Towards Next-Generation Cybersecurity with Graph AI

Benjamin Bowman
George Washington University

H. Howie Huang
George Washington University

Abstract

Cybersecurity professionals are inundated with large amounts of data, and require intelligent algorithms capable of distinguishing vulnerable from patched, normal from anomalous, and malicious from benign. Unfortunately, not all machine learning (ML) and artificial intelligence (AI) algorithms are created equal, and in this position paper we posit that a new breed of ML, specifically graph-based machine learning (Graph AI), is poised to make a significant impact in this domain. We will discuss the primary differentiators between traditional ML and graph ML, and provide reasons and justification for why the latter is well-suited to many aspects of cybersecurity. We will present several example applications and results of graph ML in cybersecurity, followed by a discussion of the challenges that lie ahead.

1 Introduction

Cybersecurity encompasses many distinct domains, for example: network security, endpoint security, malware detection, vulnerability analysis, spam detection, and many others. In each of these domains cybersecurity professionals are in a constant battle to stay ahead of the latest threats plaguing the cyber landscape, such as critical application vulnerabilities, the ever-present insider threat, and intrusion by state-sponsored threat actors. The longstanding approach to cybersecurity has relied heavily on reactionary tactics, and herd immunity, whereby known-bad signatures and indicators-of-compromise (IOCs) are shared across the community to stem emerging threats. Unfortunately, this reactionary approach is becoming exceedingly easy to circumvent thanks to the ease of creating new network infrastructure, changing domain names, and simple software modifications, which can, in many cases, render most signature-based detection techniques ineffective.

To address this problem, there has been a push to build less reactionary detectors of malicious activity. It is not surprising to see a myriad of machine learning (ML) techniques that have been developed in this area. Specifically, when there are large quantities of labeled data, supervised ML classifiers have made significant improvements in the domain of malware detection [26], spam detection [29], and others. However, for many domains, such as network security, where there is significantly less labeled data, and the labeled data that does exist may not generalize to all environments, it is not so easy to apply off-the-shelf traditional ML algorithms that produce real utility and actionable results.

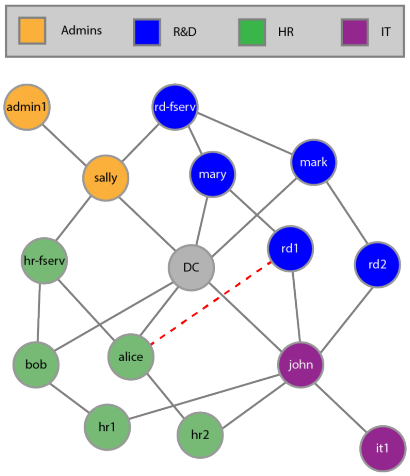


Figure 1: Example authentication graph for a simulated network. Colors correspond to organizational unit. A malicious authentication event is shown as the dashed red line.

It is our position that **graph-based machine learning (Graph AI) has the potential to make significant impact in the next-generation cybersecurity systems**. Graph AI [9] is a family of machine learning algorithms and techniques that are adapted to work on graph structured data, characterized by a set of nodes and edges. Figure 1 provides a simple example of network authentication activity in the form of a graph structure. In this example organization, there are several users distributed in various departments such as HR, R&D, and IT. In a nutshell, graph AI incorporates the relationships encoded by the graph data structure into the learning algorithms and

consequently the learned representations, allowing for the ability to capture critical relational patterns the may elude traditional ML techniques.

The remaining of the paper will be organized as follows. Section 2 will discuss the current state of ML in cybersecurity. Section 3 will provide several arguments and justifications for why Graph ML is a good fit for various domains in cybersecurity. Section 4 will discuss some the key challenges and future work in this domain. Section 5 will conclude.

2 State of ML in Cybersecurity

Advances in ML research have led to notable improvements in a number of application domains, especially those endowed with large quantities of labeled data which allows for relatively straightforward supervised learning. For example, VirusTotal [2] is a commercial service that provides the ability to scan unknown files and determine if any antivirus tools identify that file as malicious. It also provides the ability to monitor their feed of files, which provides a constant and up-to-date source for labeled malicious and benign code samples. Similarly, vulnerabilities in software and source code are often published openly in the form of bug reports, software patches, as well as in the National Vulnerabiltiy Database (NVD) [1]. This again provides a solid foundation for various ML techniques in the domain of vulnerability detection and malware detection.

