PatchGNN - A Graph-based Toolkit to Identify Security Patches

Demo

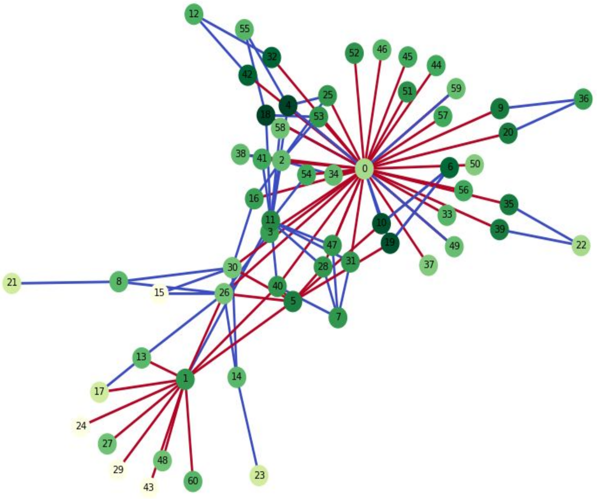
George Mason University
01/06/2022

PatchGNN Overview

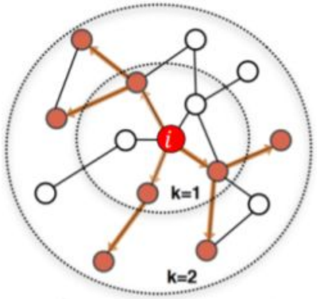
Problem: vendors may secretly release security patches.

Our work: identify security patches with graph neural networks.

- **Input:** PatchCPGs constructed from software patches.
- **Output:** if the given patch is security-related.



A PatchCPG sample from the patch
torvalds.linux.fd6040ed57d8f200ab0cc2abf706c54995a48370



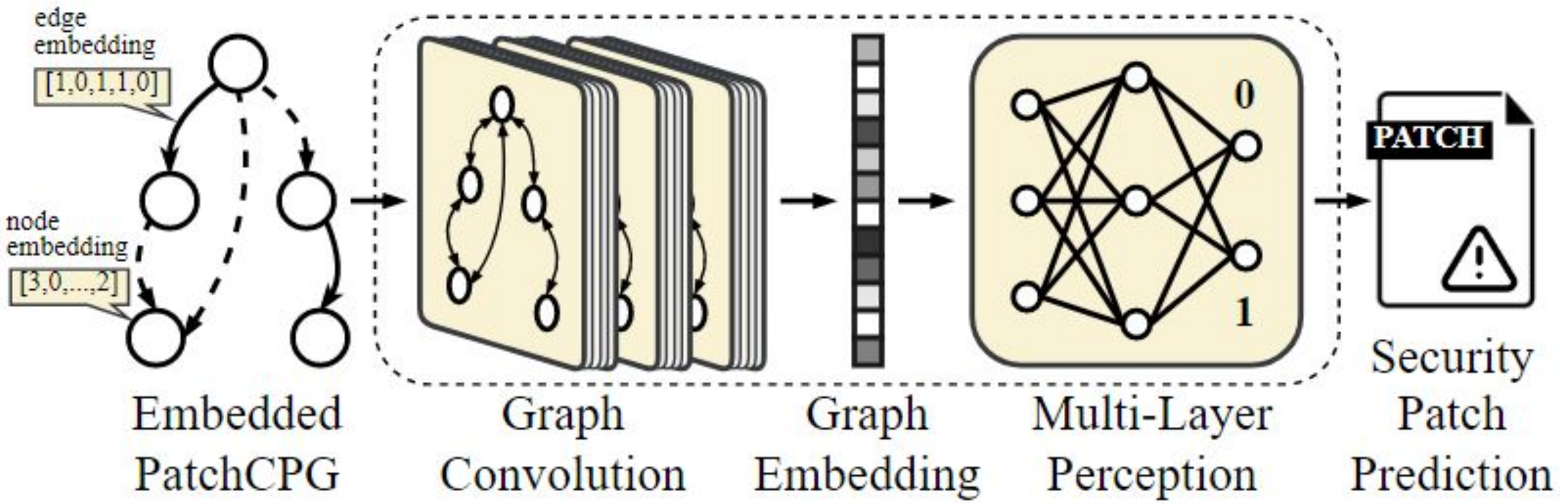
PatchGNN

Prediction

PatchGNN Architecture

PatchGNN achieves identification via 3 steps:

