# Detecting DoS Attack in Smart Home IoT Devices Using a Graph-Based Approach

Ramesh Paudel, Timothy Muncy, William Eberle

Graduate Student Seminar

Fall 2019

# Agendas


**G**raph-based **O**utlier **D**etection
in **I**nternet of **T**hings

# Smart Home

Internet of Things (IoT) devices are designed to optimize, automate, and improve quality of life.

Device for Energy Management, Cameras, Appliances, Health-Monitor, Controllers, etc. are actively connected to the internet.

We must begin to address the security risks associated with them


[1]

# Attack Examples

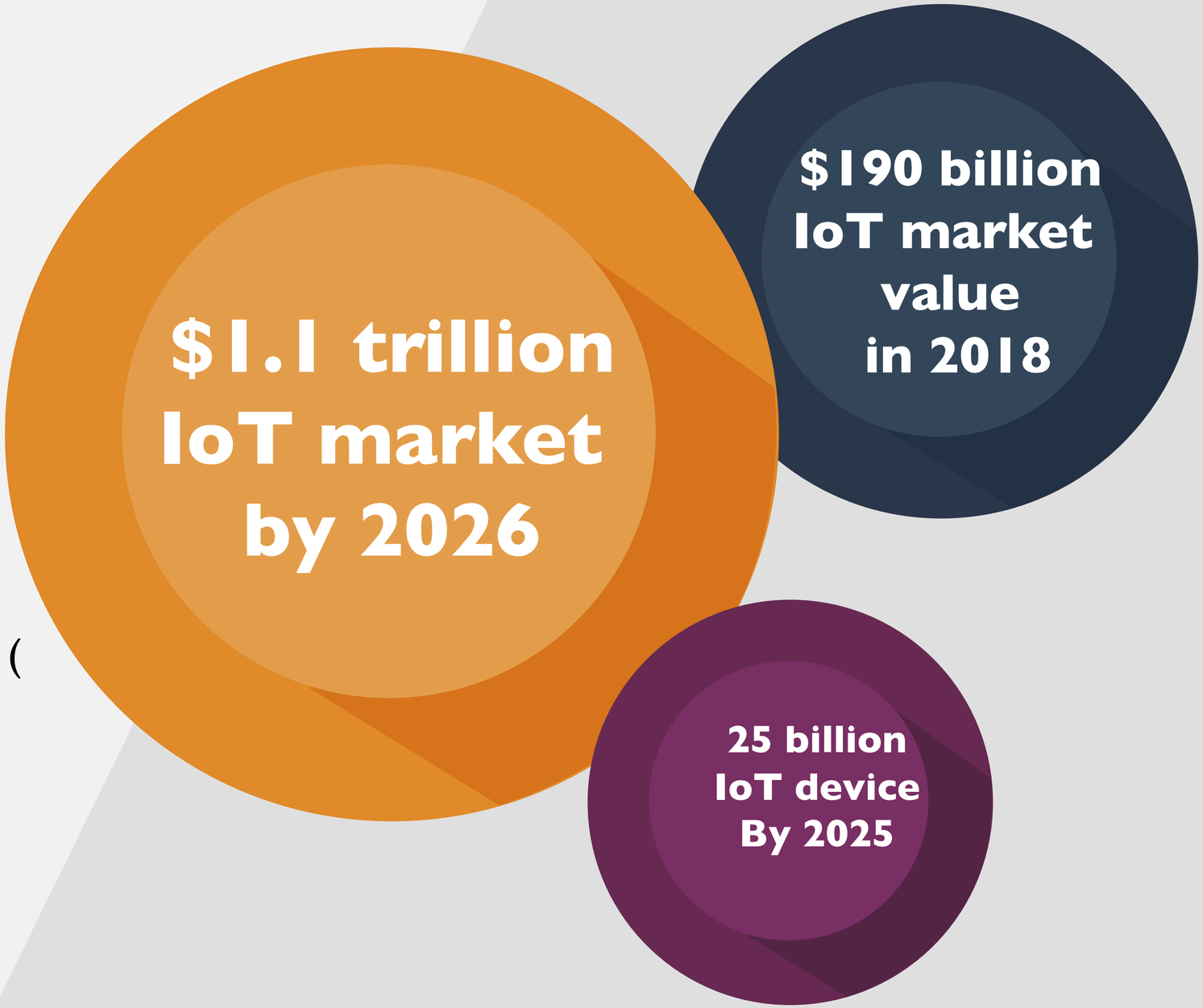

- Mirai malware amassed IoT devices (Cameras + DVR Players) as botnet to do DDoS attack on Dyn Infrastructure.
- 100,000 malicious endpoints
- 1.2Tbps attack strength
- Took down internet of East Cost


- Hacker got into the fish tank sensor
- Used it to move around into other areas of the network and sent out data
- 10 GB of data were sent out to a device in Finland.
- Reported by Darktrace

# Statistics [2]

**IoT Vulnerabilities:**

- IoT devices were attacked within 5 min after connecting to the internet
- 32.7 million IoT malware attack in 2018 ( ↑ by 55% in 2019)