Not all domains of cybersecurity have such easy access to labeled data samples, which makes the application of ML algorithms much more challenging. Unsupervised learning has been applied in various domains for outlier detection. For example Kitsune [22] is a deep model of stacked auto-encoders which identify anomalous network flow activity. Unfortunately anomaly detection techniques are often very noisy, generating many false positives, and suffer from the challenge of explainability.

One may argue that ML is not a good fit for cybersecurity outside of building improved signature detectors due to the lack of representative and generalizable labeled training data. Additionally, unsupervised techniques will often be plagued by explainability challenges in a field that almost always requires a human analyst to confirm or deny the existence of an event that requires remediation. These are two very serious challenges in this domain, however next we will explain why and how graph ML can alleviate some of these challenges.

3 Graph AI for Cybersecurity

3.1 Background

Traditional machine learning operates on a grid structure of data, where each row of that grid could represent pixel values of images, words of a sentence, or manually generated features from some observed physical phenomenon. The ML algorithm will process each observation and learn patterns of features that span the dataset.

In contrast, graph structured data does not directly fit this processing paradigm. The data encoded into a graph is relational, with the knowledge contained in the graph spanning any individual node or edge in isolation. In order to overcome this challenge, Graph ML has adopted two primary techniques for applying ML concepts to graph structured data:

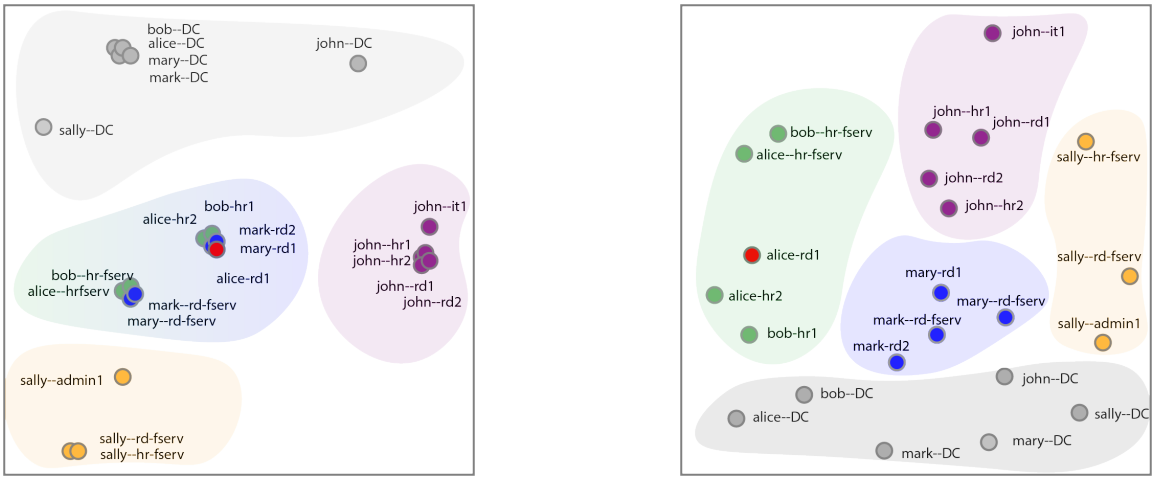
- **Walk-based sampling** is a technique whereby graph structured data is sampled via walks through the graph [7, 25, 27]. In the simplest case, random walks are performed which converts the unstructured graph data, into structured sequences of nodes and edges. These sequences of nodes and edges can then be processed by more traditional ML techniques, such as skip-gram models [21] popularized in natural language processing.
- **Graph convolutions** are re-designed convolutional filters from well-known domains such as image processing [15], but applied to graph structured data [8, 13, 28] . At their core, they are simply a way to learn information about a node based on some function of their neighbors. This could be based on some simple mean or sum of neighboring features, or a more complex recurrent neural network for example.