- **Embedding:** convert PatchCPGs into numpy format.
  - Node Embedding: 20-dimensional numeric features.
  - Edge Embedding: 5-dimensional binary vectors.
- **Graph Convolution & Pooling:** obtain graph embedding.
- **Multi-Layer Perceptron:** obtain the final prediction.



PatchGNN Training and Inference

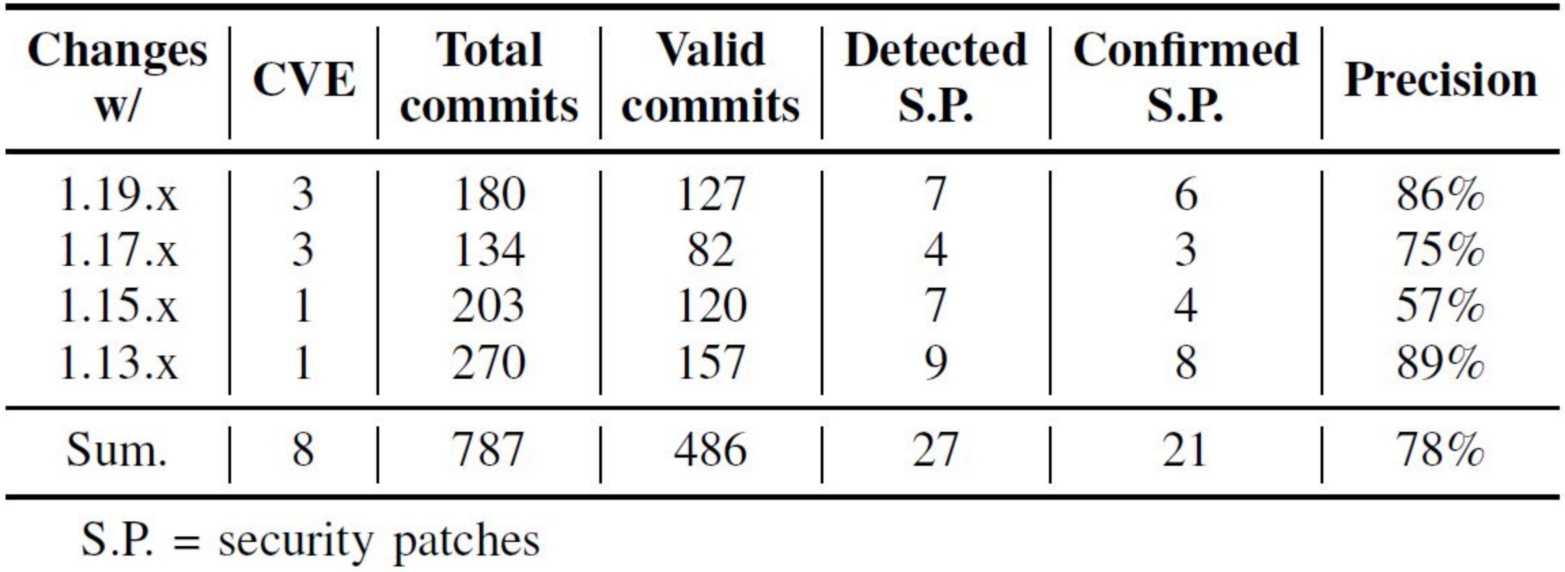
1. Training phase

- Training dataset: PatchDB (38K)
  - https://sunlab-gmu.github.io/PatchDB/
- Adam optimizer and cross-entropy loss function.
- Yield a Graph Neural Network (GNN) model.

