Network Intrusion Detection using GNNs

- Yash Malik (2001CS79)

1.

Previous Work

XAI based approaches -

- Local Interpretable Model Agnostic Explanation (LIME)
- SHAP Values

"Why Should I Trust You?" Explaining the Predictions of Any Classifier

Marco Tulio Ribeiro
University of Washington
Seattle, WA 98105, USA
marcotcr@cs.uw.edu

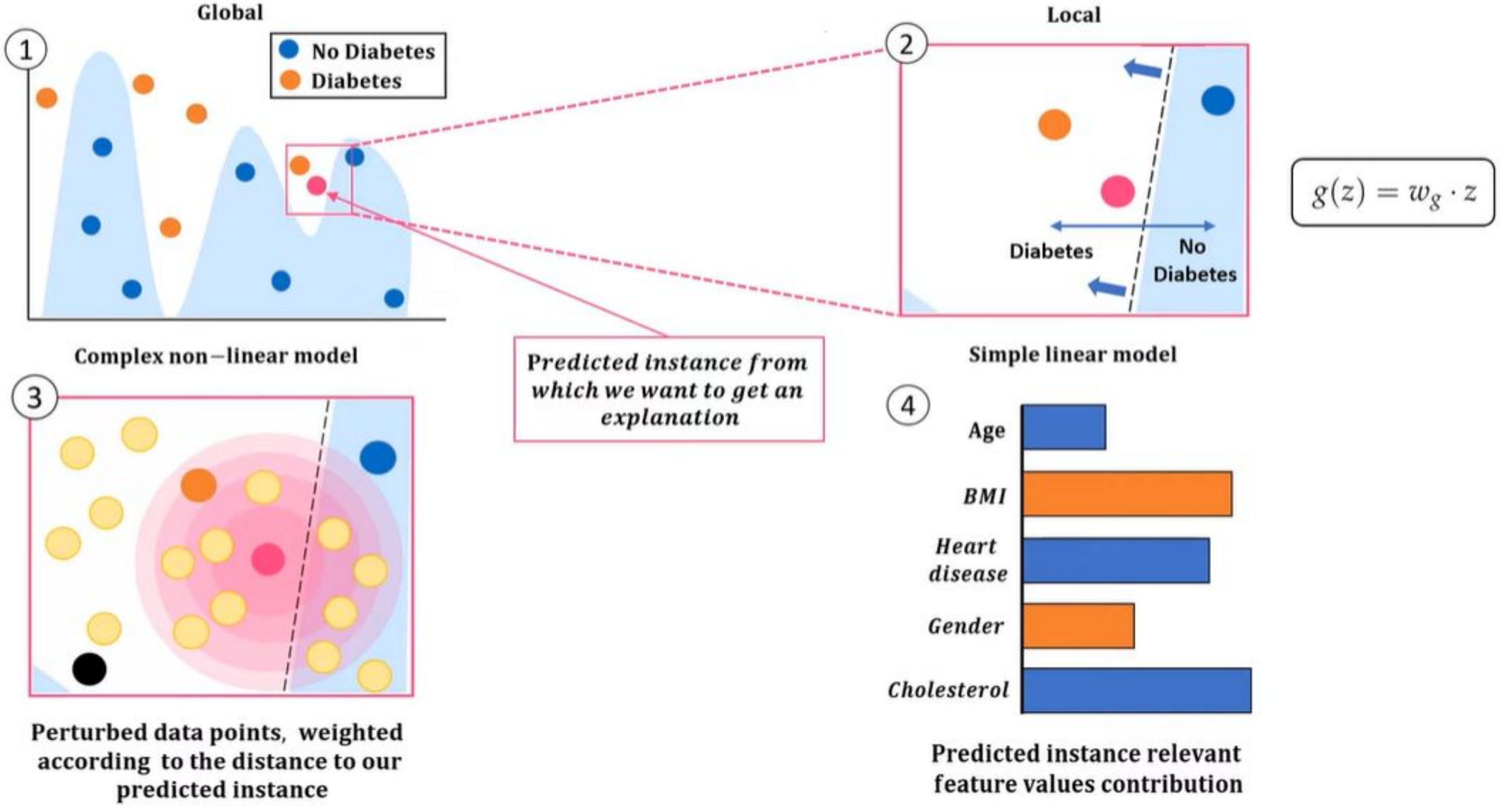
Sameer Singh
University of Washington
Seattle, WA 98105, USA
sameer@cs.uw.edu

Carlos Guestrin
University of Washington
Seattle, WA 98105, USA
guestrin@cs.uw.edu

(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 4: Explaining an image classification prediction made by Google's Inception neural network. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$), "Acoustic guitar" ($p = 0.24$) and "Labrador" ($p = 0.21$)

LIME step by step



Calculating shapley values

Blackbox model

Input datapoint

Age

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \underbrace{\frac{|z'|!(M - |z'| - 1)!}{M!}}_{\text{Weighting}} \underbrace{[f_x(z') - f_x(z' \setminus i)]}_{\text{Contribution}}$$

