en considérant une autre méthode de détection d'anomalies comme un filtre qui élimine les données empoisonnées.

1 Introduction

Les graphes dynamiques, ou en flux, représentent l'une des solutions les plus utilisées dans la modélisation des données produites par des systèmes qui évoluent dans le temps. Ce sont généralement des données volumineuses et inter-connectées que l'on retrouve dans plusieurs applications parmi lesquelles on peut citer : les réseaux sociaux, les systèmes informatiques, le Web et la biologie. Un graphe dynamique est un graphe dont la structure change avec le temps (i.e., à un instant t, on n'aura qu'une vision partielle sur le graphe). A chaque instant t, un évènement de suppression ou d'ajout d'un élément (nœud, arête, étiquette ou poids) du graphe pourrait avoir lieu.

Nous nous sommes intéressés, dans ce travail, à l'étude des détecteurs d'anomalies dans les flux de graphes. D'une manière générale, un détecteur d'anomalies identifie les éléments du système qui sont considérablement différents des autres éléments. L'existence de ces éléments peut être due à une activité anormale comme une cyber-attaque, comme elle peut représenter simplement un évènement intéressant pour le système étudié. Dans un flux de graphes, une

anomalie peut avoir plusieurs formes : une arête qui ne devrait pas exister entre deux noeuds, un nœud diffèrent des autres noeuds, un sous-graphe plus dense que le reste du graphe, etc.

Les algorithmes d'apprentissage, généralement utilisés pour la détection d'anomalies, reposent sur le fait que les jeux de données sont fiables. Cependant, des études récentes de Kang et al. (2018) et He et al. (2017) soulignent l'augmentation des données biaisées qui peuvent altérer de manière malveillante les données pour induire en erreur les modèles d'apprentissage automatique. L'efficacité de ces algorithmes dépend de la véracité des données collectées pour avoir des modèles fiables. Par conséquent, les détecteurs d'anomalies représentent des cibles potentielles des cyber-attaquants dont le but est de perturber l'apprentissage et de causer son dysfonctionnement. Parmi les attaques qui menacent ces détecteurs, nous nous intéressons particulièrement à l'attaque par empoisonnement qui a pour but d'éviter la détection ou de dégrader les performances du système en injectant des données malveillantes.

Dans cet article, nous nous intéressons à l'impact de l'attaque par empoisonnement sur les détecteurs d'anomalies dans un flux de graphes. Dans un premier temps, nous montrons la vulnérabilité de la méthode F-Fade, récemment proposée par Chang et al. (2021), à l'attaque par empoisonnement, puis nous proposons dans un deuxième temps notre solution nommée F-Fade+ pour éviter cette attaque. Le reste de l'article est organisé comme suit : la Section 2 discute et classifie les détecteurs d'anomalies dans les flux de graphes. Ensuite, la Section 3 discute leur vulnérabilité à l'attaque d'empoisonnement. Dans le Section 4, nous présentons notre solution et son évaluation. La section 5 conclut l'article en donnant quelques perspectives.

2 Détection d'anomalies dans les flux de graphes

Les méthodes de détection d'anomalies dans un flux de graphes proposées dans la littérature peuvent être classées en quatre catégories selon l'approche utilisée :

— **Méthodes basées sur les probabilités :** Dans cette classe de méthodes de détection d'anomalies, l'idée de base est de construire un modèle qui décrit le comportement normal des données en analysant leur comportement dans les instants passés comme expliqué dans les travaux de Bhatia et al. (2020) et Ranshous et al. (2016). Ensuite, de déclarer toute déviation de ce dernier comme anomalie. Ces méthodes fournissent une solution intuitive et rapide et offrent la possibilité de contrôler le taux d'erreurs de l'algorithme. Cependant, les modèles construits ne retiennent pas assez d'informations sur la structure topologique du graphe qui reste assez importante pour la détection de certains types d'anomalies.

