# Topology based attribute propagation for network devices

1st Milon Bhattacharya
*Dept of Computer Engineering*
*Indian Institute of Technology, Ropar*
Ropar, India
milon.20csz0017@iitrpr.ac.in

***Abstract*—Due to their use in many appliances of common use, IoT devices have become ubiquitous in our daily lives. As their reach increases, they are handling an increasingly amount of user information, which occasionally contains sensitive data. Due to their heterogeneous nature and the constraints of memory and storage, an agent based approach to security does not work for such devices. Further, the current industry practices of estimating a devicé risk by profiling does not always work as it is a static measure which pre-supposes an access to the device. Also, such measures tend to look at the device in isolation. The study attempts to develop ways and means to estimate risk of a device based on its interactions with other devices. We take a deep learning based approach using graph neural network for such an estimation. The study was able to show the efficacy of the approach where we were able to predict the device risk with an accuracy of eighty nine percent.**

***Index Terms*—IoT, GNN, Graph neural networks, network security**

## I. Introduction

With the increasing digitization of products and services it is expected that the number of embedded devices which are connected to the Internet would cross 75 billion by the year 2025 [15] [16]. This increase would lead to even larger collection of sensitive user information. Patient vitals, user's geographic location are few examples of the data points which are increasingly being collected. Unlike convention IT devices like laptops, desktops, compute and storage is on a premium for IoT devices. This leads to watering down of the security measures for such devices [17] [4]. Therefore, current approaches [18] [19] rather quantify the risk profile of such devices and restrict access based on this measure, instead of just trying to secure the device from all possible threats.

## II. Theoretical Background

### A. Risk Quantification

Models like SAFER [14] have attempted to quantify the risk of a device getting compromised using different identification mechanisms. However, most of such approaches attempt to estimate the risk in an standalone fashion including [13]. This is what is referred to as *static risk*. However, we introduce a new formulation refereed to as *dynamic risk*, which calculates the probability of a device being compromised based on all the other devices it is communicating with. The motivation comes from the fact that malware infection travels from one device to the other and is a process to process transmission. This behaviour is well documented by existing industry standards like ATT&CK [1].Therefore, the characteristics of the device (OS version, device type) contribute to a devices vulnerability while the *behaviour* of the device makes exploitation of these vulnerability possible.

### B. Calculation of device risk

Netskope's IoT security platform [3] uses a rule based engine called HyperContext $^{TM}$ which assigns a probability to each device ( on a scale of zero to one). The system looks at global (e.g. known vulnerability for a version of device OS) and local factors (e.g. open ports , anomalous behaviour). An open source description can be found here [2].

### C. Deep Learning based estimation of dynamic risk

The previous section described how a static measure of risk can be estimated. Any such measure has a couple of problems namely,

1) The measure is made at a given point in time and does not change, irrespective of changes in device properties.
2) Such a measure requires an exact determination of many features Operating system, OS version, device-type etc.
3) Finally, it does not consider the *neighbours* of the device, i.e. the other devices which the node would be communicating with.

Our work attempts to propose a new formulation called *dynamic risk* which would ameliorate the possible problems with the above formulation. Rather than having a closed form expression for calculating risk, we propose a deep learning based model, which *guesses* the value of risk based on the values of the neighbours.

The approach assumes the presence of a network with a stable and completely specified topology. When a new device is added/discovered within this network, its incremental effect on the rest of the network is minimal, however as the attributes of the new device are unknown, an immediate quantification of the static risk is difficult.

The deep learning model looks at the know attributes of the device, its neighbours (and their complete profile) and predicts a value of device risk for the new device. Use of a deep learning model assures, that the relationship between different parameters and device risk is learned, however complicated it might be. Also, the need of specifying rules as required for by conventional rule based system is done away with as the model can be incrementally trained as more data is observed and new threat and device categories are discovered.

### D. Graph Neural Networks

Computer networks, can be abstracted as a graph $G = (V, E)$ where $V$ is a set of nodes (or vertices) and $E$ refers to the set of edges (either directed/undirected) $E \subseteq \{(x, y) \mid (x, y) \in V^2 \text{ and } x \neq y\}$.

There have been multiple attempts to model the behaviour of IoT networks using neural networks. The most successful attempts have been using graph neural networks or GNNs [11]. They have been reported to as consistently outperforming other representations like MLPs and recurrent networks. [12] [10]. The reason is that GNN acknowledge the connected structure of IoT devices and therefore are in a better position to learn the underlying behaviour. This capability comes at a cost where the training has to be attempted differently from conventional neural networks. Efforts have to be made to distribute the network equally, especially for large networks which are trained on heterogeneous platforms [9] [8].

Graph representation learning using graph neural network can be summarize as follows;

For a graph $G = (V, E)$ for a node $n \in V$ we define the neighbour $N_u$ as the set of nodes which are connected by an edge such that $\forall n_a \in N_u \, \exists n, n_a \in E$. Also each node and edge are associated with feature vectors $x_u \in (R)^p$ and $y_e \in (R)^q$ respectively.