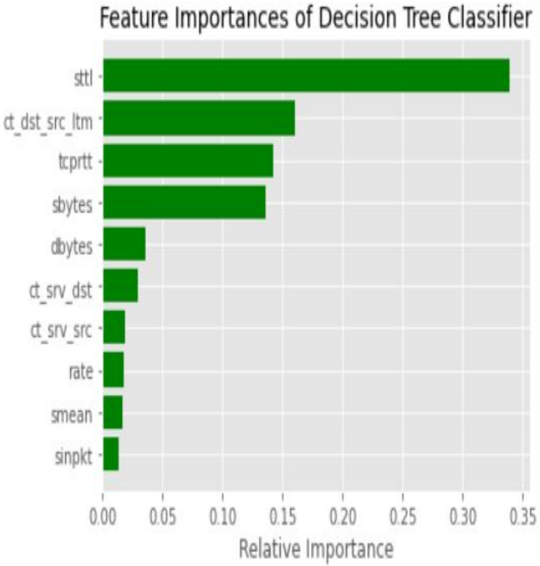
Shapley value for feature i

Subset

Simplified data input

Previous Model Results

Decision Tree

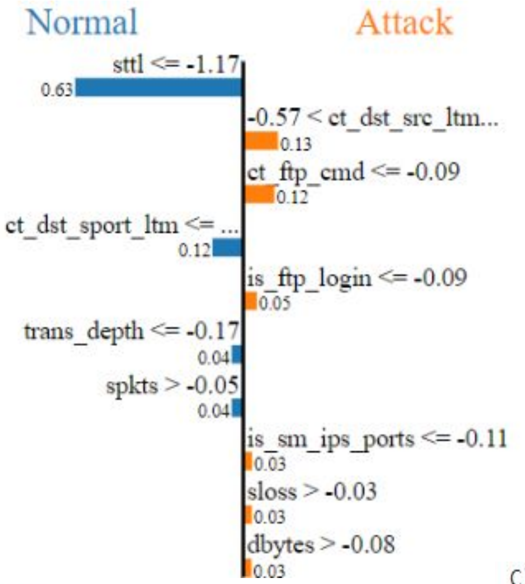


Accuracy: 0.8510901614568184

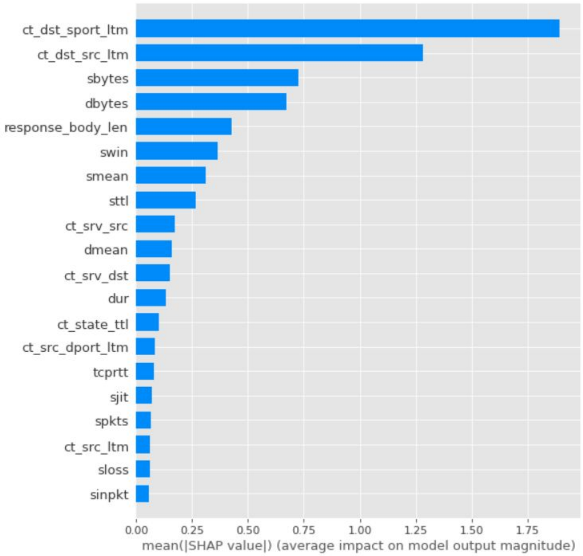
Reporting for ['Decision Tree Classifer', 'RegLog']:

	precision	recall	f1-score	support
0	0.69	0.98	0.81	56000
1	0.99	0.79	0.88	119341
accuracy			0.85	175341
macro avg	0.84	0.88	0.84	175341
weighted avg	0.89	0.85	0.86	175341

MLP



XGBoost



Classification Report:

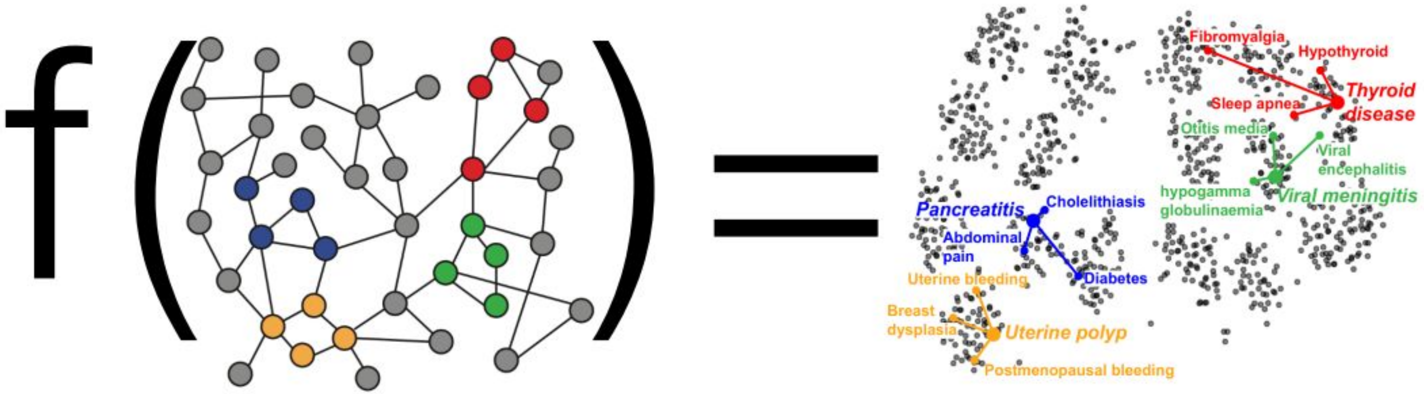
	precision	recall	f1-score	support
0	0.74	0.92	0.82	56000
1	0.96	0.85	0.90	119341
accuracy			0.87	175341
macro avg	0.85	0.89	0.86	175341
weighted avg	0.89	0.87	0.88	175341

2.