2. Inference phase

- Given a patch, our PatchGNN model tells if it is security-related.
- Our demo: NGINX

Case Study - NGINX



Changes w/	CVE	Total commits	Valid commits	Detected S.P.	Confirmed S.P.	Precision
1.19.x	3	180	127	7	6	86%
1.17.x	3	134	82	4	3	75%
1.15.x	1	203	120	7	4	57%
1.13.x	1	270	157	9	8	89%
Sum.	8	787	486	27	21	78%

S.P. = security patches

- We detect 27 security patches from 486 commits.
- 21 out of 27 patches are real security-related after confirmation.
- True Positive Rate (Precision) = 78%

Prerequisites

OS: Linux (recommended), Windows, MacOS

Python: Python 3.6 or higher

Python modules:

- numpy
- pandas
- shutil
- clang 6.0.0.2
- pytorch (cpu) 1.10.0
- pytoch-geometric (cpu)

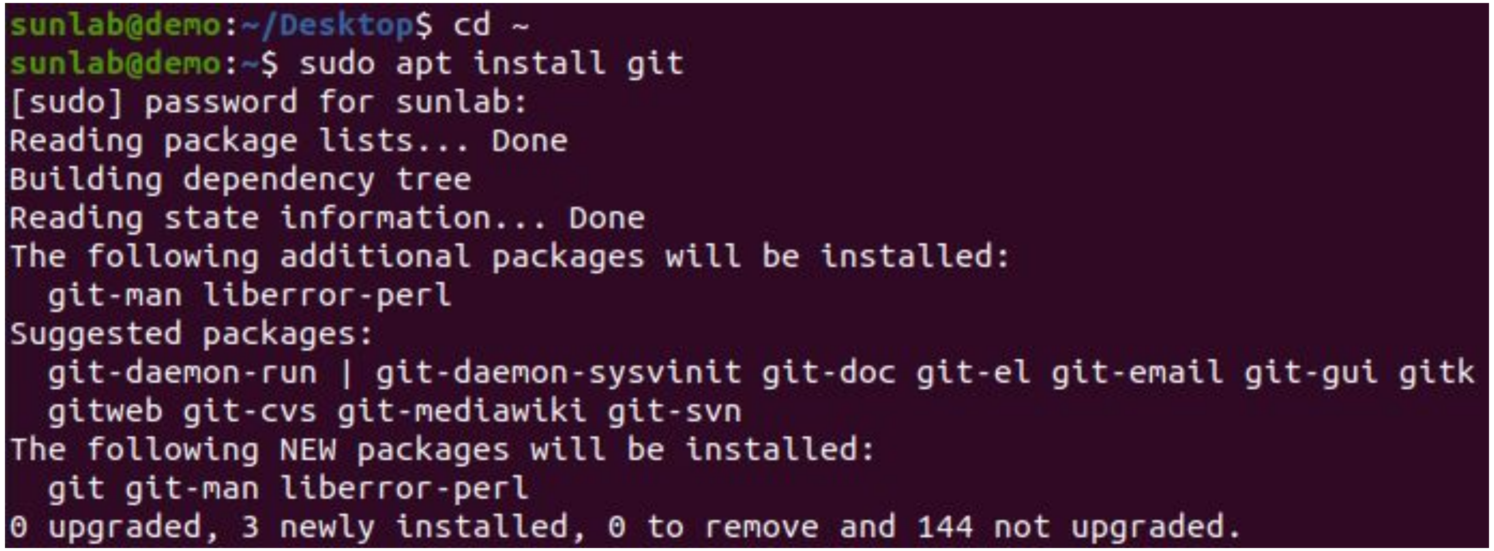
Demo

