Progess of the Project

Tsung-Min Pai

2023/8/23

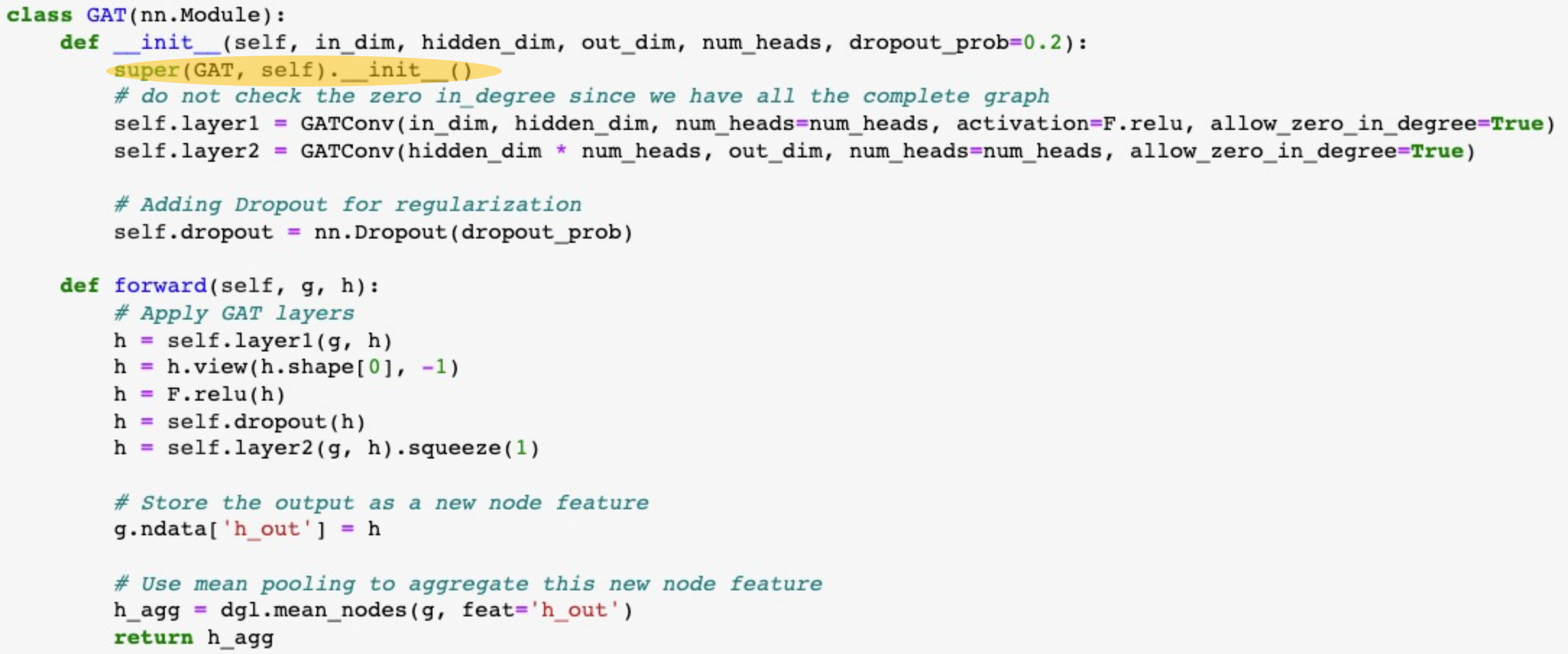
Outline

- **GAT**
- **Future Work**

Graph Attention Network - GAT

Graph Attention Network - GAT

Model:



```
class GAT(nn.Module):
    def __init__(self, in_dim, hidden_dim, out_dim, num_heads, dropout_prob=0.2):
        super(GAT, self).__init__()
        # do not check the zero in_degree since we have all the complete graph
        self.layer1 = GATConv(in_dim, hidden_dim, num_heads=num_heads, activation=F.relu, allow_zero_in_degree=True)
        self.layer2 = GATConv(hidden_dim * num_heads, out_dim, num_heads=num_heads, allow_zero_in_degree=True)

        # Adding Dropout for regularization
        self.dropout = nn.Dropout(dropout_prob)

    def forward(self, g, h):
        # Apply GAT layers
        h = self.layer1(g, h)
        h = h.view(h.shape[0], -1)
        h = F.relu(h)
        h = self.dropout(h)
        h = self.layer2(g, h).squeeze(1)

        # Store the output as a new node feature
        g.ndata['h_out'] = h

        # Use mean pooling to aggregate this new node feature
        h_agg = dgl.mean_nodes(g, feat='h_out')
        return h_agg
```

