ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

Inductive Lateral Movement Detection in Enterprise Computer Networks

Corentin Larroche

French Cybersecurity Agency (ANSSI)
51 boulevard de La Tour-Maubourg, 75700 Paris 07 SP - France

Abstract. Lateral movement is a crucial phase of advanced cyberattacks, during which attackers propagate from host to host within the targeted network. State-of-the-art methods for detecting this behavior rely on graph-based learning algorithms, which typically leverage node embeddings to detect anomalous edges between hosts. Once trained, such models cannot easily generalize to new hosts joining the network or to a different network, which is impractical in real-world applications. We investigate the detection performance of an inductive link prediction model, which can generalize to graphs not seen during training, and find that it performs as well as state-of-the-art transductive methods in a zero-shot setting. This opens promising perspectives for practical lateral movement detection.

1 Introduction

The most valuable assets within enterprise computer networks are usually not the most readily accessible to malicious intruders. As a consequence, advanced cyberattacks often comprise a lateral movement phase, during which the attacker initiates remote connections from already compromised hosts to more interesting ones. The standard approach to detecting such behavior represents internal traffic within an enterprise network as a graph whose nodes are the hosts. Lateral movement generates new edges in this graph, which are assumed to be anomalous with respect to benign edges. Therefore, a link prediction model can be used to distinguish legitimate new connections (which are well predicted by the model) from malicious ones. The models used in the cybersecurity literature typically learn parameters specific to each computer network, such as node embeddings summarizing hosts' behavior [1, 2, 3, 4, 5, 6]. As a consequence, a model trained on one enterprise network cannot easily be deployed in another one, and it must be retrained periodically to accomodate both the introduction of new hosts within the network and distribution shifts in the behavior of existing hosts.

Recent advances in inductive link prediction might help alleviate these limitations. Indeed, the advent of inductive graph neural networks (GNNs) has opened the possibility of training a model on one graph and using it for zero-shot prediction on another [7]. This approach was recently extended to knowledge graphs with the introduction of Ultra [8], a fully inductive GNN designed for knowledge graph completion. As described in Section 2, such models can be repurposed as lateral movement detectors. However, it remains unclear whether Ultra is fit for the specific, challenging task of detecting lateral movement. We thus study its performance on two benchmark datasets in Section 3, and find that Ultra performs comparably to state-of-the-art transductive methods.

ESANN 2024 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 9-11 October 2024, i6doc.com publ., ISBN 978-2-87587-090-2. Available from http://www.i6doc.com/en/.

2 Using Inductive GNNs for Lateral Movement Detection

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$ be a knowledge graph, where $\mathcal{V}$, $\mathcal{E}$ and $\mathcal{R}$ denote the node set, the edge set and the set of edge types (or relations), respectively. Here, the graph $\mathcal{G}$ represents the traffic observed within a computer network during a reference period: the nodes are hosts, the edges represent directed communication between these hosts, and the relations correspond to different types of communication (e.g., different protocols and ports). Given the reference graph $\mathcal{G}$ and a set of K new edges $\{(s_k, r_k, d_k) \notin \mathcal{E}\}_{k=1}^{K}$, where s_k (resp. d_k, r_k) is the source (resp. destination, type) of the k-th edge, lateral movement detection consists in sorting these edges from most to least suspicious.

An inductive link predictor such as Ultra can be described as a function $f : (s, r, d; \mathcal{G}) \mapsto h \in \mathbb{R}$, which computes a score h indicating how likely a new edge $(s, r, d) \in \mathcal{V} \times \mathcal{R} \times \mathcal{V}$ is given the graph $\mathcal{G}$. We build a lateral movement detector using this function as follows. In accordance with the methodology used for training Ultra [8], we first add reciprocal edges to the original graph: for each edge $(s, r, d) \in \mathcal{E}$, we create an additional edge (d, r^{-1}, s), where r^{-1} is the reciprocal of relation r. Given the context graph $\mathcal{G}$ and a new edge (s, r, d), we then define the anomaly score of the new edge as

$$z(s, r, d; \mathcal{G}) = \frac{1}{2}\left(\frac{\exp\left(f(s, r, d; \mathcal{G})\right)}{\sum_{d' \in \mathcal{V}} \exp\left(f(s, r, d'; \mathcal{G})\right)} + \frac{\exp\left(f(d, r^{-1}, s; \mathcal{G})\right)}{\sum_{s' \in \mathcal{V}} \exp\left(f(d, r^{-1}, s'; \mathcal{G})\right)}\right).$$