3.2 Observations

In this subsection, we will present four main observations as to why we believe graph AI will make significant contributions in the domain of cybersecurity.

Observation #1: Cybersecurity data is inherently relational, graph structured data. Unlike some traditional ML domains such as image or speech recognition, cybersecurity data almost always involves a set of interacting entities. These entities could be users on a network, processes on a server, code segments in a program, etc. In each case, capturing the information as atomic observations that would be suitable for a traditional grid-based ML, is not straightforward and would likely cause significant loss of information.

For example, consider the very simple act of network authentication within an enterprise computer network. Figure 1 provides the representation of the authentication activity, which happens to be most-easily displayed in a graph-based representation. Traditional machine learning would likely consider each authentication event as an individual observation based on some features of the event. A simple approach to representing this information in a grid-based structure would be to simply 1-hot encode each authentication event based on what entities are involved. Figure 2a shows the embedding space generated by the first two principal components based on this representation of the data. We can see that this data representation did capture some coarse-grained relationships of the data, such as the DC (domain controller) authentication



(a) PCA embedding based on traditional non-graph features

(b) PCA embedding based on node2vec embedding

Figure 2: Comparison between non-graph and graph embedding

activity, IT activity, and admin activity. However, it fails to distinguish between the finer-grained activity between the R&D entities and the HR entities. Additionally, we can see the malicious authentication edge involving an HR user and a R&D workstation (*alice-rd1*) is co-located with the benign authentications involving both the user and the workstation (*alice-hr2*). This means downstream anomaly detectors would have essentially no way of differentiating this malicious authentication based on this data alone.

Observation #2: The Graph structure can help make up for lack of labeled data. In cybersecurity we suffer from a labeled data shortage, especially in areas tied to network and host-level defenses. It is not uncommon to see academic papers cite the DARPA 1999 dataset [16] which is undoubtedly extremely outdated, yet it is one of the few labeled datasets available with a variety of attacks and visibility into both network and host events. However, graph learning provides a unique ability to perform unsupervised or semi-supervised learning which can utilize the highly informative structural information captured in the graph to drive the learning algorithm. This means we can do much more than the traditional clustering or principal component analysis typically associated with unsupervised machine learning. In other words, we can train unsupervised models to perform tasks such as link prediction, or graph reconstruction, using the ground truth information derived from the graph structure.

For example, rather than 1-hot encoding our entities and embedding them as discussed in the previous example, we can leverage the topological data encoded in the graph structure displayed in Figure 1 and embed our entities with a graph ML algorithm such as node2vec [7]. Figure 2b shows the embedding space generated by applying the same PCA analysis as previous, however this time we are looking at the PCA of the edge embedding which are based on the concatenation of the node embeddings generated by node2vec. We can see that the embedding space in fact has several similarities, such as the spaces clearly tied to DC authentication, admin authentication, and IT authentication. However, we can see in this case there is a distinct separation between HR users and R&D users that was extracted from the topology of the graph. We can also see that the malicious authentication edge (*alice-rd1*) is embedded in a location that allows for the ability to identify this edge as an anomaly due to the fact that it is an authentication to a R&D entity which is far outside of the R&D cluster.

Observation #3: Many important security relationships lend themselves to graph learning. As mentioned in the first point, many critical security relationships will manifest as graph relationships. For example, user groups will form strongly connected components, critical services will have high in-degrees, administrators high out-degrees, etc. Because these relationships often have an impact on graph topology, graph learning techniques are well suited to learning these patterns from the data.

The running example involving a malicious authentication event from the user *alice* to the machine *rd1* could be an example of lateral movement during an APT-scale attack campaign. Lateral movement is a challenging attack phase to detect as they often involve valid credentials and use tools native to the environment (e.g., WinRM, WMI). This means detection will need to be based purely on behavioral characteristics of the entities involved in the authentication. In the example so far, we showed how the authentication graph can be used to derive a behavioral feature capable of identifying the lateral movement based on the topology of the graph.