— **Méthodes basées sur les distances :** Dans cette classe, nous citons les travaux de Gaston et al. (2006), Eswaran et al. (2018) et Kiouche et al. (2019). Elles utilisent une métrique pour mesurer le degré de similarité ou de différence entre les objets du graphe. Plusieurs métriques existent, la différence entre elles réside dans la propriété du graphe sur la base de laquelle la métrique est fondée, i.e., le nombre de voisins, le diamètre du graphe, etc. Le choix de la métrique est primordial pour la qualité de la détection d'anomalies. Une bonne métrique est incrémentale, simple à calculer, sensible aux variations et permet la distinction entre les changements bénins et les changements anormaux.

— **Méthodes par décomposition :** Le principe des méthodes y appartenant, consiste à représenter les graphes par des matrices ou des tenseurs (Sensarma et Sarma, 2015)).

Puis, des algorithmes de décomposition sont appliqués sur ces matrices pour détecter les anomalies en interprétant les résultats ou en analysant les variations de l'erreur de reconstruction. Parmi ces algorithmes, nous pouvons citer SVD de Golub et Reinsch (1971), CUR de Drineas et al. (2006) ou CMD proposé par Sun et al. (2007). Ce dernier est plus adapté que les deux autres algorithmes aux graphes dynamiques caractérisés par leur *sparsité*. Ces méthodes permettent de capturer les informations structurelles nécessaires pour l'identification des anomalies. Cependant, leur utilisation d'algorithmes de factorisation les rendent lentes et de grande complexité de calcul. Dans cette classe, nous trouvons également les travaux de Yu et al. (2013) et de Sun et al. (2006).

— **Méthodes hybrides :** Les méthodes de cette classe combinent des techniques de plusieurs classes dans le but d'améliorer les performances. Nous citons les travaux de Chang et al. (2021) et de Aggarwal et al. (2011).

Le choix d'une méthode pour une application dépend de plusieurs paramètres tels que : le type du flux en entrée qui dépend des caractéristiques du système modélisé, le type d'anomalies qu'on cherche à détecter et comment on définit l'anomalie car il n'existe pas d'approches qui permettent de détecter toutes les anomalies. Il est aussi nécessaire de vérifier que l'application respecte les restrictions, souvent imposées par les algorithmes, afin de réduire la difficulté de représentation du comportement normal des éléments du système.

3 Vulnérabilité des détecteurs d'anomalies à l'empoisonnement

Les attaques par empoisonnement des données consistent à injecter des données corrompues dans l'ensemble d'entraînement. Le but est de forcer un algorithme, à une certaine itération de détection, à accepter une attaque comme élément normal du système, autrement dit, impacter la capacité de l'algorithme à effectuer des prédictions correctes.

Les détecteurs d'anomalies représentent des cibles potentielles des cyber-attaquants qui veulent faire passer inaperçues des attaques futures. Les méthodes de détection d'anomalies supervisées et semi-supervisées sont les méthodes les plus vulnérables à l'empoisonnement. En effet, dans le cas d'apprentissage supervisé (ou semi-supervisé), les cyber-attaquants peuvent facilement faire passer leurs attaques en empoisonnant les données d'entraînement par l'injection d'entrées malicieuses dans le corpus de données bénignes. Le but de ces injections est de subvertir le processus d'apprentissage afin que le système échoue à détecter certains types d'anomalies à l'avenir. Les méthodes non-supervisées partent de l'hypothèse que les anomalies sont rares et sont suffisamment différentes des données bénignes. Dans ce type de méthodes, le comportement bénin correspond au comportement majoritaire le plus observé et aucune phase d'entraînement n'est nécessaire. Cependant, avec l'apparition des données dynamiques, qui changement et évoluent au fil du temps, une phase d'initialisation est nécessaire même pour les méthodes non-supervisées afin de collecter les informations nécessaires pour construire les modèles d'évolution des données. Lors de cette phase aucune détection d'anomalie n'est faite et les algorithmes ne font qu'apprendre l'évolution des données. Cette faille rend les méthodes non-supervisées sur les données dynamiques vulnérables à l'attaque par empoisonnement. Nous considérons comme exemple d'étude dans ce travail la vulnérabilité de la méthode F-Fade (Chang et al., 2021) aux attaques d'empoisonnement.