This anomaly score can be intuitively understood as the average of two conditional probabilities, namely the probability $p(d|s, r, \mathcal{G})$ of the destination being d and the probability $p(s|d, r, \mathcal{G})$ of the source being s. One or both of these probabilities being low means that the underlying traffic is inconsistent with the patterns found within the reference graph $\mathcal{G}$, and thus more suspicious from an intrusion detection perspective. Finally, we sort the set of new edges $\{(s_k, r_k, d_k)\}_{k=1}^{K}$ in ascending order of their anomaly scores.

Research questions. We aim to investigate the overall performance of this detection methodology, as well as the influence of several factors. First, we seek to determine **on which datasets the model should be trained** (**Q1**). Specifically, the authors of Ultra pre-trained several models on different sets of openly accessible knowledge graphs from various domains. Given the peculiarities of cybersecurity-related data, a reasonable question is whether these pre-trained models perform better than models trained on host communication graphs. Secondly, we investigate **how adaptable models trained on cybersecurity-related data are** (**Q2**). In other words, how well does a model trained on data collected within one computer network detect lateral movement in another network? Finally, an important aspect of network security monitoring is that it relies on many different data sources: in addition to traffic between hosts, security analysts typically collect other types of interactions, such as users logging on to hosts. Therefore, a relevant question is whether **including additional interactions in the context graph** $\mathcal{G}$ improves detection performance (**Q3**).

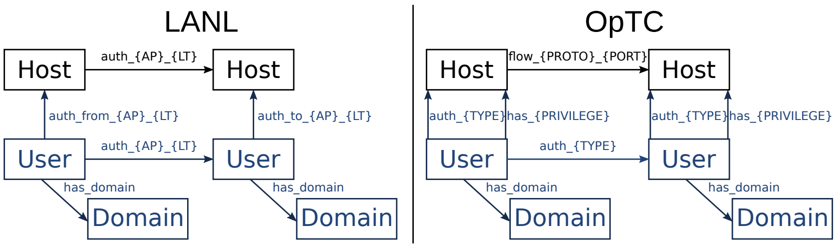

Fig. 1: Knowledge graph representations of the two datasets. The basic representation contains only the black elements, while the enriched representation also comprises the blue elements. AP and LT stand for authentication package and logon type, respectively.

3 Experiments and Results

We address the research questions exposed in the previous section through several experiments, which are described in Section 3.1. Our results are presented and discussed in Section 3.2. The code, data and configurations (including hyperparameters) used in our experiments are openly accessible[1].

3.1 Datasets and Models

Datasets. We use two datasets in our experiments. The first one is the "Comprehensive, Multi-Source Cyber-Security Events" dataset released by the Los Alamos National Laboratory (**LANL** dataset [9]). It consists of authentication logs collected over 58 days in a real-world enterprise network, with labeled edges corresponding to lateral movement carried out during a red team exercise. The context graph $\mathcal{G}$ is built using the 40 days of data without red team activity, and the rest of the data makes up the test set. The second dataset is the Operationally Transparent Cyber dataset released by DARPA (**OpTC** dataset). It contains host audit logs from a simulated enterprise network collected over nine days, with several attack scenarios including lateral movement. We build the context graph over the first six days and use the last three days for evaluation. For both datasets, we build two versions of the context graph $\mathcal{G}$ (see Figure 1). The basic representation contains only host-host edges: remote authentications for the LANL dataset, and flow start events for the OpTC dataset. In the enriched representation, we add user-related information from the authentication logs. Note that only new host-host edges are included in the test set.