Observation #4: Graph AI can leverage the native representation to provide useful visualization to accompany the model outputs. In cybersecurity, we rely heavily on our

analysts to make the final decisions and perform the necessary remediation when security events occur. Unfortunately, it is not uncommon for malicious activity to generate an alert, but be overlooked or handled inappropriately by the analysts defending the network. This is because traditional alerts based on signatures or traditional ML detectors require analysts to essentially stitch together the events into a compelling and believable incident report. When analysts are receiving hundreds to even thousands of low-level alerts per day, with the vast majority being false positives, it is no surprise that sometimes they fail to properly stitch together the events that are in fact true positives .We believe that graph AI has an inherent advantage here because graph structured data is highly intuitive to humans, and can serve as a very useful investigation, reporting, and remediation tool. This fact can be illustrated entirely by the way we chose to represent the simulated authentication information displayed in Figure 1. We could have just as easily displayed this data in a table containing source and destination entities, however this would not have been nearly as intuitive and interpretable as the graph representation.

3.3 Graph AI in Use

The examples here were intentionally simple to illustrate the idea, however this technique based on applying graph AI to authentication graphs for lateral movement detection has been evaluated further in our related work [6]. In that work, we expand these ideas and test our hypothesis and show how this technique can be used to detect lateral movement in large real-world networks with thousands of users and machines. One of our evaluation datasets was from Los Alamos National Labs [12], which contains user authentication events, as well as labeled malicious activity for a 58-day period.

We applied a graph AI algorithm by first building an authentication graph similar to the one discussed previously, and then applying node2vec [7], a walk-based embedding technique, which provided us with node embeddings that we could use to train a logistic regression link predictor using the ground-truth edge information in the graph structure. This gave us the ability to, for any pair of nodes in the graph, predict the likelihood of authentication.

We compared our graph AI technique against four other approaches. We compared with some simple rule-based analytics, such as detecting all authentication events never previously observed during the training period, or detecting all failed login events. We also compared with two traditional machine learning algorithms which do not incorporate the graph structure. Specifically, we built a traditional 1-hot encoded feature vector for each authentication event encoding what entities were involved in the authentication, and applied a local outlier factor analytic and isolation forest analytic in order to identify outliers. Table 1 displays the true positive rate (TPR) and false positive rate (FPR) for each of the detectors. We can see that, without the graph structure, the traditional machine learning techniques were not able to differentiate benign from malicious, for the reasons stated in the previous subsection. We can also see that the unknown authentication detector was able to detect many of the malicious authentication events, however with a false positive rate four times higher than the graph AI approach. The graph AI approach was able to achieve the best true positive rate, at the lowest false positive rate, making it the most effective analytic of the comparison works on this dataset.

Table 1: Anomaly Detection Results on LANL Dataset

Algorithm	**TPR (%)**	**FPR (%)**
Unknown Authentication	72	4.4
Failed Login	4	1.0
Local Outlier Factor	12	9.6
Isolation Forest	9	16.9
Graph AI	**85**	**0.9**

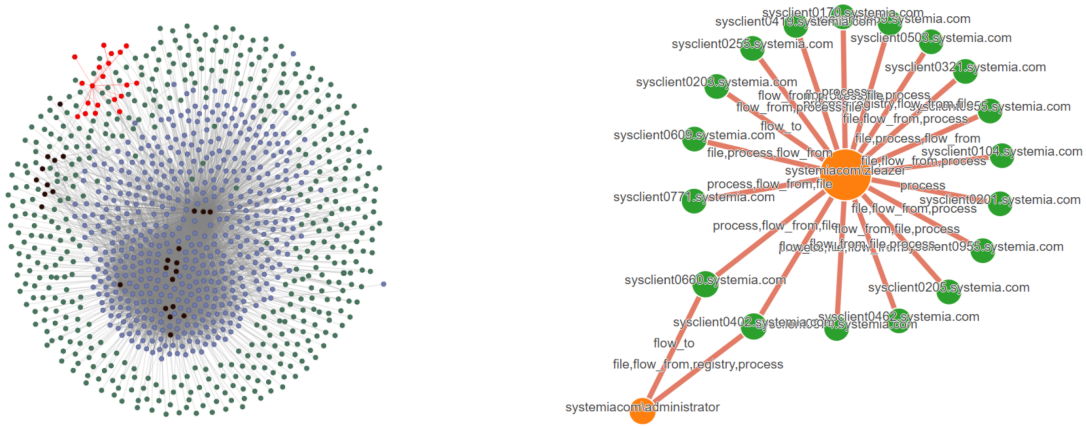
For full details on this experiment, as well as the many other experiments on this dataset and others, we recommend the reader to our full paper on this topic [6].