# What is the problem?

## Current Threat Detection Approaches:

| Methods | Drawbacks |
|---|---|
| Signature-based | • Ineffective to detect new attacks and variants of known attacks. |
| Anomaly-based | • Learning the scope of the normal behavior is not a simple task.<br>• Hence has high false positive rates. |
| Specification-based | • Human expert should manually define the rules of each specification.<br>• This may not adapt to different environments and could be time-consuming and error-prone. |
| Hybrid Approach | • Computationally expensive |

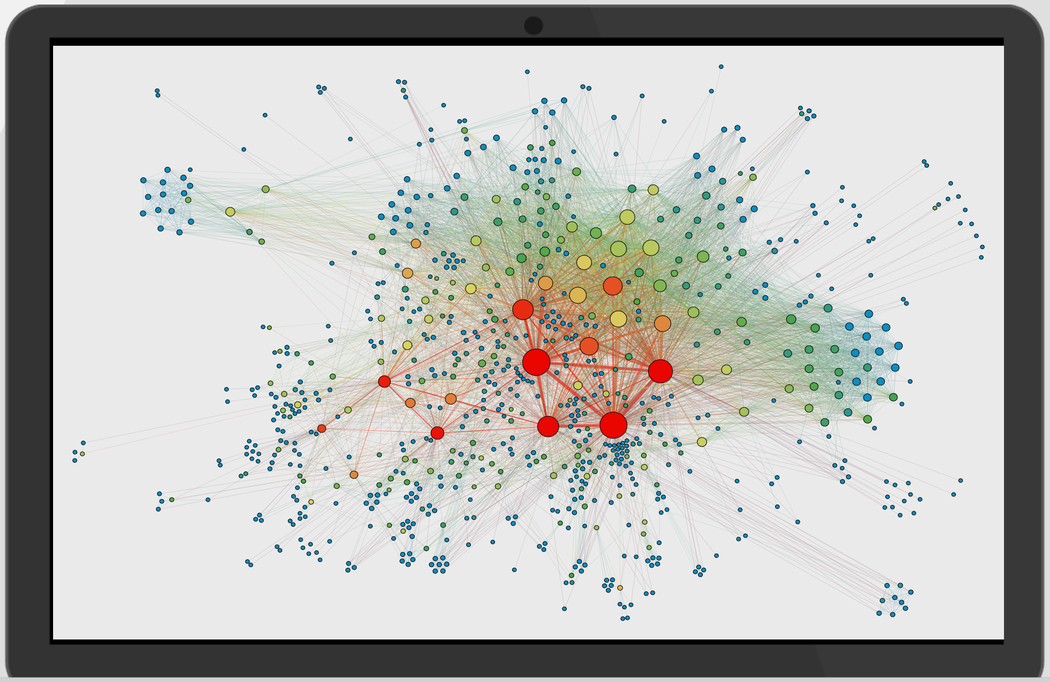
## Current security solutions in IoT devices:

- Has little to no significant security (like password).
- Computationally expensive and high communication overhead.
- Unscalable as they require known attack signatures or full packet inspection.

# Graph-based Approach

## Why Graphs?

- A graph naturally represents highly relational and inter-dependent data
- In IoT network, device can be treated as a node and the communication as edges.
- Anomalies could exhibit themselves as relational.
- Only need the source and destination IP to build a network graph.
- Do not need any extra information like traffic packet size, protocol, etc.,



## Proposed Solutions :

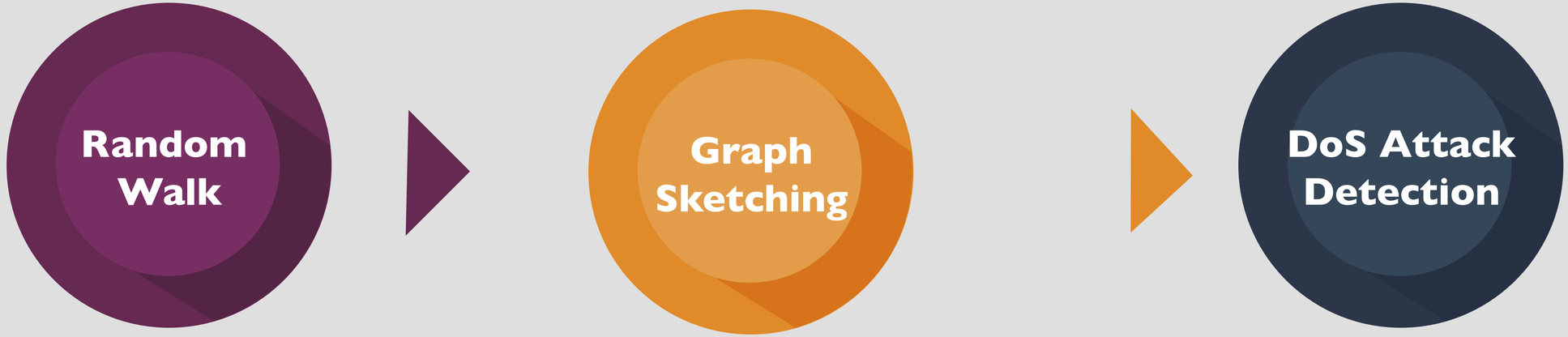
We propose a novel approach called Graph-based Outlier Detection in Internet of Things (GODIT) that

(i) Represents smart home IoT traffic as a real-time graph stream,

(ii) Efficiently processes graph data, and

(iii) Detects DoS attacks in real-time.