- Use the **new** verison of the dataset

Observation

- 69APs + 20benign x 10000 times in training
  - Larger learning rate
    - Since epoch 0(to epoch 157): validation accuracy = 0.6957
  - Smaller learning rate with scheduler
    - Since epoch 12 (to epoch 152): validation accuracy = 0.6957
- 165APs + 35 benign x 10000 times in training
  - Always guess benign

Future Work

Future Work

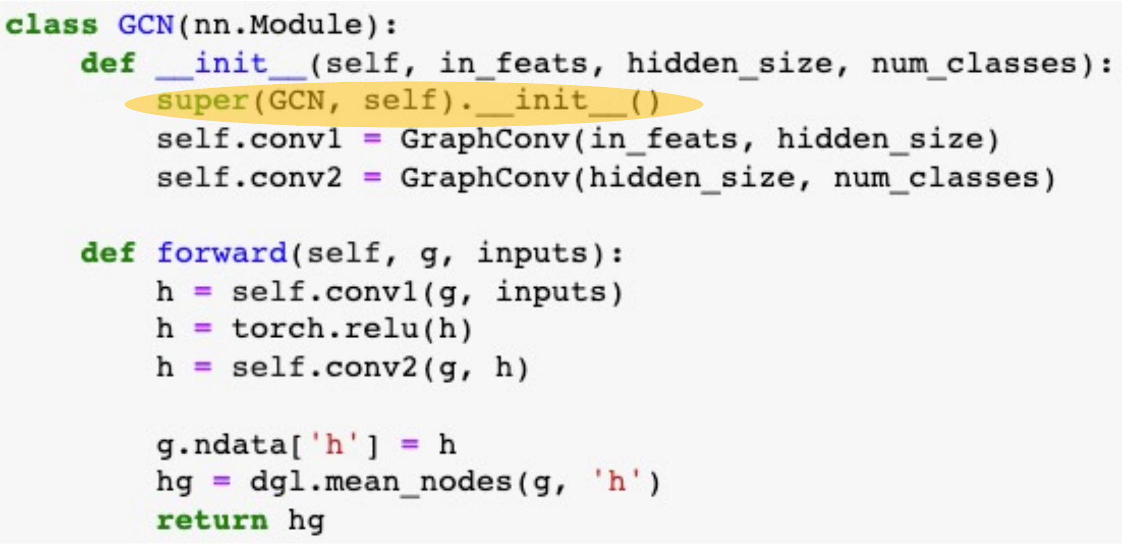
- **GNN**
  - **Use the new dataset(1 label matching more graph)**
  - Figure out the reason causing the currently bad performance on both GAT
  - Read some paper about these GNN models in classification
  - Try the **GCN**

Thanks!!

Graph Convolutional Network - GCN

Graph Convolutional Network - GCN

Model:

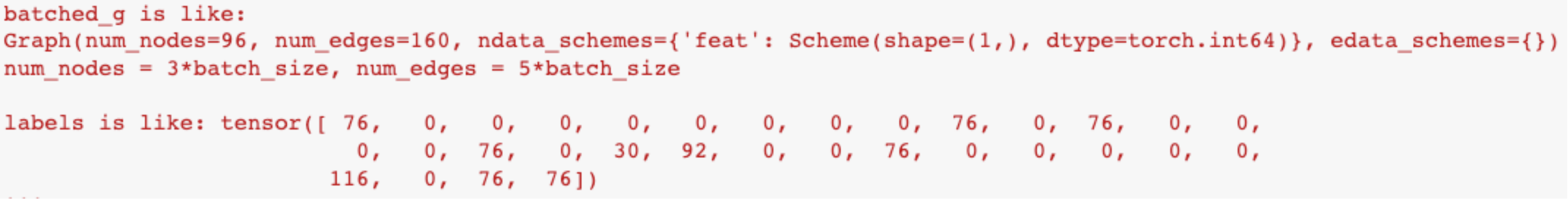


```
class GCN(nn.Module):
    def __init__(self, in_feats, hidden_size, num_classes):
        super(GCN, self).__init__()
        self.conv1 = GraphConv(in_feats, hidden_size)
        self.conv2 = GraphConv(hidden_size, num_classes)

    def forward(self, g, inputs):
        h = self.conv1(g, inputs)
        h = torch.relu(h)
        h = self.conv2(g, h)

        g.ndata['h'] = h
        hg = dgl.mean_nodes(g, 'h')
        return hg
```