In addition to authentication activity, there is a large spectrum of additional details about how users interact with systems that we can incorporate into our graph, and ultimately into our analytics. This provides our algorithms with a more granular and comprehensive view of user activity, and ultimately produces a result with more utility. Figure 3a shows a graph-based representation of the day-1 activity from the OpTC dataset [3], which contains over 500 hosts and users. The graph nodes highlighted in red correspond to those entities that were detected as anomalous. Figure 3b shows a zoomed-in representation of the anomalous activity, which contains various relationships pertaining to file, process, network, and registry activity between users and systems. This type of analysis and result could be directly utilized by security analysts for detection purposes or threat hunting.

4 Challenges and Future Work

Graph AI has gained a lot of momentum recently, with graph-based machine learning techniques dominating many of the top ML conferences. Security research is beginning to adopt some of these techniques in domains such as code analysis [5, 10], network security [6], and APT detection [17]. However, applying these algorithms to real-world cybersecurity problems and datasets still requires overcoming many challenges, a few of which we discuss below.

Scalability. Graph AI and graph processing in general has a host of scalability challenges that do not apply to traditional ML techniques. As discussed previously, since graph data is not inherently grid-based, graph ML does not benefit equally to GPU-based training as does traditional ML. Scalable data



(a) Graph representation of the full network. Red nodes correspond to the anomalous activity detected.

(b) A zoomed-in representation of the anomalous activity. The user 'zleazer' had their credentials compromised leading to many anomalous and malicious edges spanning many different edge types.

Figure 3: Graph AI applied to the OpTC dataset exposes anomalous activity of a user on day-1 of a redteam campaign.

management and processing techniques developed for graph analytics, such as our prior work [4, 11, 14, 18, 19], can help shed some light in this area.

Dynamic Graphs. In cybersecurity graph data is often of a dynamic nature, with typically a temporal dimension, meaning that the graph algorithms will need to handle a changing graph. There are many recent advancements in performing time series graph learning [23,24,31] which may be applicable here.

Explainability. Although graph learning has the benefit of a intuitive graph data structure which can be used as an explanation tool, there is still the challenge of explaining what entities in the graph contributed the most to a particular model outcome. There has been some exploratory research which may provide utility on this front as well [20, 30].

5 Conclusion

The cybersecurity domain is ready for transformative ML/AI based technology that can accurately and explainably identify malicious and anomalous activity. The data in this domain is largely unlabeled, and unstructured, which makes it challenging to apply off-the-shelf AI/ML techniques. In this paper, we present some arguments for why and how graph AI can provide advantages over traditional ML techniques in this domain. We explain how cybersecurity data is inherently graph-structured, and fits well into a graph ML paradigm. We describe how graph-learning can leverage the topology of the graph to produce improved unsupervised learning results. We discuss how many cybersecurity problems manifest themselves as graph-based features and structures. Finally we discuss how the graph byproduct of graph learning is extremely useful in the form of an explanation tool to the cyber analysts who must take the algorithm results and turn them into action.

Acknowledgment

The authors thank the anonymous shepherd and reviewers for their help on improving this paper. This work was supported in part by DARPA under agreement number N66001-18-C-4033 and National Science Foundation CAREER award 1350766 and grants 1618706 and 1717774, as well as support from the ARCS Foundation. The views, opinions, and/or findings expressed in this material are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense, National Science Foundation, ARCS, or the U.S. Government.

References

[1] National vulnerability database. https://nvd.nist.gov.

[2] Virus total. https://www.virustotal.com/.

[3] Operationally transparent cyber data. https://github.com/FiveDirections/OpTC-data, 2021.