E-GraphSAGE

Node Embeddings

- **Intuition:** Map nodes to d-dimensional embeddings such that similar nodes in the graph are embedded close together

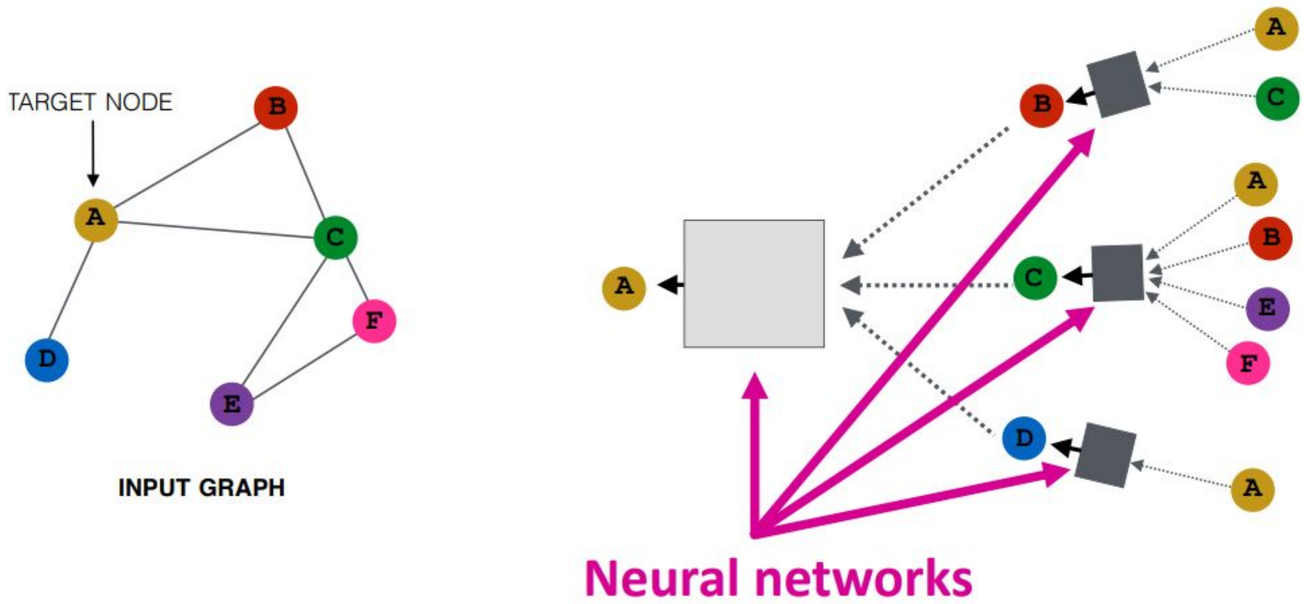


Input graph 2D node embeddings

How to learn mapping function f?

Idea: Aggregate Neighbours

- **Intuition:** Nodes aggregate information from their neighbors using neural networks

