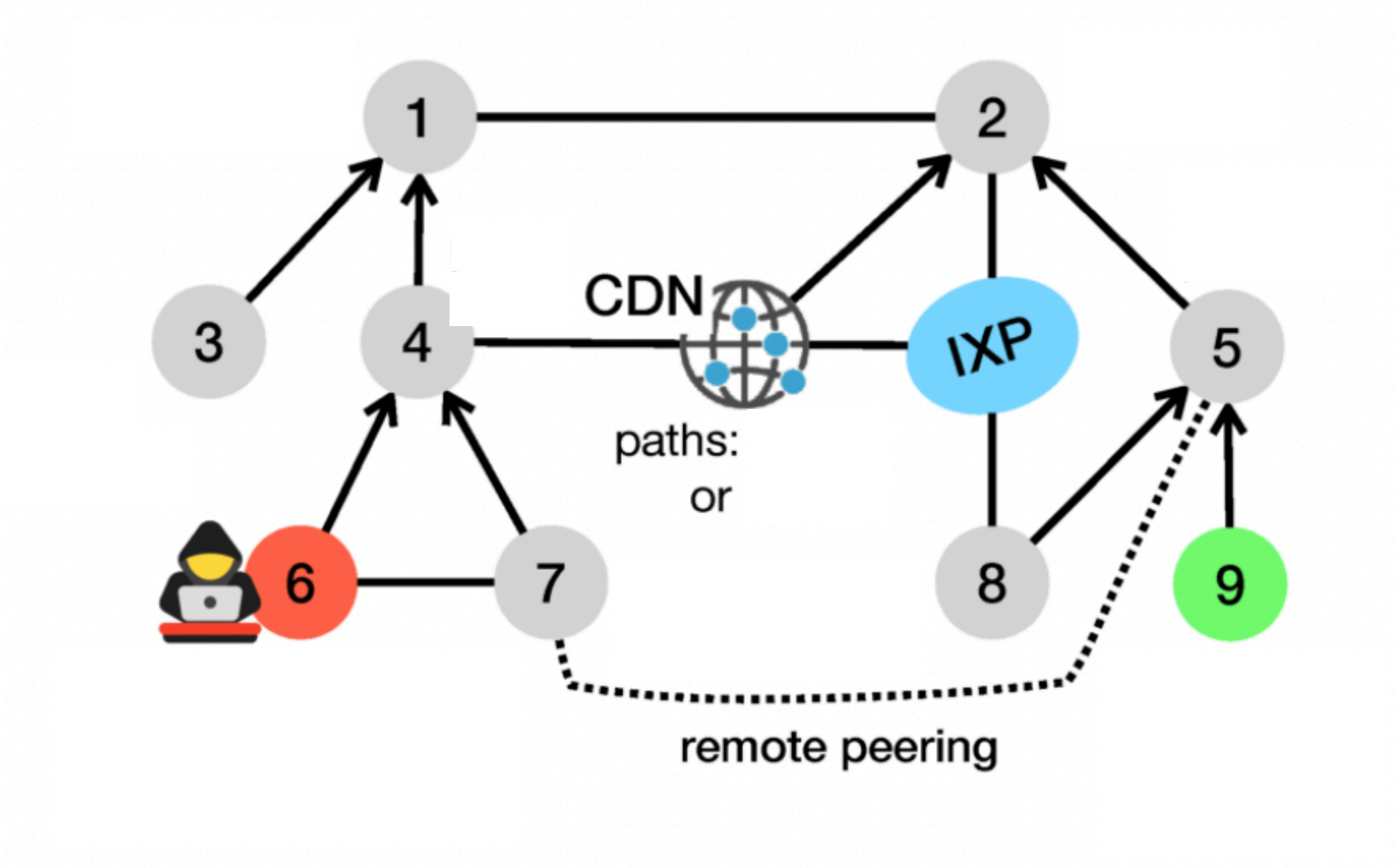
A System to Detect Forged-Origin BGP Hijacks