GNNs use the concept of message passing, where neighbouring nodes exchanges bits of information or messages. For a given node, all the incoming message (all having the same dimension) are aggregated (or the commonly used terminology COMBINE). Mathematically, we can represent it as following:

$$\mathbf{h}_u = \phi\left(\mathbf{x_u}, \bigoplus_{v \in N_u} \psi(\mathbf{x}_u, \mathbf{x}_v, \mathbf{e}_{uv})\right) \tag{1}$$

where $\phi$ and $\bigoplus$ are the update and aggregation function respectively [7] [6].

## III. Experiments

### A. Description of the dataset

As an illustration of the concept, the study attempted to see if this approach can be use to *predict* the currently used metric (i.e. static risk). The attempt was to use the knowledge of the communication between the different nodes in the network and predict the risk of a device (as quantified by the rule based system) using a deep learning model.

For the purpose of this study we selected a set of twelve networks. These are private networks where Netskope's custom monitoring software was deployed to monitor the network activity. The platform would parse the network traffic infer the source and destination nodes (uniquely identified using mac IDs) and the nature of communication (TLS, TCP, SSH etc.). The platform also records the attribtues of the device. These would include device type, make and model. To comply with extant regulations on data privacy and confidentiality, the customers (tenants) names have been replaced with a single letter abbreviations. These networks vary in size by two order of magnitude (in terms of the node size). They belong to different customers from domains like healthcare, manufacturing and retail.

TABLE I
NETWORKS WITH SIZE

| Sr. No | Name | Device Count | Edge Count |
|---|---|---|---|
| 1 | A | 409 | 315 |
| 2 | B | 15614 | 15000 |
| 3 | C | 459 | 443 |
| 4 | D | 1343 | 1309 |
| 5 | E | 36 | 27 |
| 6 | F | 459 | 414 |
| 7 | G | 15056 | 15000 |
| 8 | H | 10289 | 10075 |
| 9 | I | 223 | 199 |
| 10 | J | 294 | 191 |
| 11 | K | 114 | 103 |
| 12 | L | 13 | 3 |

Additionally, the platform records attributes for each of the nodes. The nodes could be any device. As described in the previous section, calculation of device risk is done using a rule base system and depends on a number of attributes. For the purpose of this study, we look at three attributes device type, device OS and device risk. The attempt is to predict the static device risk, which is calculated using a set of comprehensive rules by using the other two attributes of the device and all attributes of the device's immediate neighbours II.

TABLE II
ATTRIBUTES AND THEIR SAMPLE VALUES

| Attribute | Sample values |
|---|---|
| Device Type | computer,camera, access point |
| Device OS | macos, fireos, ios |
| Ownership | managed/un-managed |
| Device risk | 0.4,10,0 |

### B. Model Architecture

As described in previous sections we use a GNN to model the interactions between the device risk of a node and the attributes of its neighbours. To demonstrate the power of GNN we use a simple architecture, which is easy to train and has a low inference time. Below 1 is the architecture of the model used for this study.

The pipeline was written in python using the PyGeometric library [5]. It's wrapper over the PyTorch library developed for processing graph networks. As we wish to establish a baseline for the approach no data augmentation methods were used.

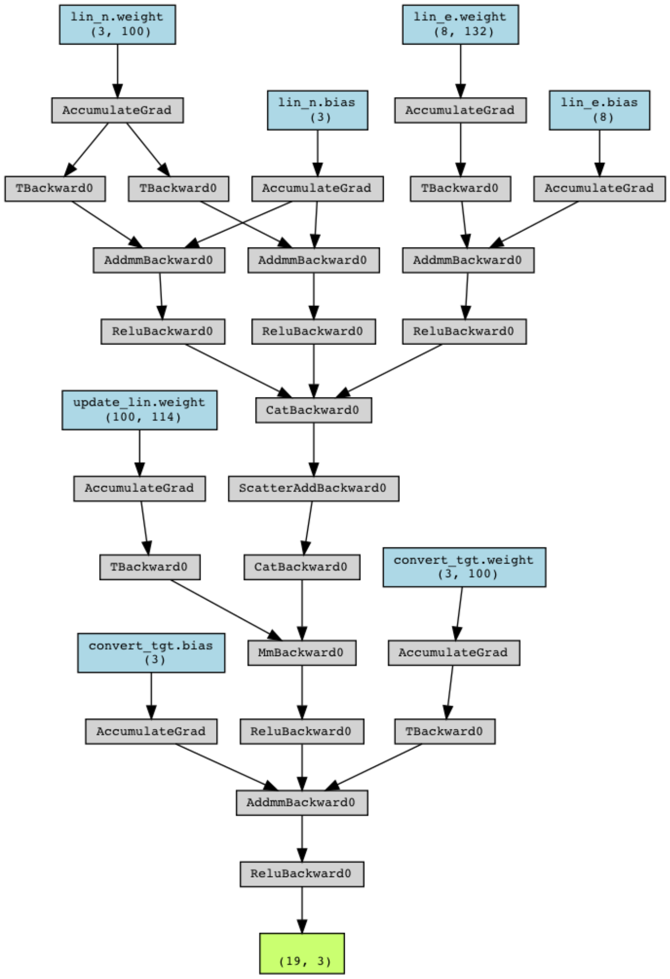

Fig. 1. The architecture of the GNN model

## IV. Results

The model was trained for 1000 epochs, in two modes, firstly when the networks were divided in two disjoint groups; one set used for training and other for testing.