Experiments. We perform three kinds of experiments: zero-shot lateral movement detection with the pre-trained models (Ultra3g, Ultra4g, and Ultra50g), fine-tuning of these models on the considered datasets and training of new Ultra models from scratch. The models trained from scratch have

[1] https://github.com/cl-anssi/UltraLMD

	Basic representation		Rich representation	
Model	AUC	AP	AUC	AP
Ultra3g	88.4	2.2	90.8	5.7
Fine-tuned on LANL	89.9±1.8	11.6±6.3	81.6±1.7	1.5±0.2
Fine-tuned on OpTC	87.9±3.7	11.8±6.6	94.1±2.4	12.3±3.9
Ultra4g	98.4	25.6	94.4	17.6
Fine-tuned on LANL	92.5±1.3	14.2±7.7	86.3±4.3	4.6±3.7
Fine-tuned on OpTC	91.5±1.4	12.4±5.8	96.9±0.9	18.2±4.4
Ultra50g	88.0	2.9	90.9	26.4
Fine-tuned on LANL	88.8±2.5	7.7±6.5	84.5±3.2	2.9±2.0
Fine-tuned on OpTC	91.1±3.6	14.1±10.0	90.0±2.3	8.4±4.9
Ultra (no pre-training)				
Trained on LANL	80.7±4.0	3.4±3.1	82.8±9.2	5.0±7.0
Trained on OpTC	75.4±8.7	2.6±3.8	74.6±12.5	1.1±0.3
HPF [5]	92.3±0.4	16.0±2.7	91.3±0.7	10.7±0.7
PTF [10]	90.4±2.6	3.8±1.6	88.1±2.9	3.1±1.3

Table 1: Results for the LANL dataset. For each metric, we report the mean and standard deviation over 10 runs (except for the pre-trained models).

the exact same architecture as the pre-trained models. Each fine-tuned or fully retrained model is trained on one of the two datasets and evaluated on both. Detection performance is evaluated using both the area under the ROC curve (AUC) and the average precision (AP), the latter giving a more realistic picture when lateral movement edges are scarce (which is especially true in the LANL dataset, where malicious edges account for less than 1% of new edges).

Baselines. We compare Ultra models with two transductive models used in the lateral movement detection literature, namely hierarchical Poisson factorization (**HPF** [5]) and Poisson tensor factorization (**PTF** [10]). The former does not take the edge types into account, while the latter does. Both models rely on node embeddings, the dimension of which is set by maximizing the predicted probability of a held-out validation set. For each dataset, the baselines are trained on the context graph and evaluated on the test set, providing a realistic picture of how a standard lateral movement detector would perform.

3.2 Results and Discussion

The results of our experiments on the LANL and OpTC datasets are displayed in Tables 1 and 2, respectively. Several points stand out: the pre-trained models perform well overall, their performance improves when enriching the knowledge graph representation of the data, fine-tuning them leads to contrasted results and training an instance of Ultra from scratch on the benchmark datasets yields inferior performance. The next paragraph further discusses each one of these points and links them to the research questions highlighted in Section 2.

	Basic representation		Rich representation	
Model	AUC	AP	AUC	AP
Ultra3g	51.7	10.4	56.6	11.2
Fine-tuned on OpTC	53.6±2.5	10.9±0.5	56.5±5.7	12.3±2.7
Fine-tuned on LANL	56.7±4.0	12.0±1.0	58.0±5.3	12.1±1.5
Ultra4g	66.3	18.7	71.4	20.8
Fine-tuned on OpTC	60.6±7.7	13.4±2.9	56.5±4.6	13.3±2.2
Fine-tuned on LANL	54.8±3.2	11.4±0.7	65.1±6.6	17.6±4.6
Ultra50g	59.6	15.7	68.7	19.9
Fine-tuned on OpTC	61.0±4.8	13.1±1.7	56.6±6.8	12.3±1.8
Fine-tuned on LANL	63.8±3.6	15.7±2.0	75.4±5.1	23.0±5.0
Ultra (no pre-training)				
Trained on OpTC	64.3±5.9	13.7±1.6	45.0±11.8	10.0±2.5
Trained on LANL	70.4±8.9	22.0±11.3	58.2±11.3	14.2±5.4
HPF [5]	64.6±2.1	19.8±1.4	64.1±1.5	14.1±0.9
PTF [10]	76.1±0.6	25.2±0.6	72.2±6.6	18.9±2.2

Table 2: Results for the OpTC dataset. For each metric, we report the mean and standard deviation over 10 runs (except for the pre-trained models).