NSDI ‘24

mhkang 2024-08-20

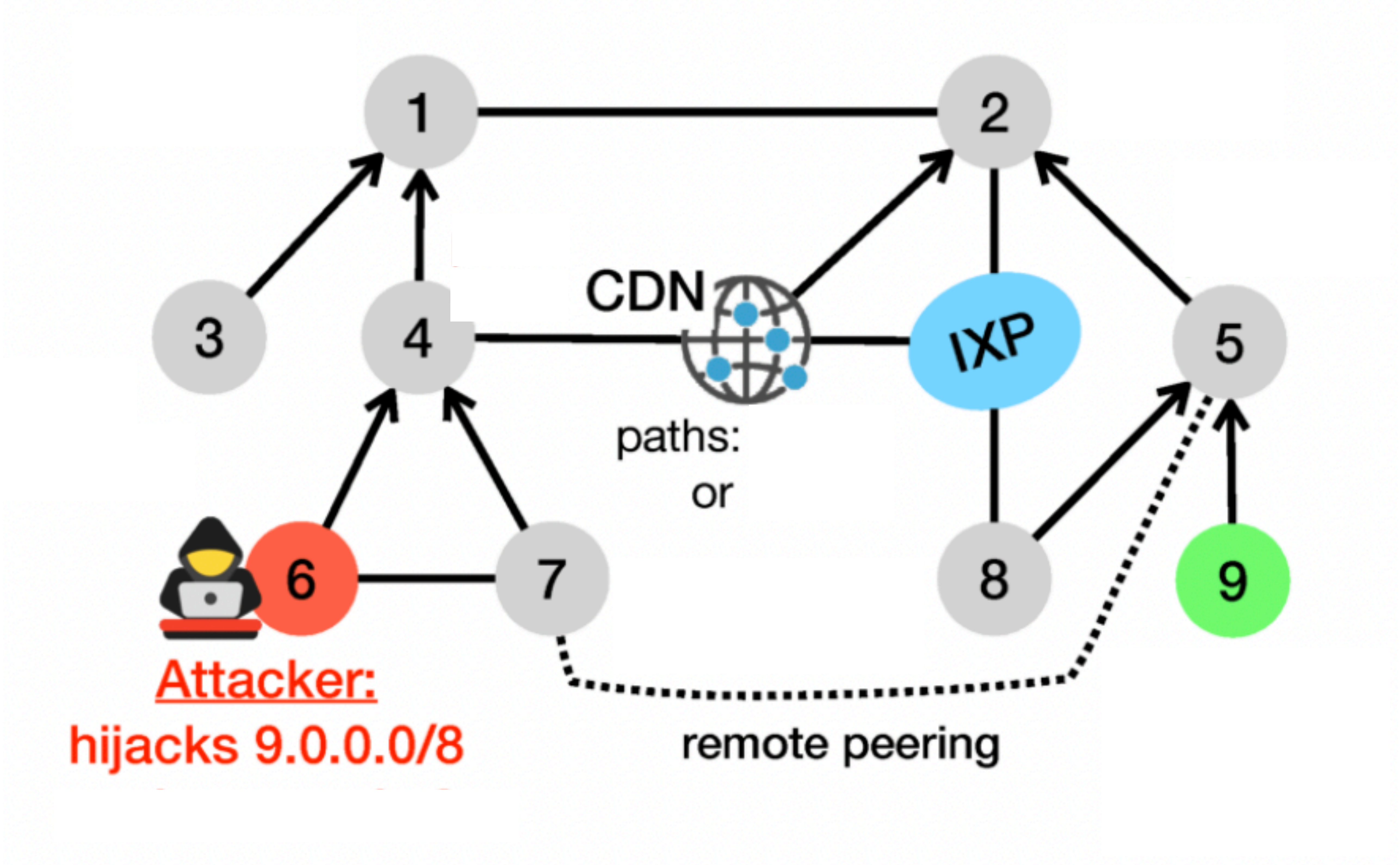
Forged-origin BGP hijack

- a BGP hijack attack
  - an attacker **announces forged AS paths** towards a victim prefix **by prepending the victim's origin AS number** to make them appear legitimate