[4] Bibek Bhattarai, Hang Liu, and Howie Huang. Ceci: Compact embedding cluster index for scalable subgraph matching. In *Proceedings of the International Conference on Management of Data (SIGMOD)*, 2019.

[5] Benjamin Bowman and H. Howie Huang. Vgraph: A robust vulnerable code clone detection system using code property triplets. In *IEEE European Symposium on Security and Privacy (EuroSP)*, 2020.

[6] Benjamin Bowman, Craig Laprade, Yuede Ji, and H Howie Huang. Detecting lateral movement in enterprise computer networks with unsupervised graph {AI}. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2020)*, pages 257–268, 2020.

[7] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[8] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. *arXiv preprint arXiv:1706.02216*, 2017.

[9] William L. Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *IEEE Data Eng. Bull.*, 40(3):52–74, 2017.

[10] Yuede Ji, Lei Cui, and H. Howie Huang. Buggraph: Differentiating source-binary code similarity with graph triplet-loss network. In *ACM Asia Conference on Computer and Communications Security (ACM ASIACCS)*, 2021.

[11] Yuede Ji and H. Howie Huang. Aquila: Adaptive parallel computation of graph connectivity queries. In *International Symposium on High-Performance Parallel and Distributed Computing (HPDC)*, 2020.

[12] Alexander D. Kent. Comprehensive, Multi-Source Cyber-Security Events. Los Alamos National Laboratory, 2015.

[13] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[14] Pradeep Kumar and Howie Huang. Graphone: A data store for real-time analytics on evolving graphs . In *Proceedings of USENIX conference on File and storage technologies (FAST)*, 2019.

[15] S. Lawrence, C. L. Giles, Ah Chung Tsoi, and A. D. Back. Face recognition: a convolutional neural-network approach. *IEEE Transactions on Neural Networks*, 8(1):98–113, 1997.

[16] Richard Lippmann, Joshua W Haines, David J Fried, Jonathan Korba, and Kumar Das. The 1999 darpa offline intrusion detection evaluation. *Computer networks*, 34(4):579–595, 2000.

[17] Fucheng Liu, Yu Wen, Dongxue Zhang, Xihe Jiang, Xinyu Xing, and Dan Meng. Log2vec: a heterogeneous graph embedding based approach for detecting cyber threats within enterprise. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1777–1794, 2019.

[18] Hang Liu and H. Howie Huang. Simd-x: Programming and processing of graph algorithms on gpus. In *2019 USENIX Annual Technical Conference (USENIX ATC 19)*, 2019.

[19] Hang Liu and Howie Huang. Graphene: Fine-grained io management for graph computing. In *Proceedings of USENIX conference on File and storage technologies (FAST)*, Santa Clara, CA. February 27 - March 2, 2017. (Acceptance rate 28/116=24%).

[20] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. *Advances in Neural Information Processing Systems*, 33, 2020.

[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*, 2013.

[22] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. Kitsune: an ensemble of autoencoders for online network intrusion detection. *arXiv preprint arXiv:1802.09089*, 2018.

[23] Giang Hoang Nguyen, John Boaz Lee, Ryan A Rossi, Nesreen K Ahmed, Eunyee Koh, and Sungchul Kim. Continuous-time dynamic network embeddings. In *Companion Proceedings of the The Web Conference 2018*, pages 969–976, 2018.

[24] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. Evolvegcn: Evolving graph convolutional networks for dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5363–5370, 2020.

[25] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 2014.

[26] Konrad Rieck, Philipp Trinius, Carsten Willems, and Thorsten Holz. Automatic analysis of malware behavior using machine learning. *Journal of Computer Security*, 19(4):639–668, 2011.

[27] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077, 2015.

[28] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018.

[29] Alex Hai Wang. Detecting spam bots in online social networking sites: a machine learning approach. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pages 335–342. Springer, 2010.

[30] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32:9240, 2019.

[31] Wenchao Yu, Wei Cheng, Charu C Aggarwal, Kai Zhang, Haifeng Chen, and Wei Wang. Netwalk: A flexible deep embedding approach for anomaly detection in dynamic networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2672–2681, 2018.