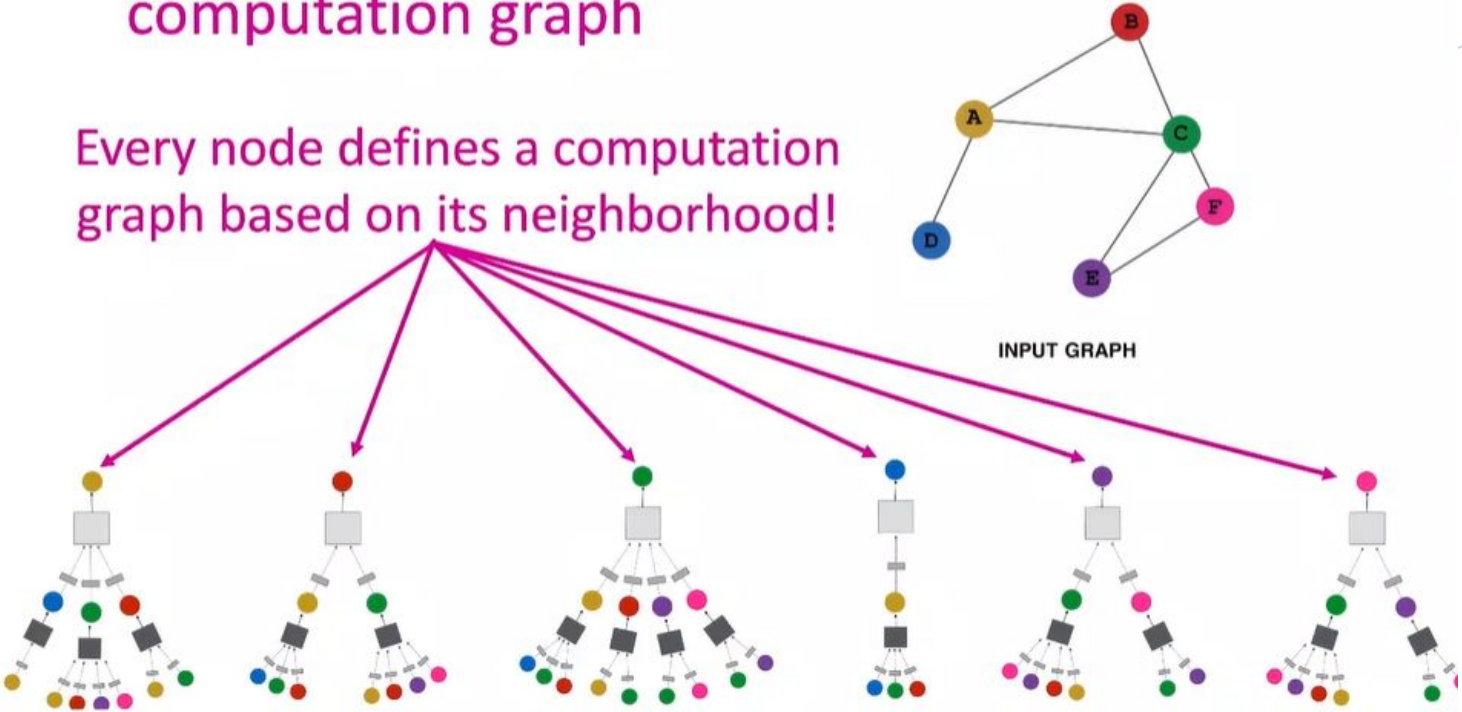


12/6/18 Jure Leskovec, Stanford CS224W: Machine Learning with Graphs, http://cs224w.stanford.edu 23

Idea: Aggregate Neighbours

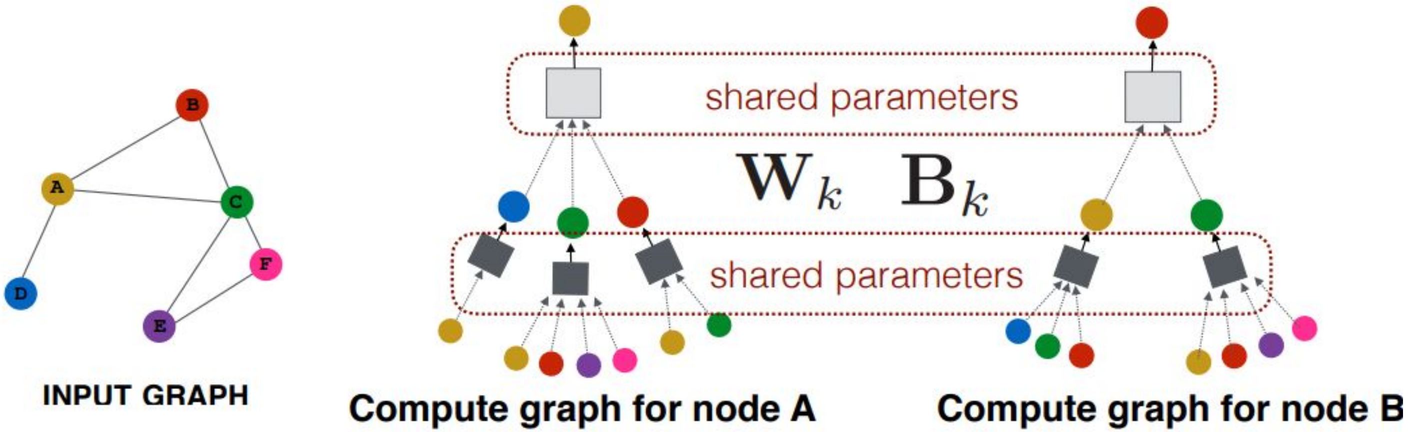
- **Intuition:** Network neighborhood defines a computation graph

Every node defines a computation graph based on its neighborhood!