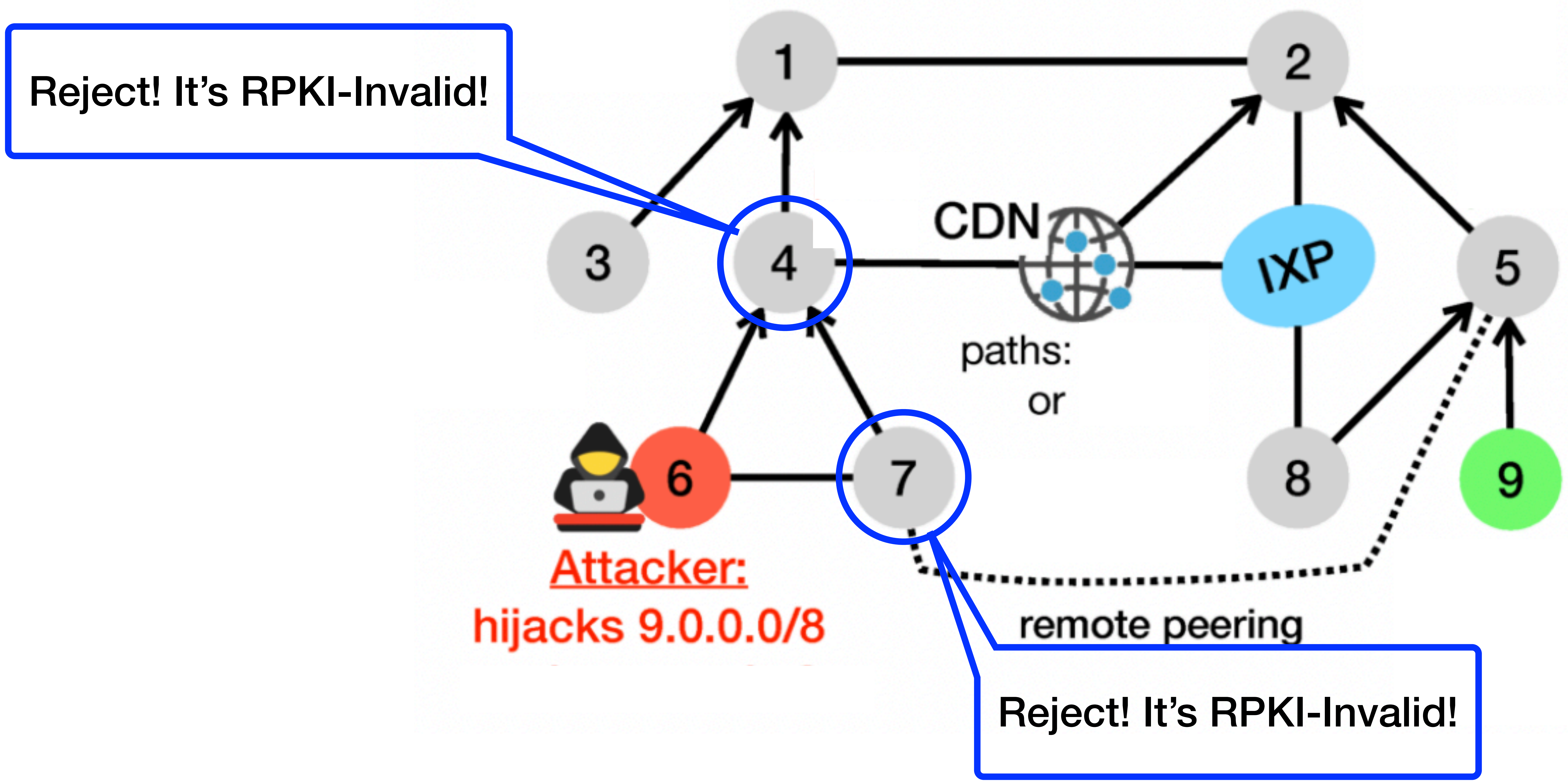
Forged-origin BGP hijack

- a BGP hijack attack
  - an attacker **announces forged AS paths** towards a victim prefix **by prepending the victim's origin AS number** to make them appear legitimate



