Détection avancée de malwares dans les dépôts de code à l'aide des réseaux de neurones de graphes (GNN)

Malak Gouasmia*, Abd Errahmane Kiouche**
Hamida Seba**

*École nationale Supérieure d'Informatique (ESI) Oued Smar Alger Algérie
jm_gouasmia@esi.dz
** Universite Claude Bernard Lyon 1, CNRS, INSA Lyon,
LIRIS, UMR5205, 69622 Villeurbanne, France
abderrahmane.kiouche@gmail.com, hamida.seba@univ-lyon1.fr

1 Description

La détection de malwares, bien que largement étudiée dans le domaine de la cybersécurité, reste un défi complexe. Les méthodes traditionnelles, comme l'analyse statique ou dynamique sont très limitées. Pour y remédier, des approches comme *Repo2Vec* (Rokon et al., 2021), combinant les métadonnées, la structure des répertoires et le code source, ont apporté des améliorations notables. Dans ce travail, nous proposons une approche, nommée *Repo2Graph* qui exploite la puissance des *Graph Neural Networks (GNN)* pour la détection de malwares dans les dépôts de code source. Contrairement aux méthodes traditionnelles, elle s'appuie sur une modélisation approfondie des interactions internes des programmes.

La Figure 1 illustre l'architecture globale de notre approche. Le processus commence par l'extraction des graphes d'appels de fonctions (ou classes / modules) à partir du code source, où chaque nœud représente une fonction et chaque arête représente un appel entre fonctions. Ensuite, il consiste à transformer chaque fonction, classe ou module de code en un vecteur numérique pour qu'il soit le vecteur caractéristique associé à chaque nœud du graphe. À cet effet, nous utilisons *CodeBERT* (Feng et al., 2020) , un modèle de *Transformer* spécifiquement conçu pour le traitement du code source. En parallèle, les métadonnées du dépôt de code (telles que les fichiers README) sont collectées, puis converties en vecteurs numériques à l'aide de méthodes de plongement du langage naturel. Ces deux types de données (contenu du code et métadonnées) sont ensuite fusionnés pour enrichir la représentation globale du dépôt de code. Le modèle GNN est appliqué aux graphes d'appels, passant par plusieurs couches de convolution qui permettent de capturer les relations à différents niveaux (locaux et globaux) au sein du graphe. Ce processus génère une représentation vectorielle compacte des interactions fonctionnelles, qui est ensuite combinée avec les vecteurs de métadonnées. Enfin, la représentation agrégée est passée à un classifieur qui prédit si le dépôt de code est malveillant ou bénin.

La Table 1 présente une comparaison des performances entre les approches *Repo2Graph* et *Repo2Vec* (Rokon et al., 2021). Les résultats montrent une nette supériorité de *Repo2Graph*. avec des scores F1 jusqu'à 0.98.

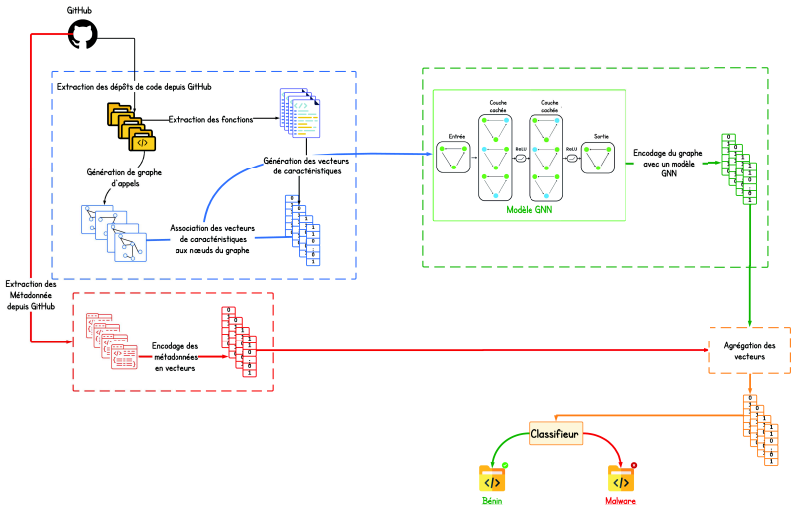

FIG. 1 – *Architecture globale de Repo2Graph*

Dataset	Approche	Exactitude	Précision	Rappel	F1-Score
Botnet	**Repo2Graph**	**0.97**	**1.0**	**0.94**	**0.97**
	Repo2Vec	0.97	1.0	0.94	0.97
Backdoor	**Repo2Graph**	**0.84**	**0.79**	**1.0**	**0.89**
	Repo2Vec	0.64	0.62	1.0	0.77
KeyLogger	**Repo2Graph**	**0.87**	**0.86**	**0.94**	**0.90**
	Repo2Vec	0.55	0.60	0.81	0.68
Trojan	**Repo2Graph**	**0.84**	**0.77**	**0.94**	**0.85**
	Repo2Vec	0.55	0.51	1.0	0.68
Virus	**Repo2Graph**	**0.90**	**0.92**	**0.89**	**0.90**
	Repo2Vec	0.57	0.55	1.0	0.71

TAB. 1 – *Comparaison des performances sur les différents types de malwares*

Références

Feng, Z., D. Guo, D. Tang, N. Duan, X. Feng, M. Gong, L. Shou, B. Qin, T. Liu, D. Jiang, et al. (2020). Codebert : A pre-trained model for programming and natural languages. *arXiv preprint arXiv :2002.08155*.

Rokon, M. O. F., P. Yan, R. Islam, et M. Faloutsos (2021). Repo2vec : A comprehensive embedding approach for determining repository similarity. In *2021 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pp. 355–365. IEEE.