# Node Embeddings for Botnet Detection

Brinton, Joseph
jcbrinton@gmail.com

Harker, Christopher
chris.harker77@gmail.com

Zhao, Jack
u1129777@utah.edu

April 27, 2020

## 1 Introduction

Botnets are a significant source of network attacks. They consist of several compromised computers capable of performing coordingated attaks, including DDoS attacks, click-fraud scams, spamming, and identity theft. A key component in preventing attacks from botnets requires methods to detect them.

Modern botnet detection methods still primarily rely on heuristics built upon the manual analysis of traffic patterns, malware code, etc. Machine learning approaches help automate the process, relieving some of the manual labor placed on analysts while showing promise in their ability to detect botnets. However, most machine learning algorithms require inputs defined in the euclidean domain. Therefore, when used on network data, machine learning algorithms fail to take into account crucial relational information between nodes in the network.

A new subfield of machine learning called Geometric Deep Learning attempts to address this concern by studying algorithms that can take non-euclidean data, such as graphs, as inputs. Geometric deep learning algorithms are usually divided into two categories, namely graph embedding algorithms and graph neural networks. Graph embedding algorithms attempt to learn latent representations of a graph's nodes, encoding relational information in a continuous vector space that can be used in traditional machine learning algorithms. Graph neural networks attempt to perform machine learning using the graph data structure itself as input.

Our project involves studying the node embeddings produced by two of the most commonly used graph embedding algorithms, DeepWalk and Node2Vec. Botnets generally have different structural properties than background networks. For example, centralized botnets have a strong hierarchical structre, while decentralized botnets are designed for efficient diffusion of information. If the botnet has different structural properties than the other nodes in the network, the node embeddings produced by DeepWalk and Node2Vec should be different than nodes elsewhere in the network.

We trained DeepWalk on the Aposemat [10] dataset, which contains three types of botnet: Mirai, Okiru and Torii. Since there are multiple types of botnet, it doesn't give us a clear result. We also trained both DeepWalk and Node2Vec on two scenarios provided as part of the CTU-13 dataset, both of which contain an IRC botnet. The embeddings of nodes in the botnet produced by both algorithms are compared to the embeddings of normal and background nodes by visually inspecting them as well as examining the pointwise euclidean and cosine distances between them. We find that both of these algorithms produce botnet embeddings that are resonably different from other nodes in the network.

## 2 Related Work

Representation Learning was popularized in Natural Language Processing [1],[4]. Interestingly, words that are similar to each other have similar embeddings while dissimilar words have dissimilar embeddings. The introduction of the word2vec Skip-gram model [5] provided more efficient and accurate vector representations. Researchers have applied these techniques in other networks, especially in biomedical networks [6],[9].

Node embeddings have been used in the context of network security, generally as features in supervised learning problems [8]. To the best of our knowledge, however, there is not any research examining the node embeddings themselves. We aim to close this gap, hoping that node embeddings of nodes inside the botnet are similar to eachother and dissimilar from nodes elsewhere in the network.

# 3 Methodology

## 3.1 Adversarial Model

It is assumed that the adversary is capable of performing DDoS attacks. Specifically, the adversary uses an IRC protocol and is capable of performing either UDP or ICMP flood attacks.

## 3.2 Data

We utilize two botnet scenarios provided as part of the CTU-13 dataset [2]. This dataset provides large captures of real botnet traffic mixed with normal traffic and background traffic. Each of the thirteen scenarios included in the dataset have different characterists. We focus on scenario 10 and scenario 11, both of which are IRC botnets performing DDoS attacks. Scenario 10 contains 10 nodes in the botnet, which is performing a UDP DDoS attack, and contains 6 normal nodes. Scenario 11 contains three nodes in the botnet, which is performing a ICMP DDoS attack, and contains six normal nodes. All other IPs in each dataset are background nodes.

Each dataset consists of botnet, normal, or background traffic. Each flow consists of a source IP address and a destination IP address. Using this flow data, we constructed a graph by placing an edge between any two IPs where there was a packet sent between them.

Another dataset we use is the IoT-23 dataset. This dataset is not specific for the botnet, it contains multiple types of malicious IoT network traffic. The result of that is the total size of this dataset is huge and it requests a tremendous computing time to train. Therefore, we try to reduce the dataset first before train the data. We separate the dataset into small groups and use some basic statistic techniques to count which part has the densest botnet and use that part for our final training.