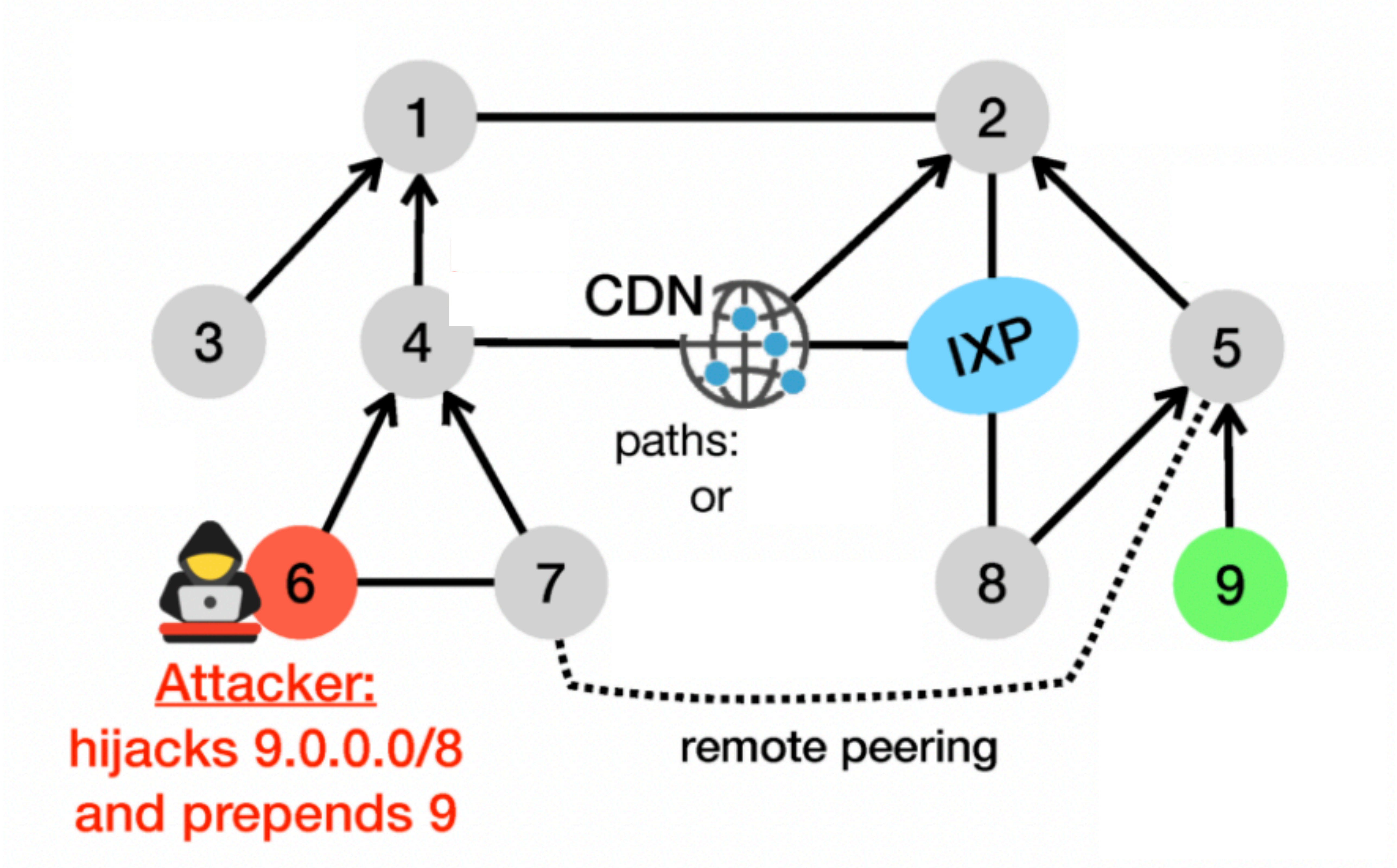
Forged-origin BGP hijack

- a BGP hijack attack
  - an attacker **announces forged AS paths** towards a victim prefix **by prepending the victim's origin AS number** to make them appear legitimate