# **G**raph-based **O**utlier **D**etection in **I**nternet of **T**hings (GODIT)

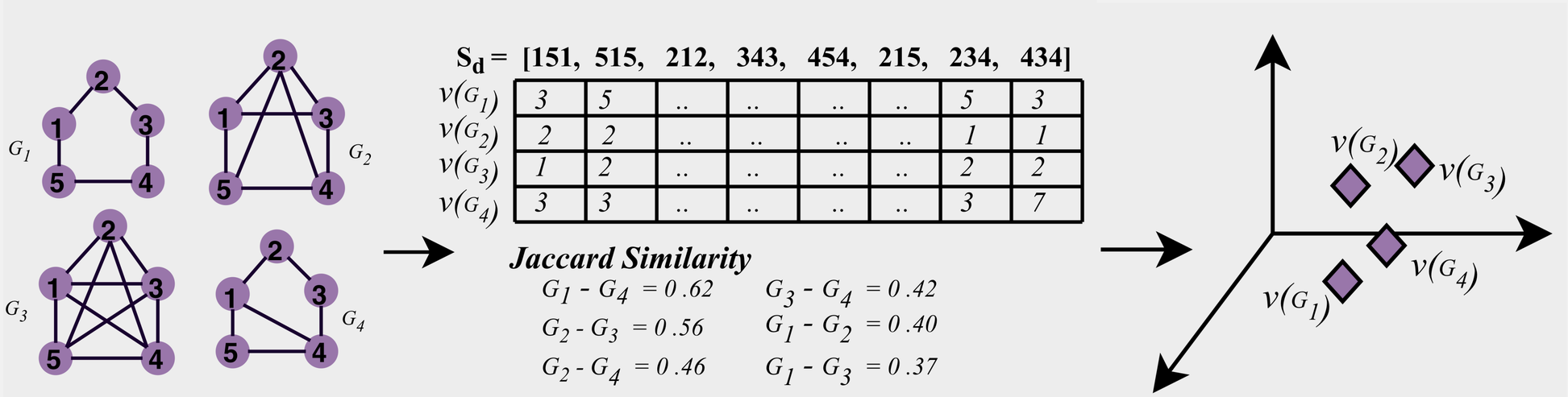



Perform biased random walk to produce a $n$ −shingle (like $n-gram$ ).

Generating a fixed-sized $d$ −dimensional graph sketch using discriminative shingles

DoS attack detection in graph stream using Robust Random Cut Forests (RRCF)

# GODIT Overview




- Represents smart home IoT traffic as a real-time graph stream
- Perform biased random walk to generate a $n$ −shingle (like $n-gram$ ).

- Find top $d$ − discriminative shingles using entropy-based measures.
- Generating a fixed-sized $d$ −dimensional graph sketch using discriminative shingles
- Sketch vector hold cost of $S_d$ in each graph $G_t$.

- Anomaly (DoS) detection on graph sketches
- The similar graph share higher proximity (shown by Jaccard similarity) and dissimilar graph are mapped far

# Shingling using Biased Random Walk

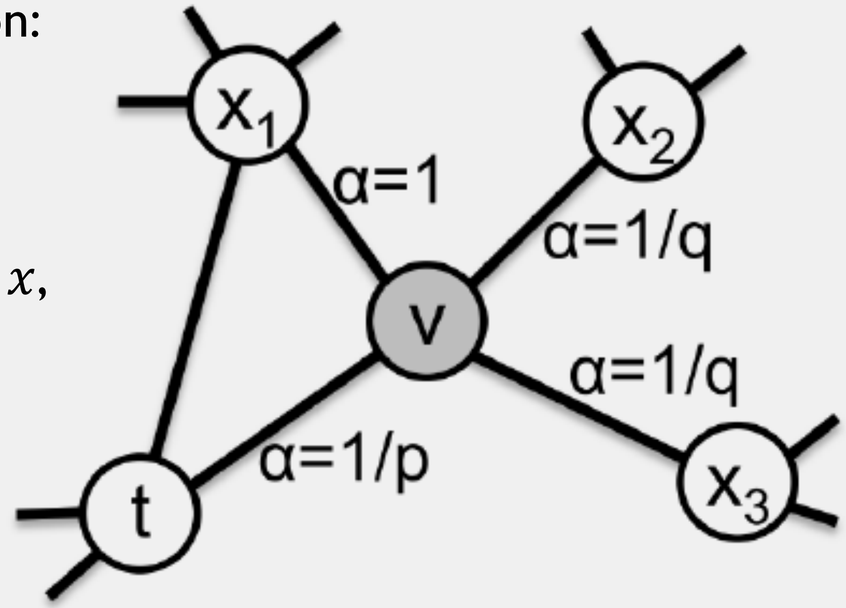
- Each $i^{th}$ node in the walk path ($c_i$) are generated using distribution:

$$P(c_i = x | c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{Z} & \text{if } (v,x) \varepsilon e \\ 0 & \text{otherwise} \end{cases}$$