1. System installation

VirtualBox + Ubuntu20.04

>=2GB RAM, >=25GB Disk, Minimal installation

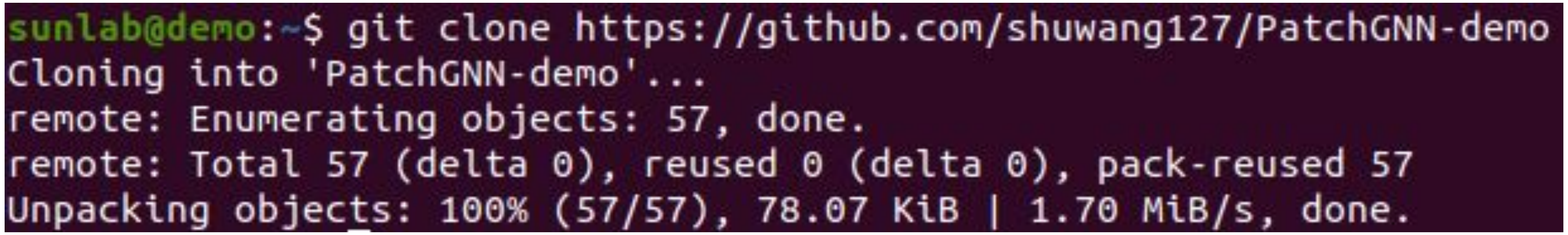
2. Install git



```
sunlab@demo:~/Desktop$ cd ~
sunlab@demo:~$ sudo apt install git
[sudo] password for sunlab:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl
0 upgraded, 3 newly installed, 0 to remove and 144 not upgraded.
```

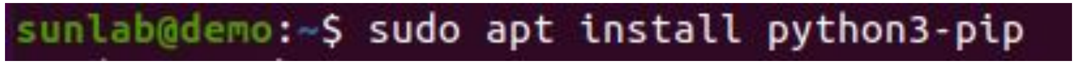
Demo (Continue)

3. Download demo source code



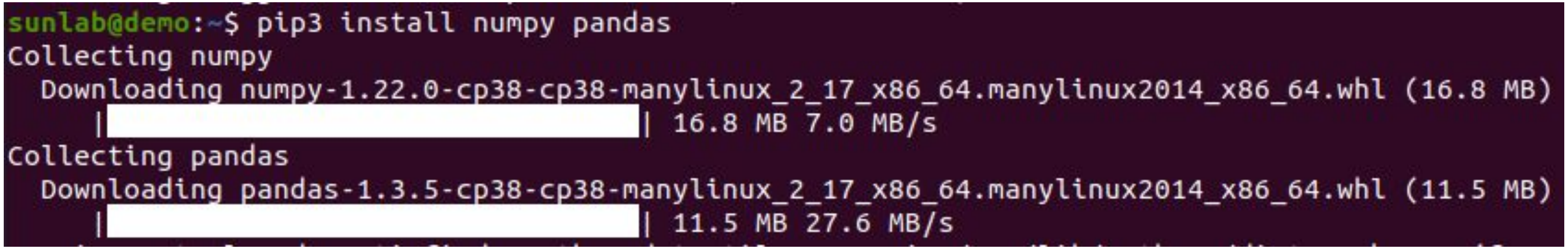
```
sunlab@demo:~$ git clone https://github.com/shuwang127/PatchGNN-demo
Cloning into 'PatchGNN-demo'...
remote: Enumerating objects: 57, done.
remote: Total 57 (delta 0), reused 0 (delta 0), pack-reused 57
Unpacking objects: 100% (57/57), 78.07 KiB | 1.70 MiB/s, done.
```