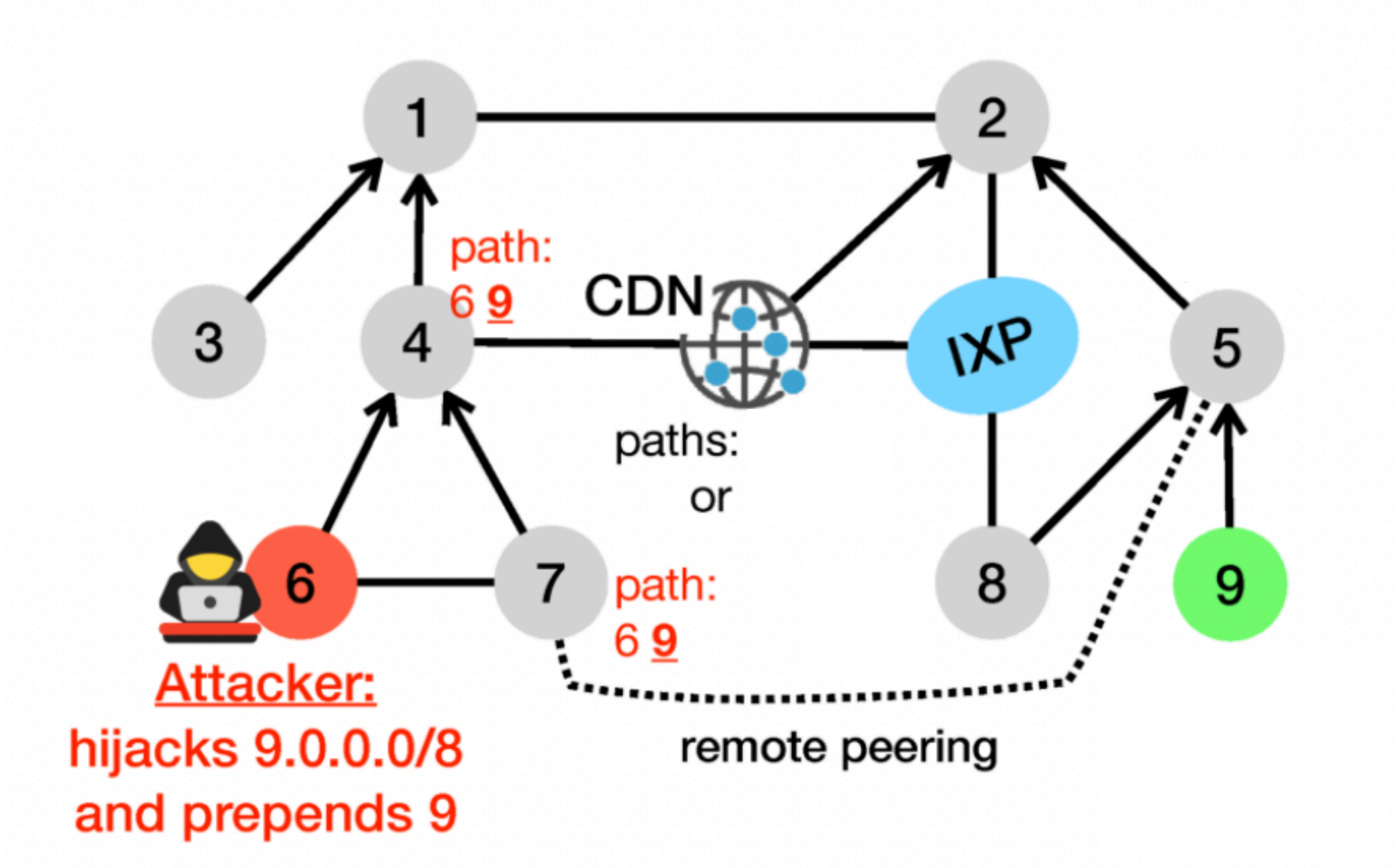
Forged-origin BGP hijack

- a BGP hijack attack
  - an attacker **announces forged AS paths** towards a victim prefix **by prepending the victim's origin AS number** to make them appear legitimate