F-Fade (Chang et al., 2021) est une approche non-supervisée dynamique de détection d'anomalies dans les flux d'arêtes qui utilise une nouvelle technique de factorisation de fréquence pour modéliser efficacement les distributions évolutives dans le temps des fréquences des interactions entre les paires de nœuds. Les anomalies sont ensuite déterminées sur la base de la vraisemblance de la fréquence observée de chaque interaction entrante. F-Fade est capable d'identifier dans un cadre de flux une grande variété d'anomalies avec des changements temporels et structurels, tout en ne nécessitant qu'un espace mémoire constant. Les expériences sur un graphe synthétique et six réseaux dynamiques du monde réel montrent que F-Fade atteint des performances de pointe et peut détecter des anomalies que les autres méthodes de la littérature sont incapables de trouver.

En raison de l'aspect dynamique des données traitées, F-Fade nécessite une phase d'initialisation pour établir les modèles des distributions évolutives des fréquences d'interactions bénignes entre les noeuds du flux. Nous avons étudié l'impact de la durée de cette phase, contrôlée par le paramètre t_{setup}, i.e., qui correspond à la durée de la phase d'initialisation, sur la performance du détecteur d'anomalies. La Figure 1 représente les performances de F-FADE en fonction de la durée d'initialisation. Nous avons utilisé le dataset Darpa (Ring et al., 2019) qui est l'un des datasets les plus populaires dans le domaine de détection d'intrusions. Nous avons calculé la métrique Area Under Curve (AUC) qui permet de mesurer la qualité d'un modèle. On remarque que les performances s'améliorent avec l'augmentation de la durée d'initialisation. À partir d'un certain seuil, les performances se stabilisent et atteignent une valeur maximale. Nous justifions cette amélioration de la performance par le fait que l'augmentation de cette durée permet d'apprendre davantage sur les distributions des fréquences d'interactions entre les noeuds et de construire, ainsi, un modèle plus précis.

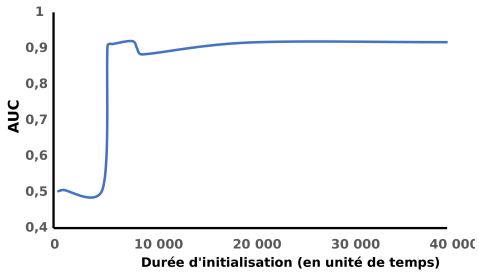

FIG. 1 – *Effet de la durée d'initialisation sur les performance de F-FADE.*

Lors de la phase d'initialisation, toutes les nouvelles arêtes qui arrivent sont utilisées directement, sans aucune vérification ni filtrage, pour construire les modèles des distributions évolutives des fréquence d'interactions. Bien que la durée d'initialisation soit généralement courte, une injection de données malicieuses durant cette phase est tout à fait concevable car aucune détection d'anomalies n'est effectuée et toutes les arêtes qui arrivent durant cette phase sont considérées comme fiables. L'injection de données va permettre de subvertir le processus de construction du modèle et affecter négativement la performance du détecteur d'anomalies. A partir de cela, nous avons constaté que F-Fade peut être vulnérable aux attaques d'empoison-

nement. Cette hypothèse a été confirmée par des expérimentations que nous avons effectuées sur les jeux de données suivants :

— **ISCX-IDS2012 (Ring et al., 2019)** : Créé en 2012, il consiste en des communications réseaux obtenues en capturant le trafic dans un environnement réseau émulé pendant une semaine. Il comporte différents types d'attaques comme l'attaque force brute SSH (*Secure Shell*), DoS (*Denial of Service*) ou DDoS (*Distributed Denial of Service*). On associe des labels aux communications atypiques pour l'évaluation des performances des détecteurs.