4. Install python3-pip



```
sunlab@demo:~$ sudo apt install python3-pip
```

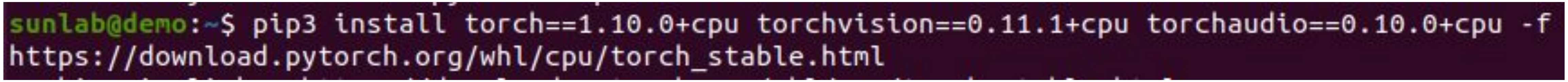
5. Install python modules (numpy, pandas)



```
sunlab@demo:~$ pip3 install numpy pandas
Collecting numpy
  Downloading numpy-1.22.0-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (16.8 MB)
     |████████████████████████████████| 16.8 MB 7.0 MB/s
Collecting pandas
  Downloading pandas-1.3.5-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.5 MB)
     |████████████████████████████████| 11.5 MB 27.6 MB/s
```

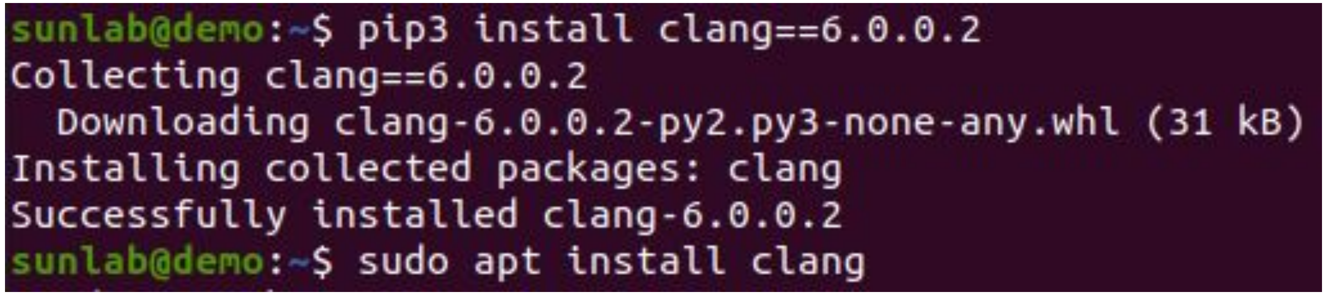
Demo (Continue)

6. Install PyTorch

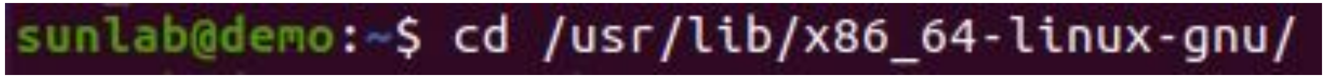


```
sunlab@demo:~$ pip3 install torch==1.10.0+cpu torchvision==0.11.1+cpu torchaudio==0.10.0+cpu -f
https://download.pytorch.org/whl/cpu/torch_stable.html
```

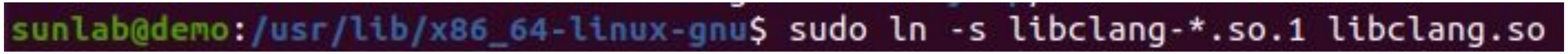
7. Install and configure clang



```
sunlab@demo:~$ pip3 install clang==6.0.0.2
Collecting clang==6.0.0.2
  Downloading clang-6.0.0.2-py2.py3-none-any.whl (31 kB)
Installing collected packages: clang
Successfully installed clang-6.0.0.2
sunlab@demo:~$ sudo apt install clang
```

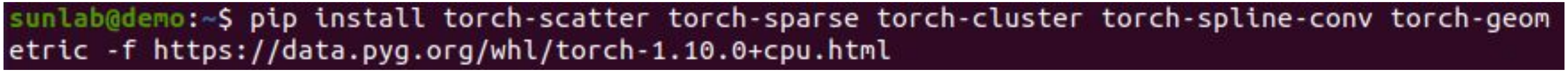


```
sunlab@demo:~$ cd /usr/lib/x86_64-linux-gnu/
```



```
sunlab@demo:/usr/lib/x86_64-linux-gnu$ sudo ln -s libclang-*.so.1 libclang.so
```