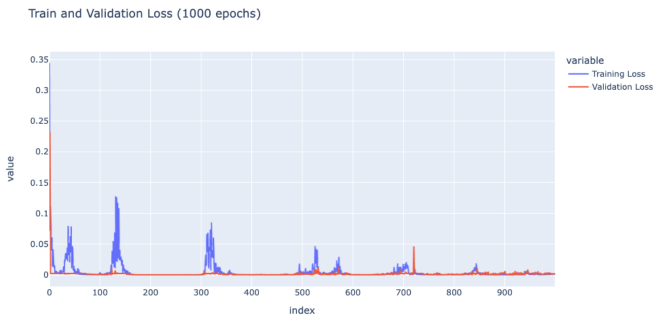

Fig. 2. Train/test error (MAE) when the graphs are split

In the second approach, we split each of the networks individually (using a mask).

The training metric were observed in both the approaches. The second approach results in lower mean absolute error (MAE) for the test set, however as the sizes of the networks differ by order of magnitudes, the accuracy is highly dependent on split.

TABLE III
Experimental results

| Split | MAE |
|---|---|
| Intra-network | 98 |
| Inter-network | 97 |

## V. Conclusion

The study proves that device attributes, especially those which are cohort based in nature can be modeled using a graph neural network. The approach is fault-tolerant as it works using the attributes of the immediate neighbours of the device. Finally, it is scalable for as the inference can be parallelized for large networks.

## References

[1] Lateral movement in MITRE ATTACK https://attack.mitre.org/tactics/TA0008/

[2] Addressing Device Security Risks in the Hybrid Enterprise with Netskope Device Intelligence https://www.netskope.com/blog/addressing-device-security-risks-in-the-hybrid-enterprise-with-netskope-iot-security

[3] Netskope's IoT Security https://www.netskope.com/products/iot-security

[4] Schiller, Eryk, et al. ”Landscape of IoT security.” Computer Science Review 44 (2022): 100467.

[5] Fey, Matthias and Lenssen, Jan E. ’Fast Graph Representation Learning with PyTorch Geometric” ICLR Workshop on Representation Learning on Graphs and Manifolds. 2019.

[6] Adjeisah, Michael, et al. ”Towards data augmentation in graph neural network: An overview and evaluation.” Computer Science Review 47 (2023): 100527.

[7] Sanchez-Lengeling, et al., ”A Gentle Introduction to Graph Neural Networks”, Distill, 2021.

[8] Wang, Qiange, et al. ”Neutronstar: distributed GNN training with hybrid dependency management.” Proceedings of the 2022 International Conference on Management of Data. 2022.

[9] Cai, Zhenkun, et al. ”DGCL: an efficient communication library for distributed GNN training.” Proceedings of the Sixteenth European Conference on Computer Systems. 2021.

[10] Zhou, Xiaokang, et al. ”Hierarchical adversarial attacks against graph-neural-network-based IoT network intrusion detection system.” IEEE Internet of Things Journal 9.12 (2021): 9310-9319.

[11] Dong, Guimin, et al. ”Graph neural networks in IoT: A survey.” ACM Transactions on Sensor Networks (TOSN) (2022).

[12] W. Zhang et al., ”Modeling IoT Equipment With Graph Neural Networks,” in IEEE Access, vol. 7, pp. 32754-32764, 2019, doi: 10.1109/ACCESS.2019.2902865.

[13] Akella, Srinivas, and Shahab Sheikh-Bahaei. ”Assessing computer network risk.” U.S. Patent No. 11,349,863. 31 May 2022.

[14] Oser, Pascal, et al. ”Risk prediction of IoT devices based on vulnerability analysis.” ACM Transactions on Privacy and Security 25.2 (2022): 1-36.

[15] A. Koohang, C. S. Sargent, J. H. Nord, and J. Paliszkiewicz, “Internet of things (IOT): From Awareness to continued use,” International Journal of Information Management, vol. 62, p. 102442, 2022.

[16] E. Schiller, A. Aidoo, J. Fuhrer, J. Stahl, M. Ziörjen, and B. Stiller, “Landscape of IOT security,” Computer Science Review, vol. 44, p. 100467, 2022.

[17] A. Cirne, P. R. Sousa, J. S. Resende, and L. Antunes, “IOT security certifications: Challenges and potential approaches,” Computers ; Security, vol. 116, p. 102669, 2022.

[18] R. Nath N and H. V Nath, “Critical analysis of the layered and systematic approaches for understanding IOT security threats and challenges,” Computers and Electrical Engineering, vol. 100, p. 107997, 2022.

[19] Y. Sunil Raj, S. Albert Rabara and S. Britto Ramesh Kumar, ”A Security Architecture for Cloud Data Using Hybrid Security Scheme,” 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2022, pp. 1766-1774, doi: 10.1109/ICSSIT53264.2022.9716379.