## 3.3 Algorithms

### 3.3.1 DeepWalk

DeepWalk [7] was inspired by methods used in natural language modeling in which vector representations of words are produced by estimating the likelihood that a sequence of context words appear given a root word. In DeepWalk, "sentences" of nodes are generated by generating random walks on a graph and then maximizing the probability that the nodes appear given the source node. However, the probabilities are never used. Rather, a mapping function that maps a node to its latent representation and is learning during training, is used as the embeddings. Specifically, the embeddings are generated by training a Skip-gram model.

### 3.3.2 Node2Vec

Node2Vec [3] is similar to DeepWalk in the sense that it trains a Skip-gram model by using random walks on the graph. It differs from DeepWalk, however, by how the random walks are generated. Instead of selecting any of its neighbors uniformly at random as the next node in the walk, Node2Vec uses two hyperparameters to weight the edges, making some nodes more likely to be selected than others. Depending on the values of these two hyperparameters, the random walks emphasize the local structure around a node or a more global structure.

# 4 Experiments and Discussion

## 4.1 CTU-13 Dataset

Both DeepWalk and Node2Vec were trained on scenario 10 and scenario 11 from the CTU-13 dataset. DeepWalk was run using several different combinations of hyperparameters then the embeddings were inspected visually. Node2Vec, however, was only run on one set of hyperparameters due to the algorithm being computationally expensive. Similarly, Node2Vec was only run on scenario 11 because of memory limitations that arise when run on scenario 10.

We analyzed the similarity between botnet nodes and normal/background nodes by calculating the pairwise euclidean and cosine distances between them. The embeddings produced by DeepWalk on scenario 10 are shown below in Figure 1. As we can see, the nodes inside the botnet are very similar to each other while being visibly different from the normal nodes.

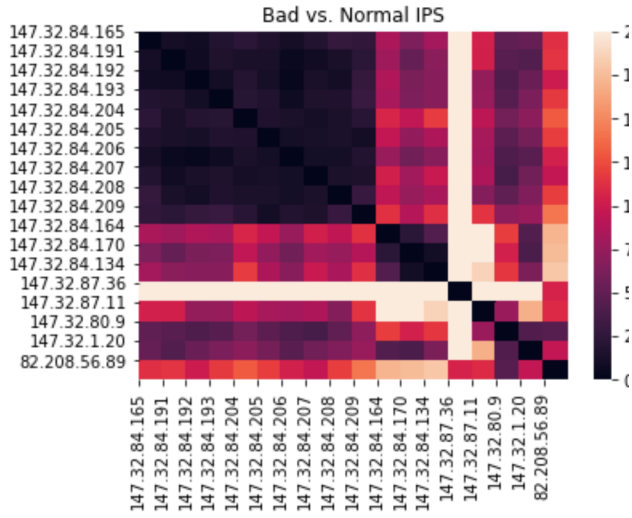
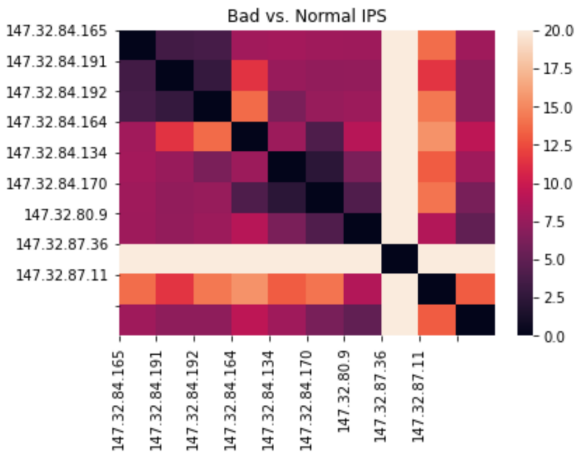
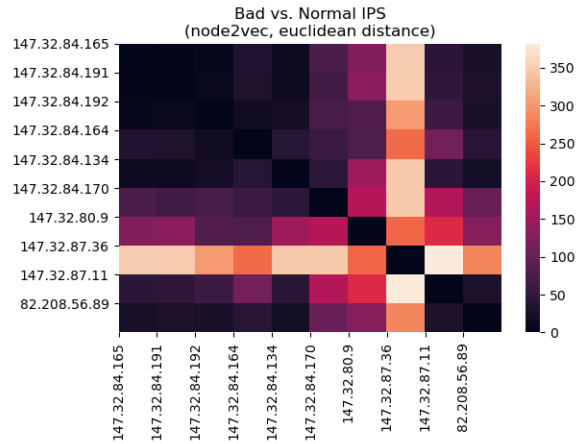



Figure 1: Pairwise euclidean distance of DeepWalk embeddings of scenario 10.