Where,

$\pi_{vx}$ *is the un-normalized transition probability between node* $v$ *and node* $x$,

$Z$ *is the normalizing constant.*




[9]

**Bias Walk:**

- Interpolate between BFS and DFS using parameter $p$ and $q$
- If a random walk just traversed the edge $(t, v)$ and is currently at node $v$, the next step to the node $x$ leading from the current node $v$ can be decided using transitional probability $\pi_{vx} = \alpha_{pq(t,x)}.wvx$

where $w_{vx}$ is the weight of the edge $(v, x)$ and

$$\alpha_{pq}(t,x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ 0 & \text{if } d_{tx} = 2 \end{cases}$$

# Graph Sketching using Discriminative Shingle

Entropy of the shingle $S_j$ in the training window $W$

$$E(S_j) = -P(G_i|S_j)\sum_{i=1}^{N} P(S_j|G_i)log_2 P(S_j|G_i)$$

Where,

$N$ is the total number of graphs $S_j$ present in

$P(\ G_i|S_j)$ is the fraction of graphs $S_j$ is present in

$P(S_j|G_i) = \frac{r_{S_j}^{G_i}}{\sum_{i=1}^{|G|} r_{S_j}^{G_i}}$ is the probability of each shingle $S_j$ in the graph $G_i$ and $r_{s_j}^{G_i}$ is the occurrences of shingle $S_j$ in graph $G_i$

- Discriminative shingles are the top $d$ shingles with the highest entropy value
- Sketch vector $S_d$ for a graph $G_i$ will hold the cost of each discriminative shingle in graph $G_i$ .
- The cost of a shingle $S_j$ in graph $G_i$ is defined as: $\boldsymbol{S_j^{cost} = wr_{s_j}^{G_i}}$

  where $w$ is the sum of edge weight in the shingle $S_j$

  $r_{s_j}^{G_i}$ is the frequency of $S_j$ in $G_i$

# Anomaly Detection

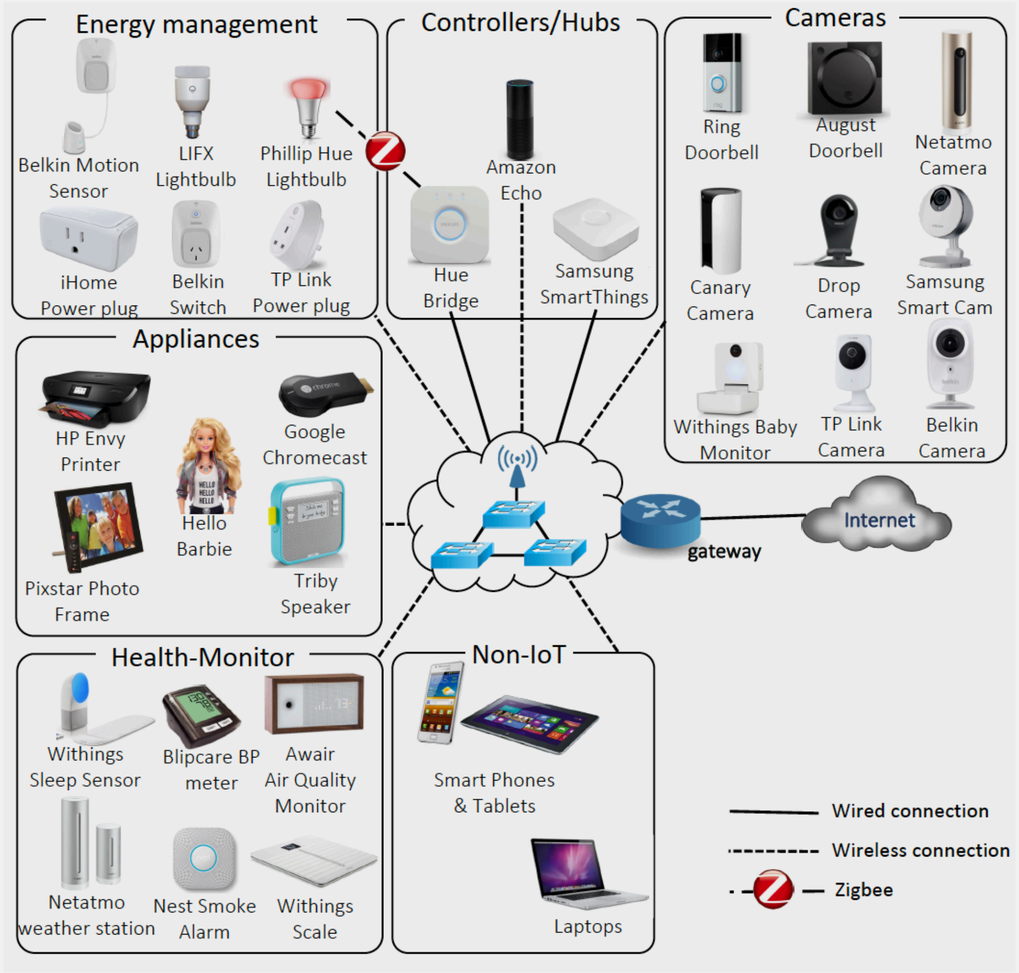
- The anomaly detection is performed by using the state-of-the-art Robust Random Cut Forests (RRCF) [4].
- RRCF algorithm is an unsupervised ensemble method for detecting outliers in streaming data.
- We initialize RRCF using 100 trees and 256 samples.
- RRCF initializes the forest of 100 trees by using the sketch vector of the first 256 graphs.