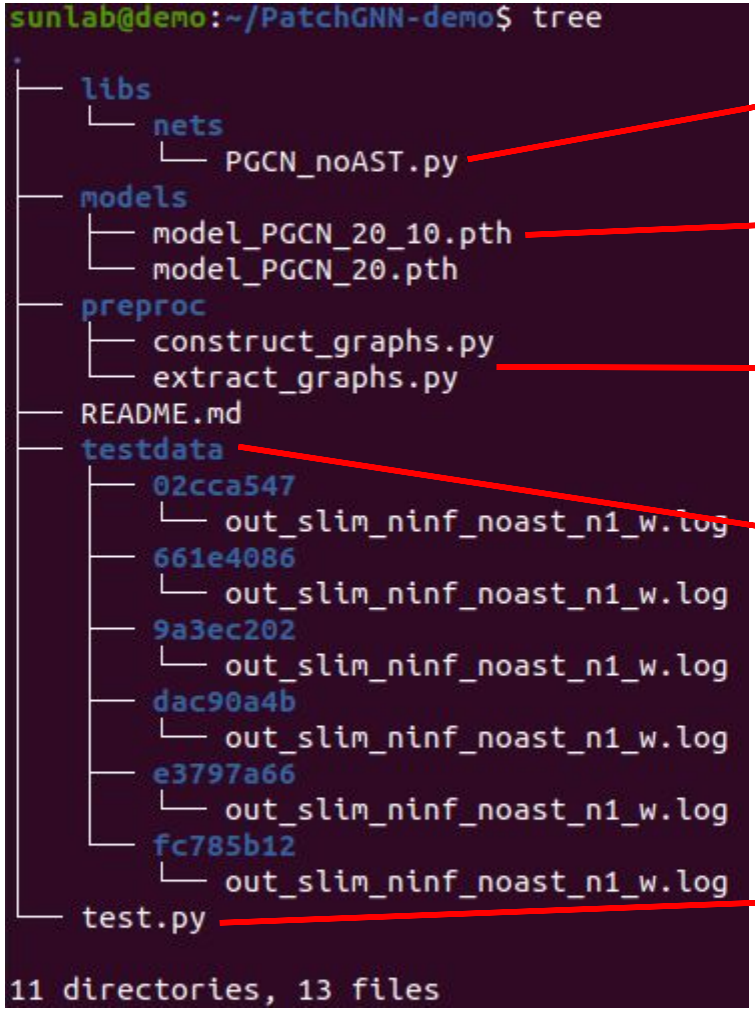
8. Install PyTorch-Geometric



```
sunlab@demo:~$ pip install torch-scatter torch-sparse torch-cluster torch-spline-conv torch-geom
etric -f https://data.pyg.org/whl/torch-1.10.0+cpu.html
```

Demo (Continue)

9. Demo folder (~/PatchGNN-demo/)



```
sunlab@demo:~/PatchGNN-demo$ tree
.
├── libs
│   └── nets
│       └── PGCN_noAST.py
├── models
│   ├── model_PGCN_20_10.pth
│   └── model_PGCN_20.pth
├── preproc
│   ├── construct_graphs.py
│   └── extract_graphs.py
├── README.md
├── testdata
│   ├── 02cca547
│   │   └── out_slim_ninf_noast_n1_w.log
│   ├── 661e4086
│   │   └── out_slim_ninf_noast_n1_w.log
│   ├── 9a3ec202
│   │   └── out_slim_ninf_noast_n1_w.log
│   ├── dac90a4b
│   │   └── out_slim_ninf_noast_n1_w.log
│   ├── e3797a66
│   │   └── out_slim_ninf_noast_n1_w.log
│   └── fc785b12
│       └── out_slim_ninf_noast_n1_w.log
└── test.py

11 directories, 13 files
```

graph network structure definition

model parameters (already trained)