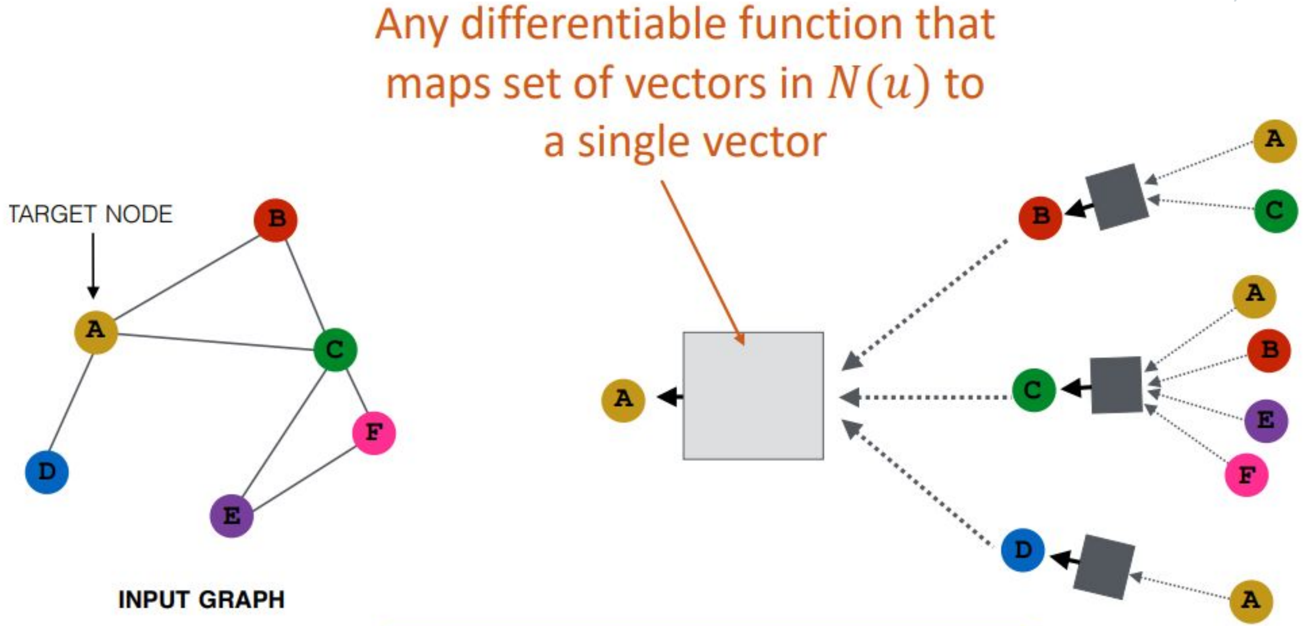


Inductive Capability

- The same aggregation parameters are shared for all nodes:
  - The number of model parameters is sublinear in $|V|$ and we can **generalize to unseen nodes**!



GraphSAGE Idea



$$\mathbf{h}_v^k = \sigma\left(\left[\mathbf{A}_k \cdot \boxed{\mathrm{AGG}(\{\mathbf{h}_u^{k-1}, \forall u \in N(v)\})}, \mathbf{B}_k \mathbf{h}_v^{k-1}\right]\right)$$

Apply L2 normalization for each node embedding at every layer

Neighbourhood Aggregation

- **Simple neighborhood aggregation:**

$$\mathbf{h}_v^k = \sigma\left(\mathbf{W}_k \sum_{u \in N(v)} \frac{\mathbf{h}_u^{k-1}}{|N(v)|} + \mathbf{B}_k \mathbf{h}_v^{k-1}\right)$$

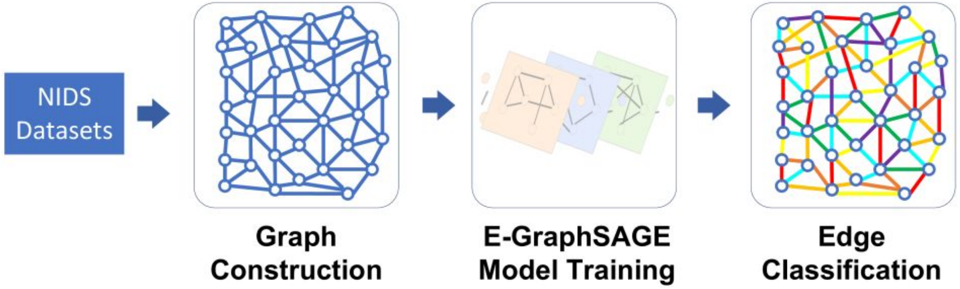
- **GraphSAGE:**

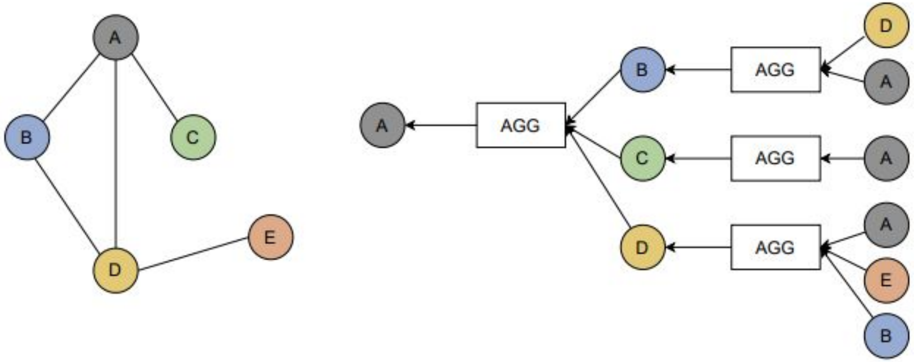
Concatenate neighbor embedding and self embedding

$$\mathbf{h}_v^k = \sigma\left(\left[\mathbf{W}_k \cdot \boxed{\text{AGG}\left(\{\mathbf{h}_u^{k-1}, \forall u \in N(v)\}\right)}, \mathbf{B}_k \mathbf{h}_v^{k-1}\right]\right)$$