# Dataset

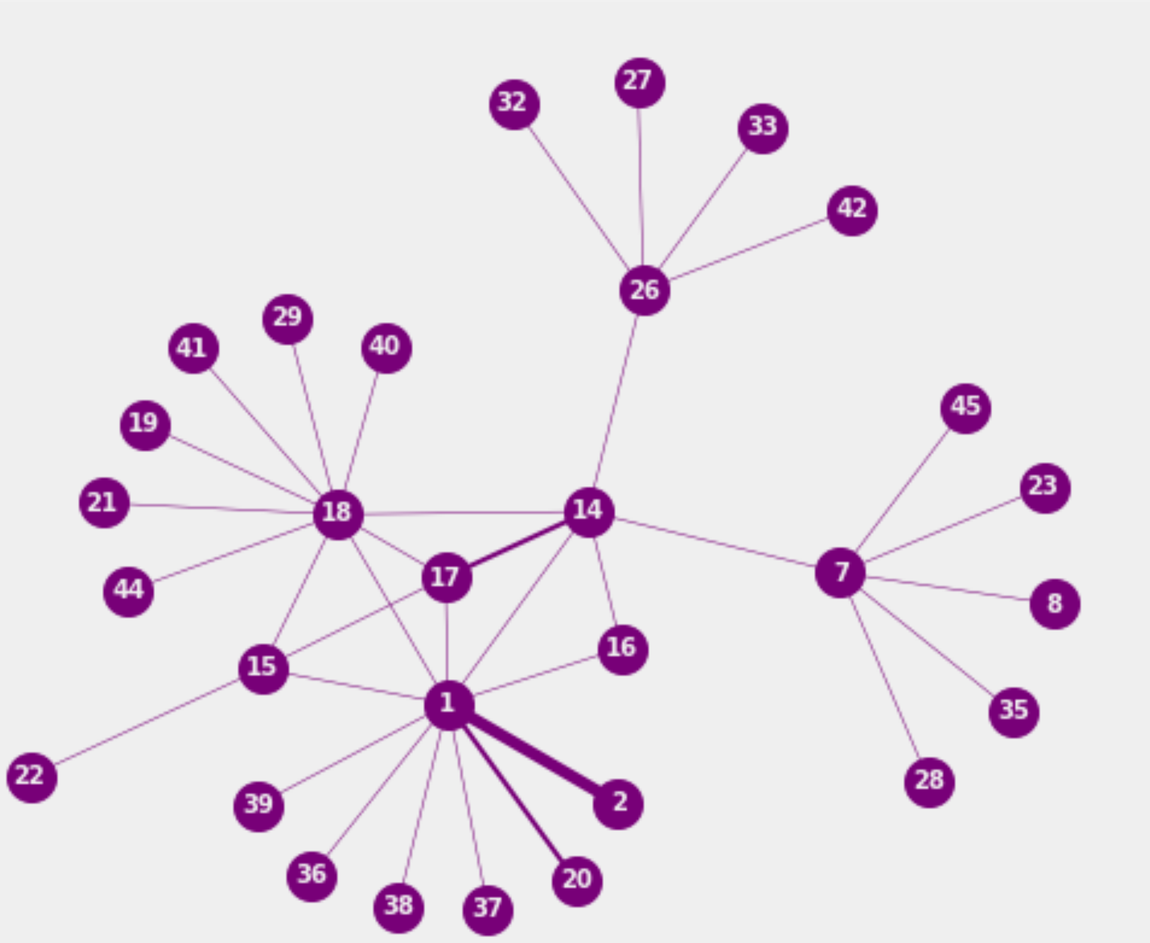
- Collected from the University of New South Wales Sydney (UNSW Sydney) smart home test-bed [8].
- Smart home that have 28 IoT devices as well as non-IoT devices.
- Each DoS attack lasts for 10 minutes and is launched with the maximum limit of 1, 10 or 100 packets per second.
- Based on this ground truth, if an individual graph contains at least 50 edges belonging to a DoS attack, the graph was labeled anomalous.

| Dataset | # of Graph | # of Anomalies | # of Edges | Total Duration |
|---|---|---|---|---|
| Static Setting | 1,236 | 306 | 4,726,992 | 1 day |
| Streaming Setting | 9,678 | 1,291 | 29,959,737 | 1 week |