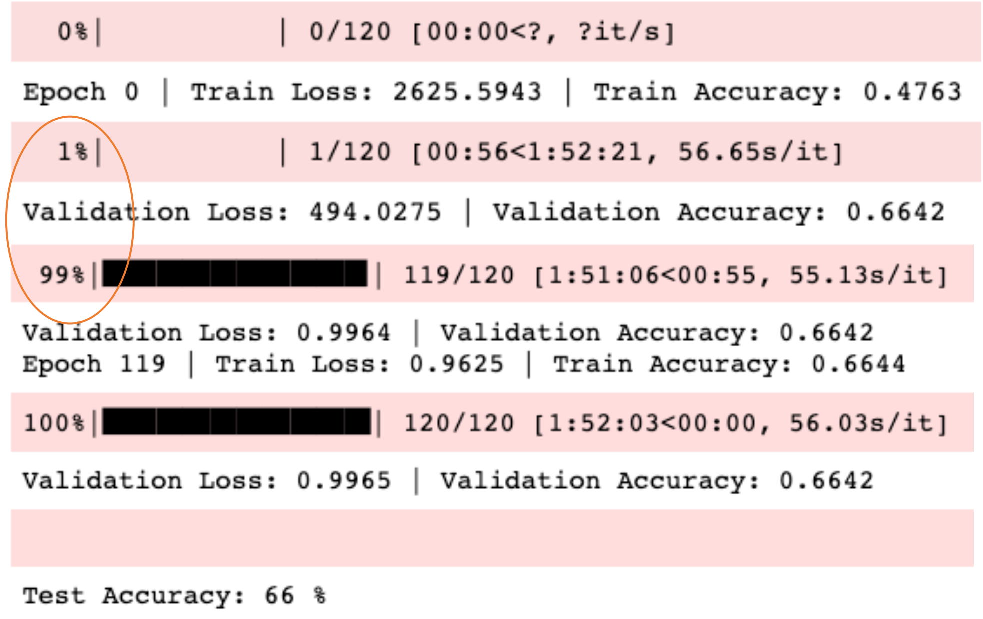
- Use the **old** verison of the dataset
- Use **DGL** to be our library
- DGL data format:



```
batched_g is like:
Graph(num_nodes=96, num_edges=160, ndata_schemes={'feat': Scheme(shape=(1,), dtype=torch.int64)}, edata_schemes={})
num_nodes = 3*batch_size, num_edges = 5*batch_size

labels is like: tensor([ 76,   0,   0,   0,   0,   0,   0,   0,   0,  76,   0,  76,   0,   0,
                          0,   0,  76,   0,  30,  92,   0,   0,  76,   0,   0,   0,   0,   0,
                        116,   0,  76,  76])
...
```

Result:



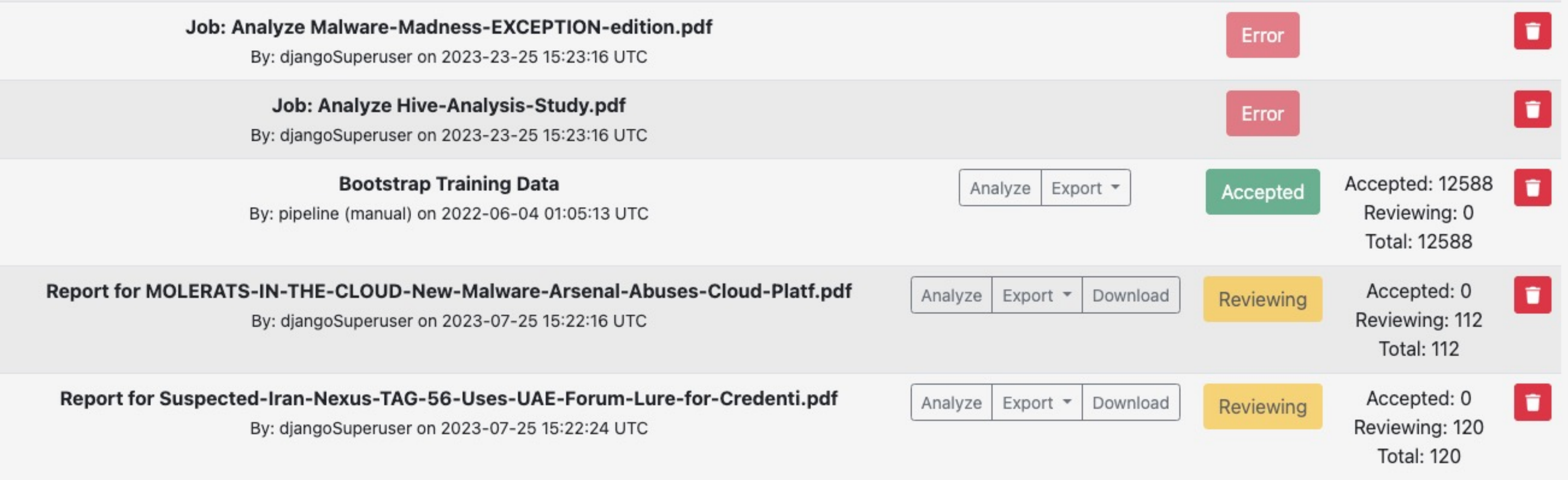
```
  0%|          | 0/120 [00:00<?, ?it/s]
Epoch 0 | Train Loss: 2625.5943 | Train Accuracy: 0.4763
  1%|          | 1/120 [00:56<1:52:21, 56.65s/it]
Validation Loss: 494.0275 | Validation Accuracy: 0.6642
 99%|█████████▉| 119/120 [1:51:06<00:55, 55.13s/it]
Validation Loss: 0.9964 | Validation Accuracy: 0.6642
Epoch 119 | Train Loss: 0.9625 | Train Accuracy: 0.6644
100%|██████████| 120/120 [1:52:03<00:00, 56.03s/it]
Validation Loss: 0.9965 | Validation Accuracy: 0.6642

Test Accuracy: 66 %
```