Generalized aggregation

E-GraphSAGE





Algorithm 1: E-GraphSAGE edge embedding

input : Graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$;
input edge features $\{\mathbf{e}_{uv}, \forall uv \in \mathcal{E}\}$;
input node features $\mathbf{x}_v = \{1, \ldots, 1\}$;
depth K;
weight matrices $\mathbf{W}^k, \forall k \in \{1, \ldots, K\}$;
non-linearity σ;
differentiable aggregator functions AGG_k ;

output: Edge embeddings $\mathbf{z}_{uv}, \forall uv \in \mathcal{E}$

```
h_v^0 ← x_v, ∀v ∈ V
for k ← 1 to K do
    for v ∈ V do
        h^k_N(v) ← AGG_k({e^{k-1}_uv, ∀u ∈ N(v), uv ∈ E})
        h^k_v ← σ(W^k · CONCAT(h^{k-1}_v, h^k_N(v)))
z_v = h^K_v
for uv ∈ E do
    z^K_uv ← CONCAT(z^K_u, z^K_v)
```

$$\mathbf{h}_v^0 \leftarrow \mathbf{x}_v, \forall v \in \mathcal{V}$$
$$\mathbf{h}^k_{\mathcal{N}(v)} \leftarrow \mathrm{AGG}_k\left(\left\{\mathbf{e}^{k-1}_{uv}, \forall u \in \mathcal{N}(v), uv \in \mathcal{E}\right\}\right)$$
$$\mathbf{h}^k_v \leftarrow \sigma\left(\mathbf{W}^k \cdot \mathrm{CONCAT}\left(\mathbf{h}^{k-1}_v, \mathbf{h}^k_{\mathcal{N}(v)}\right)\right)$$
$$\mathbf{z}_v = \mathbf{h}^K_v$$
$$\mathbf{z}^K_{uv} \leftarrow CONCAT(\mathbf{z}^K_u, \mathbf{z}^K_v)$$

Model Parameters

- **Network Graph Construction**
  - BoT-IoT dataset with 6 types of attacks
  - <Source_IP || Source_Port, Dest_IP || Dest_Port>
- **E-GraphSAGE Training**
  - 2 Layered NN i.e. 2-hop neighbourhood
  - AGG function used is element-wise Mean

$$\mathbf{h}^k_{\mathcal{N}(v)} = \sum_{\substack{u \in \mathcal{N}(v), \\ uv \in \mathcal{E}}} \frac{\mathbf{e}^{k-1}_{uv}}{|N(v)|_e}$$

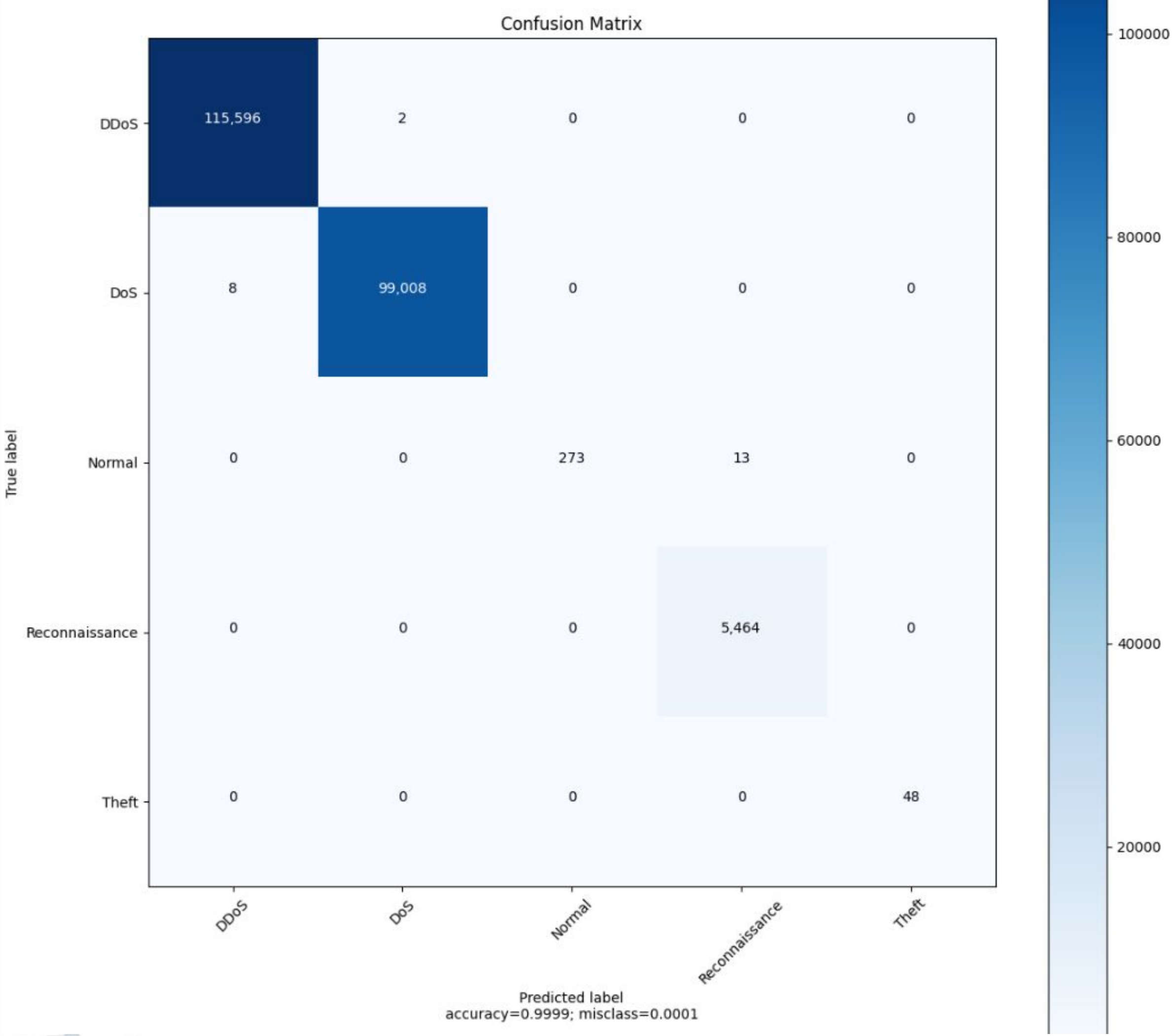
- **Edge Classification**
  - Test samples passed to get edge embeddings
  - Converted to class probabilities in final softmax layer
  - Tested against true labels