# Data Preprocessing



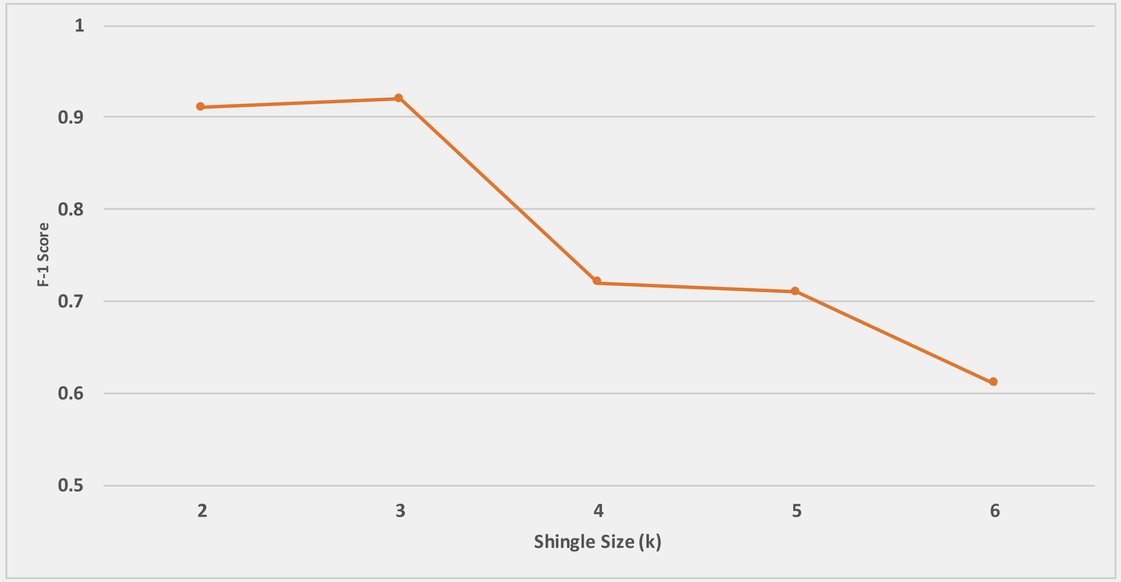

[8]

