Tennessee TECH

SNAPSKETCH: Graph Representation Approach for Anomaly Detection in Graph Stream

Ramesh Paudel and Dr. William Eberle
Department of Computer Science

Introduction

- Identify denial of service attacks, port scans, and other cyber-attacks using network graphs.
- Unique approach that identifies anomalous hotspots by tracking sudden increases/decreases edges connecting to a vertex; or the sudden (dis)appearance of edges with high weight
- **SNAPSKETCH** is fully unsupervised, has constant memory space usage, and can be used for real-time anomaly detection.

Research Objective

Problem Statement:

Given a graph stream $G_s = \{G_1, G_2, \dots, G_t, \dots\}$, *our goal is to learn a graph representation function* f *for each graph* $G_t \in \mathbb{R}^{|v|^2}$ *such that*

$$f : G_t \rightarrow v_{G_t} \in \mathbb{Z}^d \text{ and } d \ll |v|^2$$

and using v_{G_t} *detect whether a graph* G_t *at any time* t *contain an anomalous hotspot.*

Goals

- Generate a fixed-size feature vector (**SNAPSKETCH**) to represent a graph in a graph stream.
- Detect DoS attack (a type of anomalous hotspot) in network traffic using a **SNAPSKETCH**.

Experimentation

- Run RRCF [1] anomaly detection algorithm on sketch vector generated by **SNAPSKETCH** generated, Spotlight [3], and StreamSpot [2] on the following two datasets and compare their performances.

Dataset	# of Graph	# of Anomalies	Edges
Smart Homes IoT	9,678	1,007	29,959,737
DARPA 1998	3,497	361	3,904,797

SNAPSKETCH Framework

- Perform node2vec [5] random walk on the graph and construct n-shingles.
- Identify discriminative shingles (shingles with the highest frequency) and randomly project them into a d-dimensional projection h_d.
- Sketch graphs using a simplified hashing of projection vector h_d and the cost of shingles c_t.
- The sketching converts the graph G_t into a d-dimensional sketch vector v_{G_t}.
- Detect anomalous hotspot using RRCF [2] in the sketch vector.
- **SNAPSKETCH** has several advantages, fully unsupervised learning, constant memory space usage, entire-graph embedding, and real-time anomaly detection.