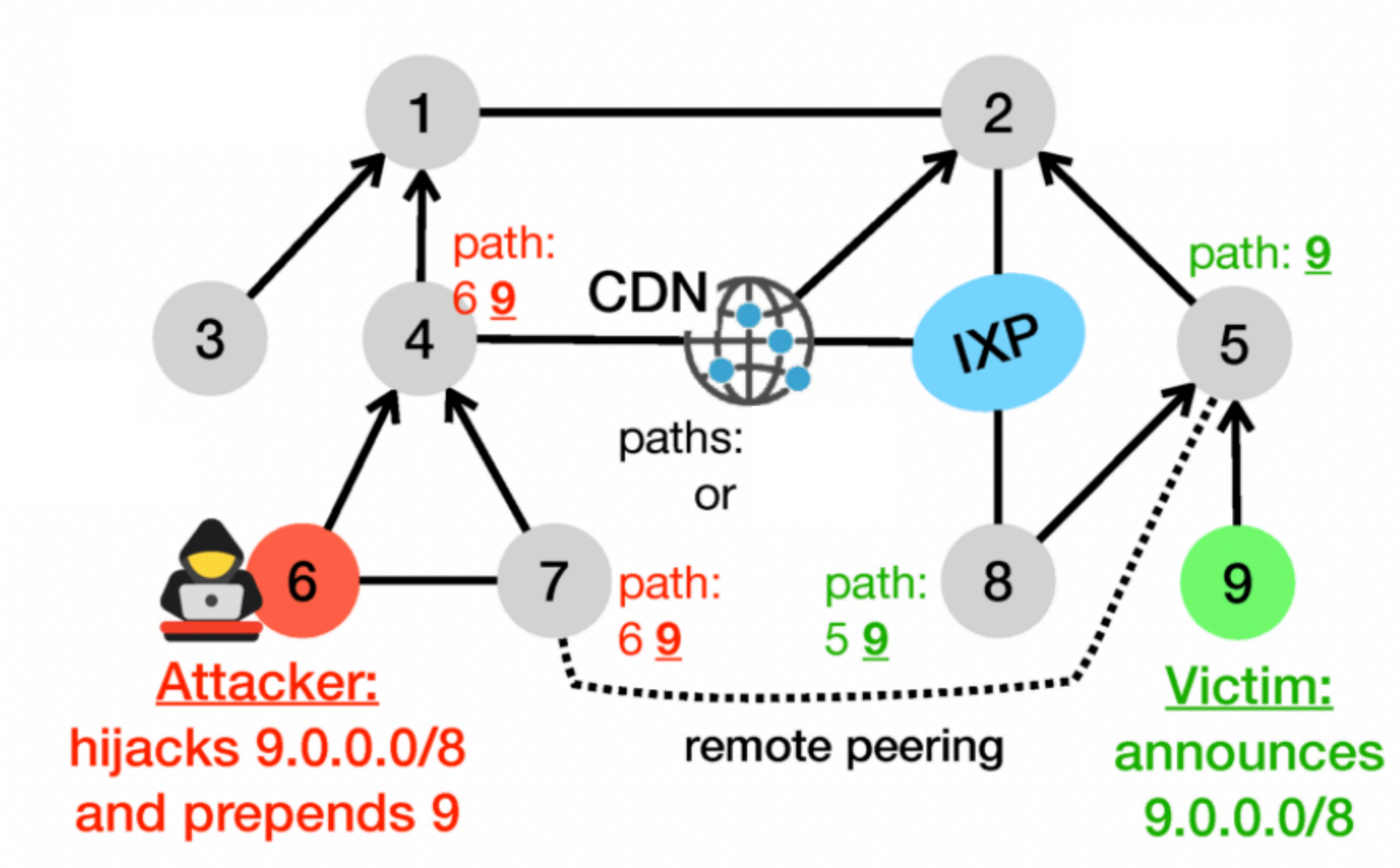
Forged-origin BGP hijack

- a BGP hijack attack
  - an attacker **announces forged AS paths** towards a victim prefix **by prepending the victim's origin AS number** to make them appear legitimate