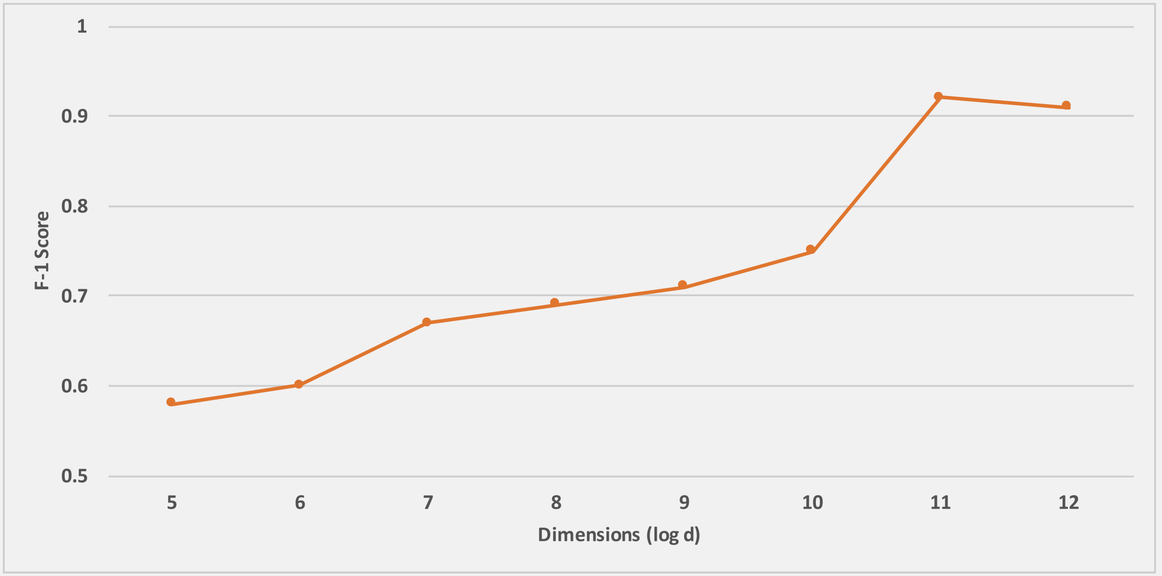


# Static Settings:

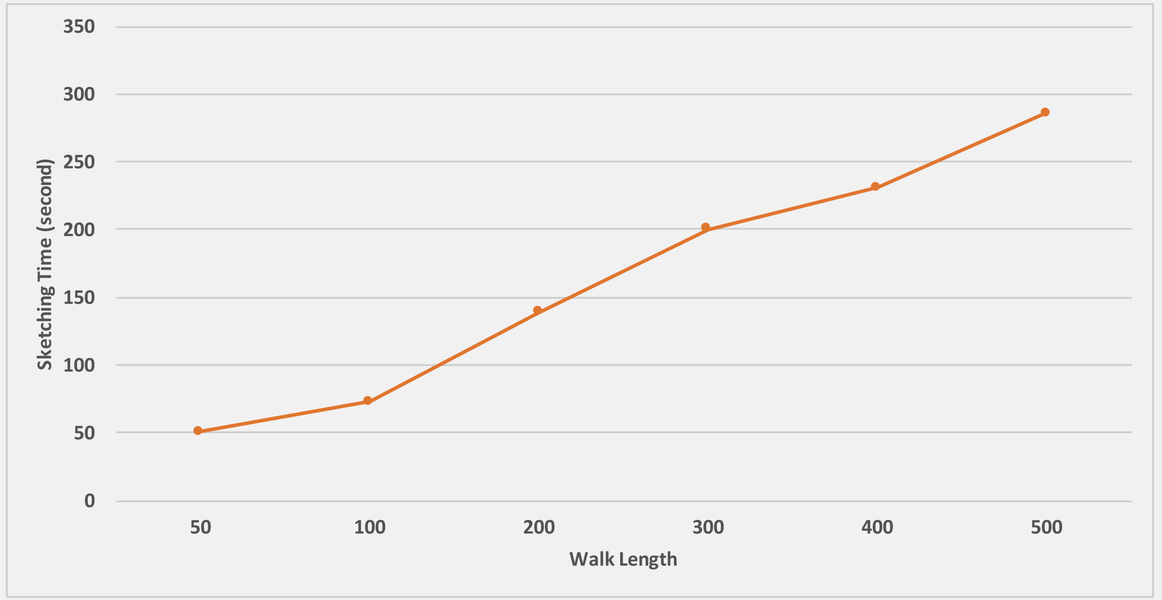
- Demonstrate that the proposed approach can classify normal and DoS attack graphs.
- One full day of data (June 01, 2018) where 5 IoT devices are involved in DoS attacks.
- Uses the sketch vector as input to the supervised machine learning algorithms: decision tree, support vector machine, gradient boosting, and random forest.
- Compare with standard machine learning approach developed by Doshi et al. [5].
- Doshi et al. [5] method primarily relies on the use of stateless features like packet size, inter-packet time interval, and protocol.