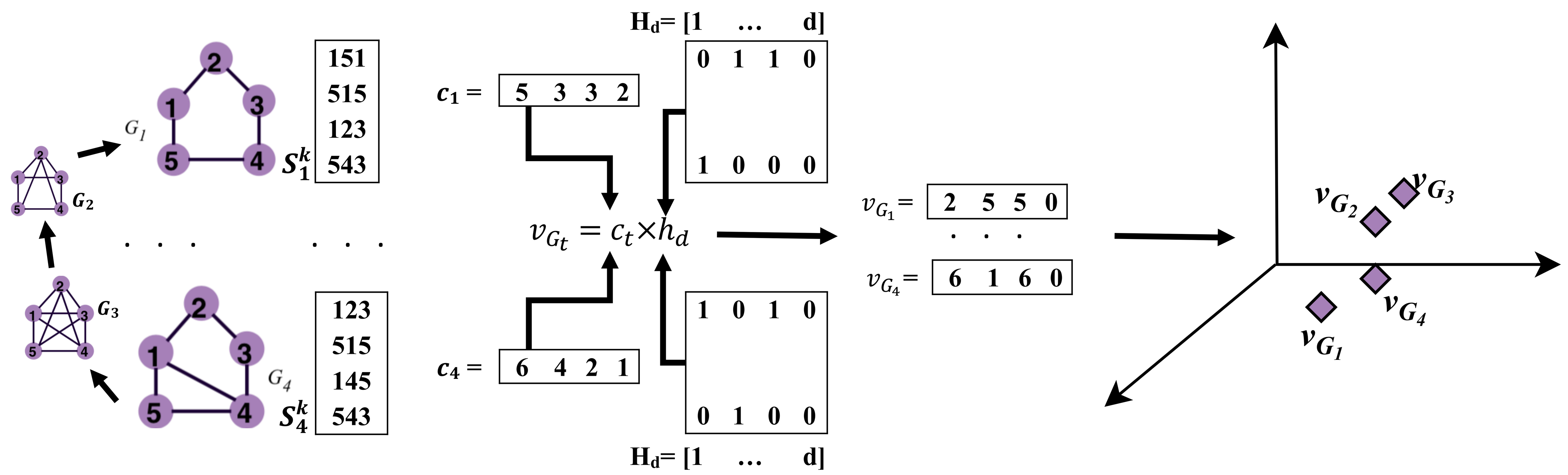

Fig 1. An Illustration of **SNAPSKETCH** framework

Real-time Anomaly Score

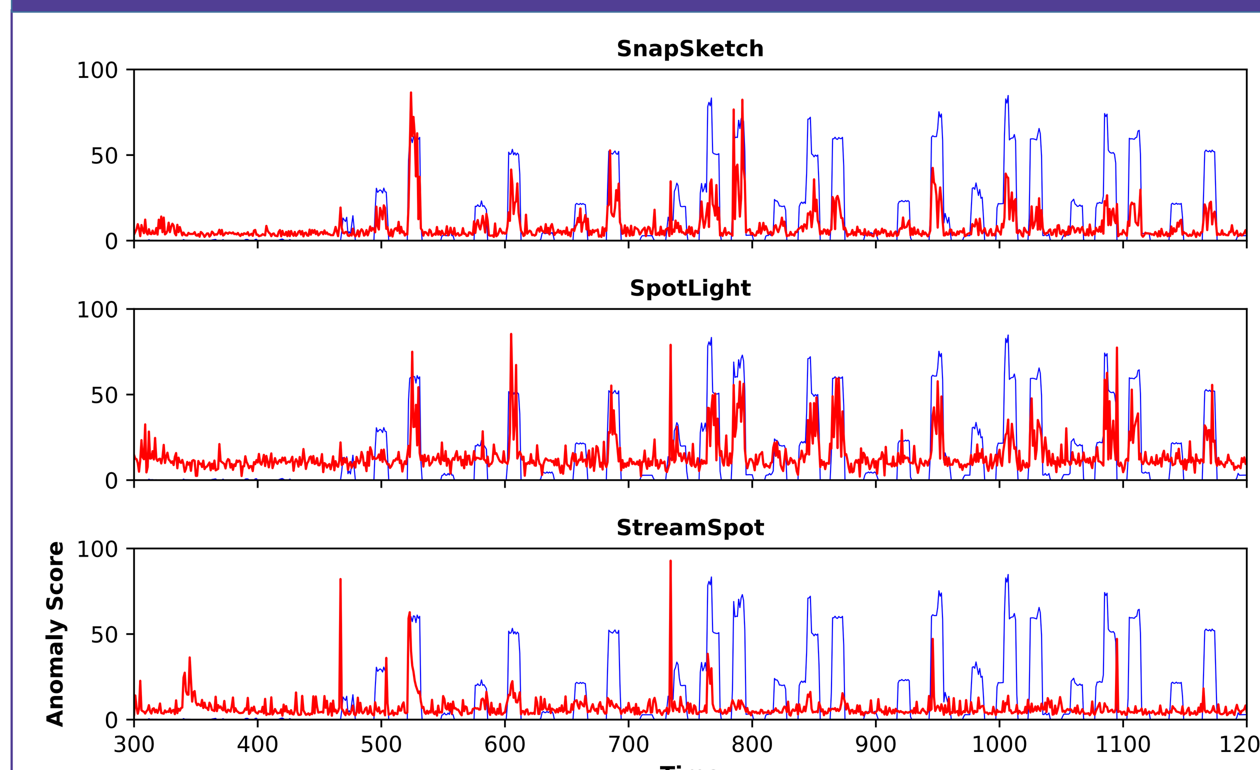

Fig 2. Anomaly score reported on smart home IoT traffic. Blue plot indicates the ground truth anomalies. Spike in red plots indicates the anomaly score reported by the respective approaches over time.