The most important outcome of our experiments is that **pre-trained models achieve competitive performance** in the zero-shot setting, especially with the enriched representation (this partly answers research questions **Q1** and **Q3**). Specifically, Ultra4g and Ultra50g outperform both baselines on the LANL dataset, while Ultra3g beats PTF but not HPF. As for the OpTC dataset, PTF performs best but Ultra4g and Ultra50g come in second. Overall, while pre-trained models do not systematically beat all baselines, the simple fact that they perform similarly without ever being trained on cybersecurity-related data is impressive. The inferior performance of Ultra3g relative to the other two suggests that pre-training on many diverse knowledge graphs is key to this result. Another notable outcome is that **pre-trained models benefit from enriched representations** of the data (**Q3**), the only exception being Ultra4g on the LANL dataset. In contrast, HPF and PTF do consistently worse with the rich representation. A plausible explanation is that Ultra relies on neural Bellman-Ford nets [7], which predict relations between two nodes using indirect paths between them. Adding more relations to the context graphs creates more indirect paths, leading to more reliable predictions. Finally and perhaps surprisingly, **training on lateral movement datasets does not consistently improve performance** (**Q1**, **Q2**). Specifically, fine-tuning alternatively degrades or slightly improves performance while Ultra models trained from scratch globally perform worse than pre-trained models. The variance of the results is also strikingly high, suggesting that the datasets used in our experiments might be too small to enable consistent training. Interestingly, using a model trained or fine-tuned on one dataset for inference on the other often works better than training on the target dataset (**Q2**), which further hints towards the importance of diversity in the training set.

4 Future Work

In light of our experiments, one promising lead for future work is integrating more data sources into the context graph. Note that for some data sources, the dyadic nature of knowledge graphs might be too restrictive: for instance, the start of a new process involves the program being started, its parent process, the user starting it and the host on which the process starts. Recent advances in inductive hypergraph completion [11] might help model such events more adequately. It would also be interesting to pre-train a model such as Ultra on many cybersecurity-related datasets: indeed, while our experiments highlight the importance of diversity in the training set, using more relevant knowledge graphs for pre-training might also lead to better performance. Finally, recent contributions on lateral movement detection [3, 6] factor in the dynamic nature of host communication graphs, combining GNNs with recurrent neural networks (RNNs) to make time-dependent predictions. As a simple way to adapt inductive knowledge graph reasoning to this setting, the context graph $\mathcal{G}$ could be built dynamically while leaving the model itself unchanged.

References

[1] Benjamin Bowman, Craig Laprade, Yuede Ji, and H. Howie Huang. Detecting lateral movement in enterprise computer networks with unsupervised graph ai. In *RAID*, 2020.

[2] Maksim E Eren, Juston S Moore, and Boian S Alexandro. Multi-dimensional anomalous entity detection via poisson tensor factorization. In *ISI*, 2020.

[3] Isaiah J King and H Howie Huang. Euler: Detecting network lateral movement via scalable temporal link prediction. In *NDSS*, 2022.

[4] Wesley Lee, Tyler H McCormick, Joshua Neil, Cole Sodja, and Yanran Cui. Anomaly detection in large-scale networks with latent space models. *Technometrics*, 64(2):241–252, 2022.

[5] Francesco Sanna Passino, Melissa JM Turcotte, and Nicholas A Heard. Graph link prediction in computer networks using Poisson matrix factorisation. *Ann. Appl. Stat.*, 16(3):1313–1332, 2022.

[6] Jiacen Xu, Xiaokui Shu, and Zhou Li. Understanding and bridging the gap between unsupervised network representation learning and security analytics. In *S&P*, 2024.

[7] Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal A. C. Xhonneux, and Jian Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. In *NeurIPS*, 2021.

[8] Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, and Zhaocheng Zhu. Towards foundation models for knowledge graph reasoning. In *ICLR*, 2024.

[9] Alexander D. Kent. Cybersecurity data sources for dynamic network research. In *Dynamic Networks in Cybersecurity*. Imperial College Press, 2015.

[10] Maksim E Eren, Juston S Moore, Erik Skau, Elisabeth Moore, Manish Bhattarai, Gopinath Chennupati, and Boian S Alexandrov. General-purpose unsupervised cyber anomaly detection via non-negative tensor factorization. *Digit. Threat.*, 4(1):1–28, 2023.

[11] Xingyue Huang, Miguel Romero Orth, Pablo Barceló, Michael M Bronstein, and İsmail İlkan Ceylan. Link prediction with relational hypergraphs. *arXiv preprint arXiv:2402.04062*, 2024.