— **TwitterSecurity (Rayana et Akoglu, 2016)** : construit à partir de 2,6 millions de *tweets* collectés sur quatre mois (mai-août 2014) contenant des mots clés du département de la sécurité intérieure liés au Terrorisme ou à la sécurité intérieure.

Nous avons choisi une attaque DoS de chaque jeu de données et nous avons tenté de subvertir la classification de cette attaque pour qu'elle soit considérée comme étant normale. Afin de réaliser cela, nous avons injecté dans le flux, durant la phase d'initialisation, un ensemble de données corrompues. Cet ensemble est constitué de noeuds ayant de fortes fréquences d'interactions entre eux. Le but est de simuler, lors de la phase d'initialisation (où le système ne fait qu'apprendre), un comportement similaire à celui d'une attaque DoS afin que le système ne parvienne pas à détecter ce type d'attaques à l'avenir. Les résultats de cette expérience sur les deux jeux de données sont présentés dans la Table 1.

Jeu de données	seuil de détection	score d'anomalies avant empoisonnement	score d'anomalies après empoisonnement
ICSX	1200	2220.0	5.10×10^{-6}
Twitter	1459.361	3439.127	751.112

TAB. 1 – *Résultats des expériences d'empoisonnements.*

Dans les jeux de données ICSX et Twitter, comme le montre la Table 1, il y a eu une baisse drastique du score d'anomalies des arêtes ciblées, qui est attribué par la méthode F-Fade à chaque arête. Ces arêtes seront classées comme normales car leur nouveau score est inférieur au seuil de classement et donc l'objectif de l'attaquant est atteint. Cette baisse est due au fait que les données injectées ont compromis le modèle de distribution des fréquences d'interactions régulières construit par F-Fade lors de l'étape initialisation. En effet, cette attaque DoS devrait être considérée comme anormale car la fréquence de l'arête dont est composée l'attaque augmente brusquement. Cependant, l'insertion des données malicieuses durant la phase d'initialisation a fait augmenter considérablement la fréquence observée (régulière) d'interactions entre les deux noeuds composant l'arête de l'attaque ciblée, de sorte qu'elle ne sera pas détectée comme une anomalie dans le futur.

4 Solution et évaluation

Comme nous venons de voir dans ce qui précède, la méthode F-Fade est vulnérable aux attaques par empoisonnement et cette vulnérabilité a un impact négatif sur sa performance. De ce fait, nous allons, dans cette partie, proposer une solution qui permet de la rendre robuste face à ces attaques. Nous proposons de rajouter un filtre, appliqué aux données d'initialisation,