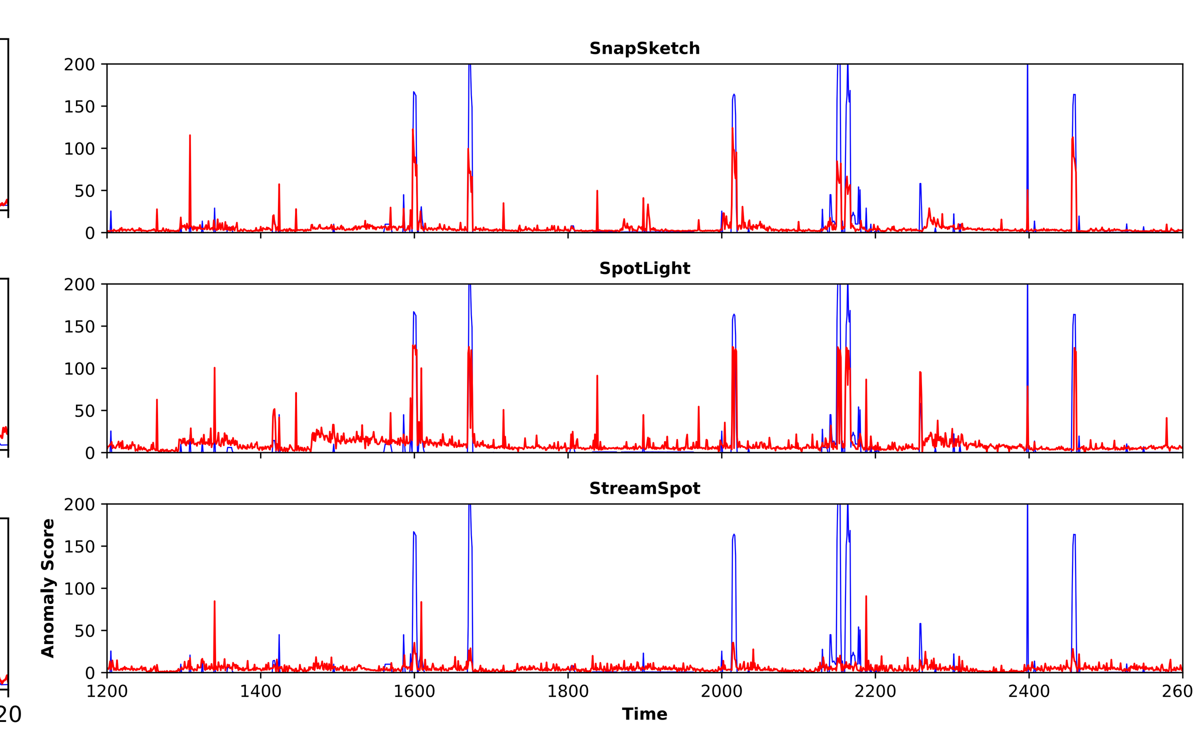

Fig 3: Anomaly score reported on DARPA dataset.

SNAPSKETCH Algorithm

```
Algorithm 1: SnapSketch Algorithm
  Input: Graph Stream G_s ={G_1, G_2, ..., G_t, ...}
  Parameters: Sketch Dimension d, Number of
              Discriminative Shingle k, Walk Length l, Size
              of Shingle n
  Output: Graph Sketch v_{G_t}
Function Main(G_s, d, p, k, l, n):
    while not end of stream do
        p_{G_t} ← node2vecWalk(G_t, l)
        S_t ← [p_{G_t}[i : i + n] for i in range ( len(p_{G_t})−(n − 1))]
        S_t^k ← S_t.sort(reverse=True)[: k] //get
         k−discriminative shingles
        h_d ←Hashing(S_t^k, d, r = 0.2)
        v_{G_t} ∪ Sketching(S_t^k, h_d)
        Anom_score ← RRCF(v_{G_t})
        d
    end
return v_{G_t}
Function Hashing(S_k, d, r):
    for S_i = S_k[1, ..., k] do
        h_d ∪ random([0,1], d, p=[1-r, r])
    end
return h_d
Function Sketching(S_k, h_d):
    for S_i = S_k[1, ..., k] do
        c_t ∪ w_{s_i} × r_{s_i}
    end
    v_{G_t} = c_t × h_d
return v_{G_t}
```

Results

Algorithm	Precision (top−m) 100	200	300	Recall (top−m) 100	200	300
Smart Home IOT Dataset						
Ground Truth	*1.0*	*1.0*	*1.0*	*.099*	*.198*	*.298*
SnapSketch	**.94**	**.86**	**.80**	**.093**	**.170**	**.239**
SpotLight	.77	.73	.63	.076	.145	.190
StreamSpot	.69	.57	.54	.068	.114	.161
DARPA Dataset						
Ground Truth	*1.0*	*1.0*	*1.0*	*.277*	*.554*	*.831*
SnapSketch	**.83**	**.52**	**.34**	**.229**	**.288**	**.288**
SpotLight	.80	.51	.34	.221	.282	.282
StreamSpot	.49	.29	.20	.135	.160	.163

Conclusion

- **SNAPSKETCH** can effectively represent the graph into a fixed-size sketch vector.
- Using RRCF [1] on sketch vector anomalous events like denial-of-service attacks can be detected.
- **SNAPSKETCH** has better precision and recall than baseline SpotLight [3] and StreamSpot [2] approaches on top –m anomalous graphs.

Reference

1. Guha, Sudipto, Mishra, Nina, Roy, Gourav, and Schrijvers, Okke. "Robust random cut forest based anomaly detection on streams." In International conference on machine learning," pages 2712-2721, 2016.
2. Manzoor, Emaad, Milajerdi, Sadegh M, and Akoglu, Leman. "Fast memory efficient anomaly detection in streaming heterogeneous graphs." In "Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining," pages 1035-1044. ACM, 2016.
3. Eswaran, Dhivya, Faloutsos, Christos, Guha, Sudipto, and Mishra, Nina. "Spotlight: Detecting anomalies in streaming graphs." In "Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining," pages 1378-1386. ACM, 2018.
4. Grover, Aditya and Leskovec, Jure. "node2vec: Scalable feature learning for networks." In "Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discoveryand data mining," pages 855-864. ACM, 2016.

Acknowledgements

Funding provided by Tennessee Tech University, College of Engineering for achieving Carnegie classification.