3.

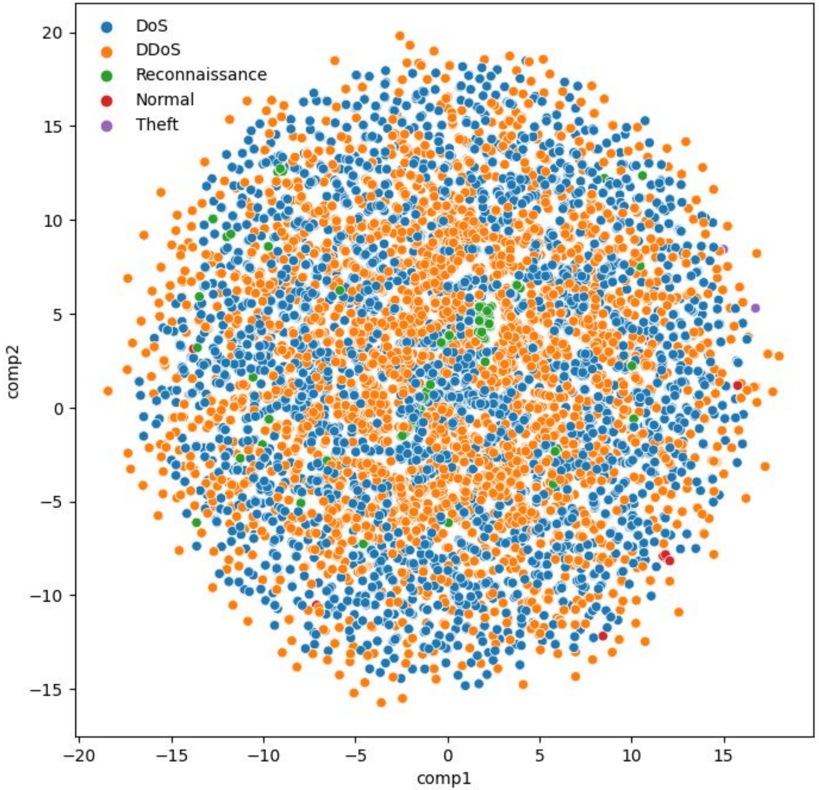
Results

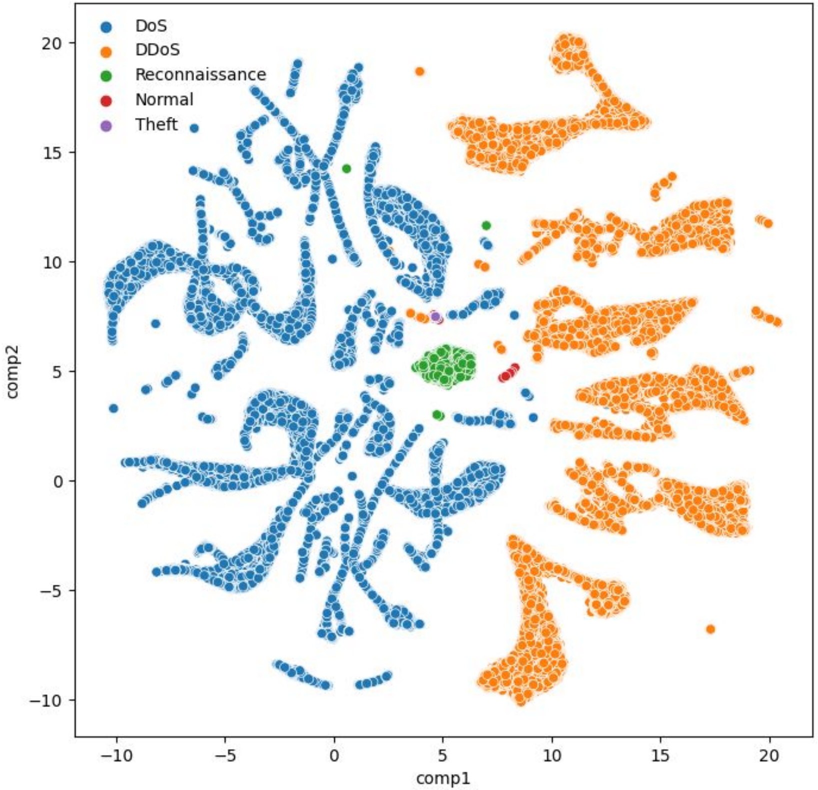
Classification Report on Test Set

	precision	recall	f1-score	support
DDoS	0.9999	1.0000	1.0000	115598
DoS	1.0000	0.9999	0.9999	99016
Normal	1.0000	0.9545	0.9767	286
Reconnaissance	0.9976	1.0000	0.9988	5464
Theft	1.0000	1.0000	1.0000	48
accuracy			0.9999	220412
macro avg	0.9995	0.9909	0.9951	220412
weighted avg	0.9999	0.9999	0.9999	220412


Confusion Matrix
True label
DDoS
DoS
Normal
Reconnaissance
Theft
115,596
2
0
0
0
8
99,008
0
0
0
0
0
273
13
0
0
0
0
5,464
0
0
0
0
0
48
100000
80000
60000
40000
20000
Predicted label
accuracy=0.9999; misclass=0.0001

Dimension Reduction using UMAP





4.

Future Work

Conclusions and Future Scope

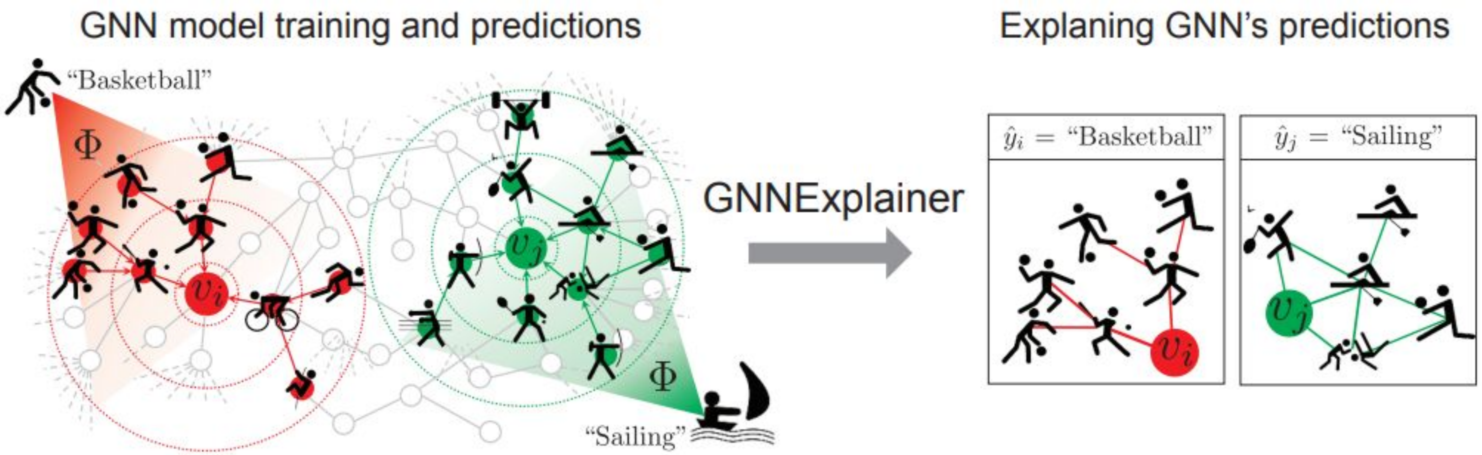
- Experimental evaluation based on Bot-IoT NIDS benchmark datasets shows that E-GraphSAGE-based NIDS performs **exceptionally well** and overall **outperforms** the state-of-theart ML-based classifiers.
- More work could be done to apply neighbourhood sampling techniques to **improve the run-time** of the E-GraphSAGE model, particularly exploring **non-uniform** sampling techniques.
- But How to combine **XAI** (previous work) based and **GNN** (current work) based approaches?

My Implementation
https://colab.research.google.com/drive/1kbc6hofWrtZj0-Ms9LO--wnNegScSYD1?usp=sharing

GNNExplainer: Generating Explanations for Graph Neural Networks

Rex Ying[†] Dylan Bourgeois[†,‡] Jiaxuan You[†] Marinka Zitnik[†] Jure Leskovec[†]

[†]Department of Computer Science, Stanford University
[‡]Robust.AI
{rexying, dtsbourg, jiaxuan, marinka, jure}@cs.stanford.edu



Thank You!