(a)




(b)

Figure 2: **(a)** Pairwise euclidean distance of DeepWalk embeddings of scenario 11. **(b)** Pairwise euclidean distance of Node2Vec embeddings of scenario 11.

We performed a similar analysis on scenario 11 using both DeepWalk and Node2Vec. DeepWalk produces results similar to those of scenario 10, where there is a visible difference between the bad nodes from the normal nodes. The embeddings produced by Node2Vec, however, while providing some separation, don't quite capture the structural differences that DeepWalk does. We were only able to try one set of hyperparameters for Node2Vec so better embeddings might be produced if we had more time to try more combinations of hyperparameters.

### 4.2 IoT-23 Dataset

Deepwalk on IoT-23 Dataset with multiple botnet doesn't give us a clear result. We use the similar techniques in IoT-23, analyzed the similarity between botnet nodes and normal/background nodes by calculating the pairwise euclidean and cosine distances between them. The embeddings produced by DeepWalk on our reduced dataset doesn't give us a clear result.

## 5 Conclusion

Using two scenarios from the CTU-13 dataset, we trained DeepWalk and Node2Vec embeddings. We compared the embeddings of nodes in the botnet

against other nodes in the network. This was done by calculating the pairwise euclidean distance between them. The DeepWalk embeddings of botnet nodes differ from other nodes significantly enought that the differences can be seen visually, suggesting that DeepWalk embeddings adequately captures the structural differences and similarities between nodes and can possibly be used to identify botnetws within networks. However, for dataset with various of malicious traffic and different types of botnet. The result might not be that clear.

Future work might include searching over a wider space of hyperparameters in order to examine if there exists a combination that produces better embeddings, especially for the more computationally expensive Node2Vec. Also, we might explore the embeddings of botnets that have different structures, such as P2P botnets.

# References


[1] Camacho-Collados, Jose and Pilehvar, Mohammad Taher. *From Word to Sense Embeddings: A Survey on Vector Representations.* arXiv. 2018. https://arxiv.org/abs/1805.04032

[2] Garcia, Sebastian and Grill, Martin and Stiborek, Jan and Zunino, Alejandro. *An empirical comparison of botnet detection methods.* Computers and Security Journal, Elsevier. 2014. Vol 45, pp 100-123. http://dx.doi.org/10.1016/j.cose.2014.05.011

[3] Grover, Aditya and Leskovec, Jure. *node2vec: Scalable Feature Learning for Networks* ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press. 2016. https://arxiv.org/abs/1607.00653

[4] Madelbaum, Amit and Shalev, Adi. *Word Embeddings and Their Use in Sentence Classification Tasks.* arXiv. 2016. https://arxiv.org/abs/1610.08229

[5] Mikolov, Tomas and Sutskever, Ilya and Chen, Kai and Corrado, Greg and Dean, Jeffrey. *Distributed Representations of Words and Phrases and Their Compositionality.* Proceedings of the 26th International Conference on Neural Information Processing Systems, Curran Associates Inc. 2013. Vol 2, pp 3111-3119. https://dl.acm.org/doi/10.5555/2999792.2999959

[6] Perkins, James and Diboun, Ilhem and Dessailly, Benoit and Lees, Jon and Orengo, Christine. *Transient Protein-Protein INteractions: Structural, Functional, and Network Properties.* Structure. 2010. Vol. 18, Iss. 10, pp 1233-1243. https://doi.org/10.1016/j.str.2010.08.007

[7] Perozzi, Bryan and Al-Rfou, Rami and Skiena, Steven. *DeepWalk: Online Learning of Social Representations.* Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press. 2014. http://dx.doi.org/10.1145/2623330.2623732

[8] Skorniakov, Kirill and Turdakov, Denis and Zhabot, Andrey. *Make Social Networks Clean Again: Graph Embedding and Stacking Classifiers for Bot Detection.* CIKM Workshops. 2018. http://ceur-ws.org/Vol-2482/paper39.pdf

[9] Yue, Xiang and Wang, Zhen and Huang, Jingong and Parthasarathy, Srinivasan and Moosavinasab, Soheil and Huang, Yungui and Lin, Simon and Zhang, Wen and Zhang, Ping and Sun, Huan. *Graph Embedding on Biomedical Networks: Methods, Applications and Evaluations.* Bioinformations. 2020. Vol. 36, Iss. 4, pp 1241-1251. https://doi.org/10.1093/bioinformatics/btz718

[10] Stratosphere Laboratory. *A labeled dataset with malicious and benign IoT network traffic.* January 22th. Agustin Parmisano, Sebastian Garcia, Maria Jose Erquiaga. https://www.stratosphereips.org/datasets-iot23