qui éliminera les données malicieuses qui pourraient être injectées par un attaquant. A l'arrivée d'une nouvelle arête durant la phase d'initialisation, le filtre lui affecte un score qui indique la probabilité qu'elle soit injectée. Ce score est ensuite comparé au seuil de filtrage $seuil_f$, défini par l'utilisateur, nous utiliserons un calibrage automatique pour trouver la valeur qui permet d'avoir les meilleures performances. Si le score est inférieur au seuil, l'arête est considérée comme normale et elle sera utilisée pour la construction du modèle. Dans le cas contraire, cette arête sera éliminée. Étant donné que les données injectées sont des arêtes anormales, notre filtre doit être une méthode de détection d'arêtes anormales dans un flux d'arêtes. A partir de l'étude bibliographique que nous avons effectuée et les différents expériences d'exploration des failles qui ont montré que l'attaque par empoisonnement est constituée d'attaques DoS, nous avons opté, d'utiliser comme filtre, la méthode MIDAS (Bhatia et al., 2020) dont nous allons rappeler le principe de fonctionnement dans ce qui suit. MIDAS est une approche de détection, en temps réel, des brusques poussées d'activités entre les nœuds d'un graphe. Elle est basée sur l'hypothèse statistique suivante : La moyenne des fréquences d'apparition d'une arête (u, v), à un instant t_i, est la même que la moyenne des fréquences d'apparition de la même arête à un instant $t < t_i$. Le score affecté par MIDAS aux arêtes est basé sur leurs fréquences d'apparition calculées par un test de χ^2. Des structures CMS (Count-Min Sketch) sont utilisées pour garder en mémoire les fréquences d'apparition de chaque élément du flux. moyennant des fonctions de hachage. MIDAS est rapide et les expériences effectuées par les auteurs ont montré qu'elle est la plus performante de sa catégorie mais elle reste moins performante que la méthode F-Fade. En résumé, notre solution est une hybridation de la méthode F-Fade et la méthode MIDAS qui jouera le rôle d'un filtre aux données d'initialisation afin d'éliminer les données malicieuses injectées par l'attaquant. Afin d'évaluer l'efficacité de notre solution, nous allons comparer les résultats obtenus par notre méthode, appelée F-Fade+, et ceux obtenus par F-Fade. Les expériences ont été faites sur les deux jeux de données que nous avons empoisonnés à savoir : Twitter et ICSX décrits dans la Section 3. En premier lieu, nous avons choisi une attaque DoS dans chaque jeu de données, et nous avons fait en sorte que l'attaque ne soit pas détectée par F-Fade en insérant des arêtes d'empoisonnement lors de la phase d'initialisation. La Table 2 montre que le score d'anomalies affecté aux arêtes de l'attaque DoS par F-Fade+ n'a pas changé lorsqu'on a injecté des arêtes d'empoisonnement. Ainsi, l'attaque par empoisonnement a échoué. Par conséquent, F-Fade+ permet de protéger le détecteur d'anomalies des attaques malicieuses.

Jeu de données	Seuil de détection	Approche	Score avant empoisonnement	Score après empoisonnement
ICSX	1200	F-Fade	2920.5	7.89×10^{-3}
		F-Fade+	2920.5	**2920.5**
Twitter	1459.361	F-Fade	3517.0	1098.0
		F-Fade+	3517.0	**3517.0**

TAB. 2 – *Score d'anomalies des deux approches.*

Dans le but de confirmer l'efficacité de F-Fade+, nous avons injecté dans les deux jeux de données cités précédemment des arêtes, ciblant 10 attaques choisies aléatoirement. La Table 4 rassemble les résultats de l'exécution des deux approches sur ces jeux de données.

Jeu de données	approche	% de détection	Temps d'exécution (s)
ICSX	F-Fade	0%	115.52
	F-Fade+	**100%**	120.09
Twitter	F-Fade	0%	1017.42
	F-Fade+	**100%**	1023.24

TAB. 3 – *Les performances des méthodes par jeu de données.*

Les résultats présentés dans les Tables 3, 2 et 4 montrent que F-Fade+, l'extension que nous proposons de F-Fade, permet de protéger le détecteur d'anomalies des attaques de type empoisonnement. En effet, les résultats obtenus montrent que le détecteur arrive bien à les détecter toutes les anomalies. Cette amélioration s'explique par la capacité du filtre, que nous avons rajouté, à éliminer toutes les arêtes insérées. Nous notons également, que ce filtre n'alourdit pas la méthode. En effet, F-Fade+ garde un temps d'exécution comparable à celui de F-Fade.

5 Conclusion

Dans cet article, nous avons abordé la vulnérabilité des détecteurs d'anomalies aux attaques par empoisonnement. Nous avons proposé une méthode de sécurisation d'un détecteur d'anomalies récent qui est vulnérable à ce type d'attaques. Notre solution consiste à filtrer les données d'initialisation en utilisant une autre méthode de détection d'anomalies. Les expérimentations que nous avons effectuées sur des jeux de données réels, que nous avons empoisonnés en insérant des données malicieuses, montre l'efficacité de notre approche. Ce travail se poursuit avec une étude plus globale de cette vulnérabilité avec des solutions plus générales applicables à un plus grands nombre de détecteurs d'anomalies ainsi qu'à d'autres attaques.