# Parameter Sensitivity







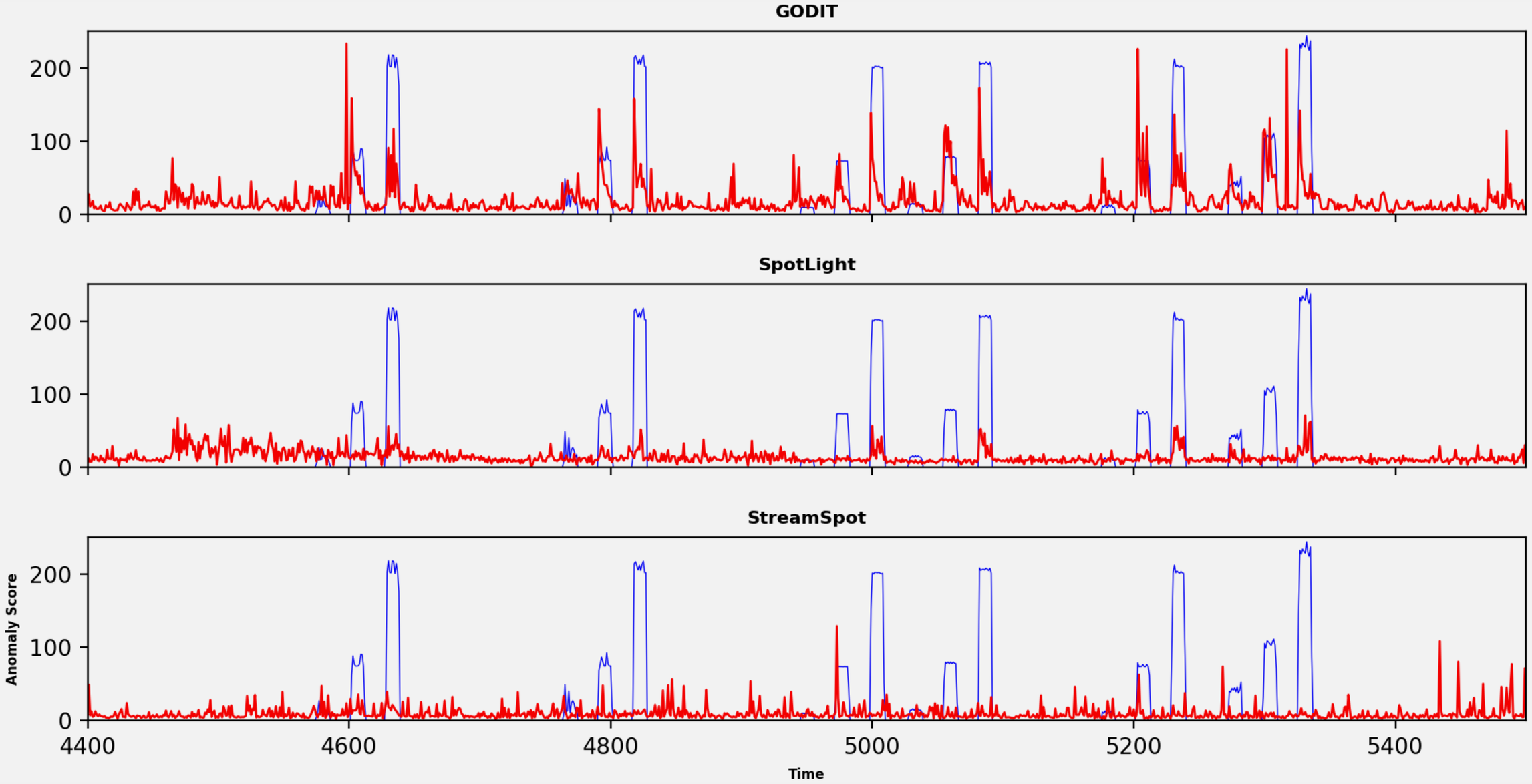
# Results

| Algorithm | Precision | Recall | F1-Score |
|---|---|---|---|
| **Our Approach** | | | |
| Decision Tree | 0.92 | **0.92** | **0.92** |
| Support Vector Machine | 0.89 | 0.84 | 0.87 |
| Gradient Boosting | 0.97 | 0.73 | 0.84 |
| Random Forest | **0.98** | 0.73 | 0.83 |
| **Doshi et al.** | | | |
| Decision Tree | 0.92 | 0.88 | 0.90 |
| Support Vector Machine [1] | - | - | - |
| Gradient Boosting | 0.92 | 0.88 | 0.90 |
| Random Forest | 0.92 | 0.82 | 0.87 |

# Streaming Settings:

- One week of data with 29.9 million traffic packets.
- Each graph represents 1-minute of network traffic.
- Total 9,879 graphs each coming every minute in the graph stream.
- First 25% of the data is used as a training set to generate discriminative shingles and anomaly detection is done in subsequent 75% of the data.
- Compare against SpotLight [6] and StreamSpot [7].
- Evaluated using precision and recall for top k anomalous graphs.

# Results


GODIT
SpotLight
StreamSpot
Anomaly Score
0
100
200
4400
4600
4800
5000
5200
5400
Time

# Results

If there are a total $N$ DoS attack graphs, for every $k$ we compute

$$precision@k = TP(k)/K$$

$$recall@k = TP(k)/N.$$

| Algorithm | Precision (top$-k$) | | | | Recall (top$-k$) | | | |
|---|---|---|---|---|---|---|---|---|
| | 100 | 200 | 300 | 400 | 100 | 200 | 300 | 400 |
| *Ground Truth* | *1.0* | *1.0* | *1.0* | *1* | *.25* | *.49* | *.74* | *.99* |
| GODIT | **.91** | **.86** | **.80** | **.67** | **.22** | **.43** | **.59** | **.66** |
| SpotLight | .75 | .53 | .38 | .30 | .18 | .26 | .28 | .30 |
| StreamSpot | .61 | .52 | .42 | .33 | .15 | .26 | .31 | .33 |

# Conclusion

- Presented GODIT to detect DoS attacks in smart home IoT networks that
  - Represent the IoT traffic as a graph stream.
  - Embed a graph into a $d$ −dimensional sketch vector $S_d$ using top $d$ − discriminative $n$ − shingle.
  - Detect DoS attack using sketch vector.
  - Outperform current graph stream anomaly detection approaches in terms of both precision and recall.

# Future Works

- GODIT requires periodical training/updates to match the latest trends (while generating discriminative shingles) if the network traffic is going through the concept drift.
- Neighborhood information is included (using BFS and DFS), but the structure of the graph (number of neighbors, degree, etc.) is omitted.
- Would like to investigate an approach to integrate this structural information in our sketch vector, which would result in a rich set of information.

# References:


1. https://medium.com/datadriveninvestor/10-amazing-cases-of-iot-applications-taken-from-the-real-life-a8682cdb48d0
2. https://securityboulevard.com/2019/09/20-surprising-iot-statistics-you-dont-already-know/
3. https://www.forbes.com/sites/leemathews/2017/07/27/criminals-hacked-a-fish-tank-to-steal-data-from-a-casino/
4. Guha, Sudipto, et al. "Robust random cut forest based anomaly detection on streams." *International conference on machine learning*. 2016.
5. Doshi, Rohan, Noah Apthorpe, and Nick Feamster. "Machine learning ddos detection for consumer internet of things devices." *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018.
6. Eswaran, Dhivya, et al. "Spotlight: Detecting anomalies in streaming graphs." *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018.
7. Manzoor, Emaad, Sadegh M. Milajerdi, and Leman Akoglu. "Fast memory-efficient anomaly detection in streaming heterogeneous graphs." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.
8. A. Hamza, H. H. Gharakheili, T. A. Benson, and V. Sivaraman, "Detecting volumetric attacks on lot devices via sdn-based monitoring of mud activity," in Proceedings of the 2019 ACM Symposium on SDN Research. ACM, 2019, pp. 36–48.
9. A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2016, pp. 855–864.

# Thank You