- GAT applied on the old data has the similar result

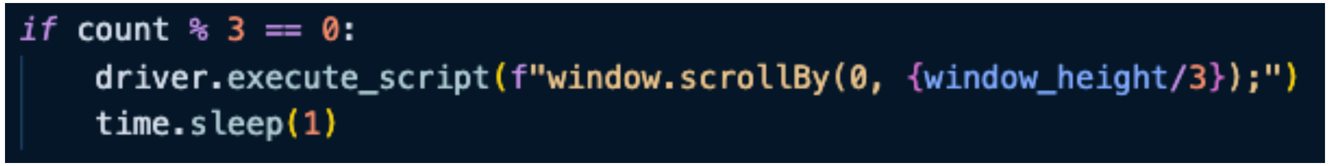
TRAM

Automation



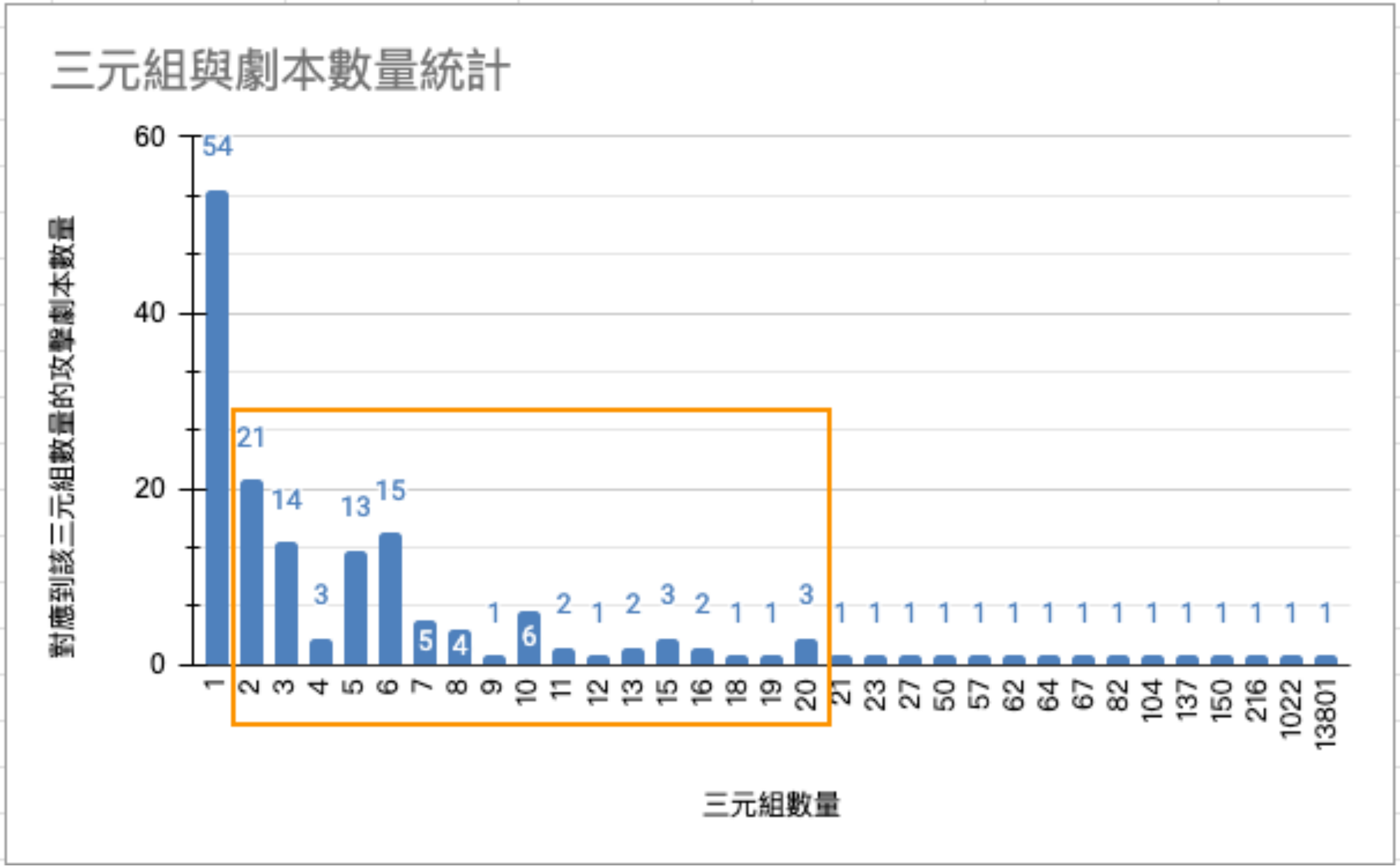
Job: Analyze Malware-Madness-EXCEPTION-edition.pdf By: djangoSuperuser on 2023-23-25 15:23:16 UTC		Error		
Job: Analyze Hive-Analysis-Study.pdf By: djangoSuperuser on 2023-23-25 15:23:16 UTC		Error		
Bootstrap Training Data By: pipeline (manual) on 2022-06-04 01:05:13 UTC	Analyze | Export ▾	Accepted	Accepted: 12588 Reviewing: 0 Total: 12588	
Report for MOLERATS-IN-THE-CLOUD-New-Malware-Arsenal-Abuses-Cloud-Platf.pdf By: djangoSuperuser on 2023-07-25 15:22:16 UTC	Analyze | Export ▾ | Download	Reviewing	Accepted: 0 Reviewing: 112 Total: 112	
Report for Suspected-Iran-Nexus-TAG-56-Uses-UAE-Forum-Lure-for-Credenti.pdf By: djangoSuperuser on 2023-07-25 15:22:24 UTC	Analyze | Export ▾ | Download	Reviewing	Accepted: 0 Reviewing: 120 Total: 120	

- Successfully **upload** the pdf files
- Successfully **export** the pdf files
  - Click 3 times and then scroll $\frac{1}{3}$ of the window size



```
if count % 3 == 0:
    driver.execute_script(f"window.scrollBy(0, {window_height/3});")
    time.sleep(1)
```

Appendix


三元組與劇本數量統計
對應到該三元組數量的攻擊劇本數量
60
40
20
0
54
21
14
3
13
15
5
4
1
6
2
1
2
3
2
1
1
3
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
2
3
4
5
6
7
8
9
10
11
12
13
15
16
18
19
20
21
23
27
50
57
62
64
67
82
104
137
150
216
1022
13801
三元組數量