N. B. : Ce travail a été effectué dans le cadre du projet ANR Gladis ANR-20-CE39-0008.

Références

Aggarwal, C. C., Y. Zhao, et S. Y. Philip (2011). Outlier detection in graph streams. In *2011 IEEE 27th International Conference on Data Engineering*, pp. 399–409. IEEE.

Bhatia, S., B. Hooi, M. Yoon, K. Shin, et C. Faloutsos (2020). Midas : Microcluster-based detector of anomalies in edge streams. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pp. 3242–3249.

Chang, Y.-Y., P. Li, R. Sosic, M. H. Afifi, M. Schweighauser, et J. Leskovec (2021). F-fade : Frequency factorization for anomaly detection in edge streams. WSDM '21, New York, NY, USA, pp. 589–597. Association for Computing Machinery.

Drineas, P., R. Kannan, et M. W. Mahoney (2006). Fast monte carlo algorithms for matrices iii : Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing 36*(1), 184–206.

Eswaran, D., C. Faloutsos, S. Guha, et N. Mishra (2018). Spotlight : Detecting anomalies in streaming graphs. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '18, pp. 1378–1386.

Gaston, M. E., M. Kraetzl, et W. D. Wallis (2006). Using graph diameter for change detection in dynamic networks. *Australasian Journal of Combinatorics 35*, 299–311.

Golub, G. H. et C. Reinsch (1971). *Singular value decomposition and least squares solutions*, pp. 134–151. Springer.

He, Y., G. J. Mendis, et J. Wei (2017). Real-time detection of false data injection attacks in smart grid : A deep learning-based intelligent mechanism. *IEEE Transactions on Smart Grid 8*(5), 2505–2516.

Kang, J.-W., I.-Y. Joo, et D.-H. Choi (2018). False data injection attacks on contingency analysis : Attack strategies and impact assessment. *IEEE Access 6*, 8841–8851.

Kiouche, A. E., K. Amrouche, H. Seba, et S. Lagraa (2019). Une nouvelle approche pour la détection d'anomalies dans les flux de graphes hétérogènes. Volume Extraction et Gestion des connaissances, RNTI-E-35, pp. 93–104.

Ranshous, S., S. Harenberg, K. Sharma, et N. F. Samatova (2016). A scalable approach for outlier detection in edge streams using sketch-based approximations. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pp. 189–197. SIAM.

Rayana, S. et L. Akoglu (2016). Less is more : Building selective anomaly ensembles. *ACM Trans. Knowl. Discov. Data 10*(4).

Ring, M., S. Wunderlich, D. Scheuring, D. Landes, et A. Hotho (2019). A survey of network-based intrusion detection data sets. *Computers & Security 86*, 147–167.

Sensarma, D. et S. S. Sarma (2015). A survey on different graph based anomaly detection techniques. *Indian Journal of Science and Technology 8*(31), 1–7.

Sun, J., D. Tao, et C. Faloutsos (2006). Beyond streams and graphs : Dynamic tensor analysis. KDD '06, New York, NY, USA, pp. 374–383. Association for Computing Machinery.

Sun, J., Y. Xie, H. Zhang, et C. Faloutsos (2007). Less is more : Compact matrix decomposition for large sparse graphs. In *In Proc. SIAM Intl. Conf. Data Mining*, pp. 366–377.

Yu, W., C. C. Aggarwal, S. Ma, et H. Wang (2013). On anomalous hotspot discovery in graph streams. In *2013 IEEE 13th International Conference on Data Mining*, pp. 1271–1276.

Summary

The problem of detecting anomalies in graph streams arises in many applications such as cybersecurity and finance. Several methods are proposed in the literature to deal with this problem. However, most of these methods are vulnerable to poisoning attacks, which consist in compromising the learning process by injecting corrupted data, during the initialization or training phases, to alter the model representing the normal behavior of the system. In this work, we extend one of the most recent and effective anomaly detection methods to deal with this attack. We proceed by hybridization by considering another method of anomaly detection as a filter which eliminates empoisoned data.