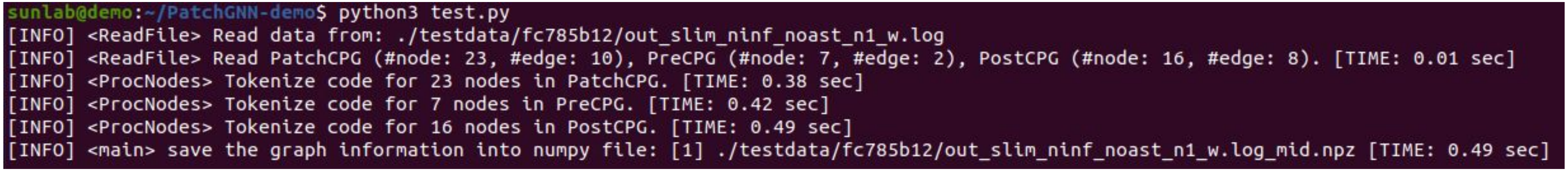
pre-processing functions

input data folder

test main entrance

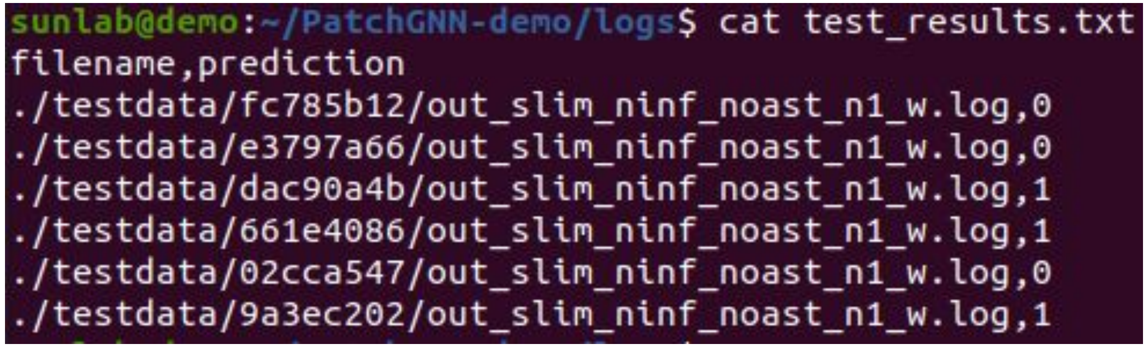
Demo (Continue)

10. Run the test program



```
sunlab@demo:~/PatchGNN-demo$ python3 test.py
[INFO] <ReadFile> Read data from: ./testdata/fc785b12/out_slim_ninf_noast_n1_w.log
[INFO] <ReadFile> Read PatchCPG (#node: 23, #edge: 10), PreCPG (#node: 7, #edge: 2), PostCPG (#node: 16, #edge: 8). [TIME: 0.01 sec]
[INFO] <ProcNodes> Tokenize code for 23 nodes in PatchCPG. [TIME: 0.38 sec]
[INFO] <ProcNodes> Tokenize code for 7 nodes in PreCPG. [TIME: 0.42 sec]
[INFO] <ProcNodes> Tokenize code for 16 nodes in PostCPG. [TIME: 0.49 sec]
[INFO] <main> save the graph information into numpy file: [1] ./testdata/fc785b12/out_slim_ninf_noast_n1_w.log_mid.npz [TIME: 0.49 sec]
```

11. Find the prediction results



```
sunlab@demo:~/PatchGNN-demo/logs$ cat test_results.txt
filename,prediction
./testdata/fc785b12/out_slim_ninf_noast_n1_w.log,0
./testdata/e3797a66/out_slim_ninf_noast_n1_w.log,0
./testdata/dac90a4b/out_slim_ninf_noast_n1_w.log,1
./testdata/661e4086/out_slim_ninf_noast_n1_w.log,1
./testdata/02cca547/out_slim_ninf_noast_n1_w.log,0
./testdata/9a3ec202/out_slim_ninf_noast_n1_w.log,1
```

Thank you & Questions?

- Dr. Kun Sun
- Email: ksun3@gmu.edu
- Homepage: https://csis.gmu.edu/ksun/