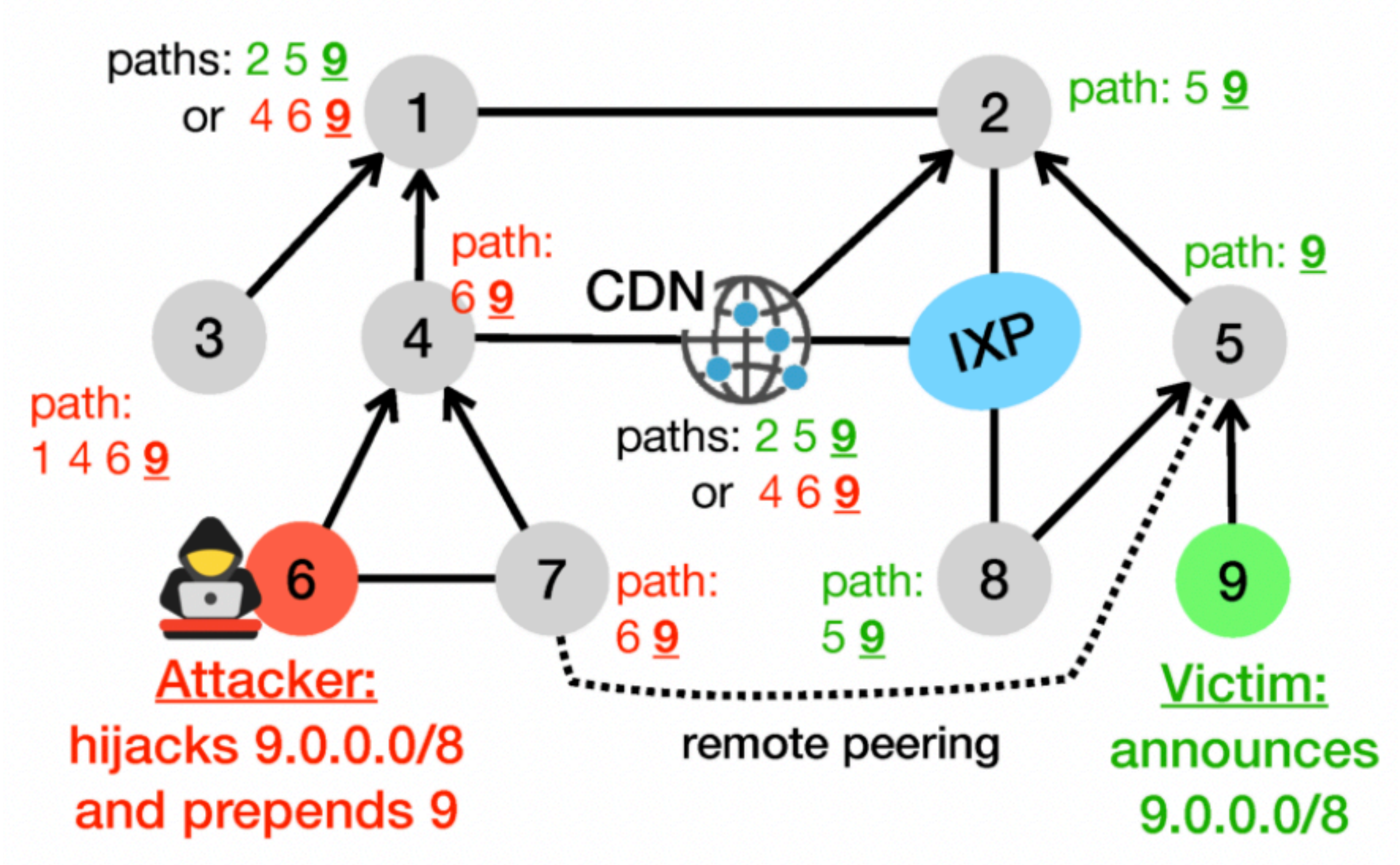
Forged-origin BGP hijack

- a BGP hijack attack
  - an attacker **announces forged AS paths** towards a victim prefix **by prepending the victim's origin AS number** to make them appear legitimate



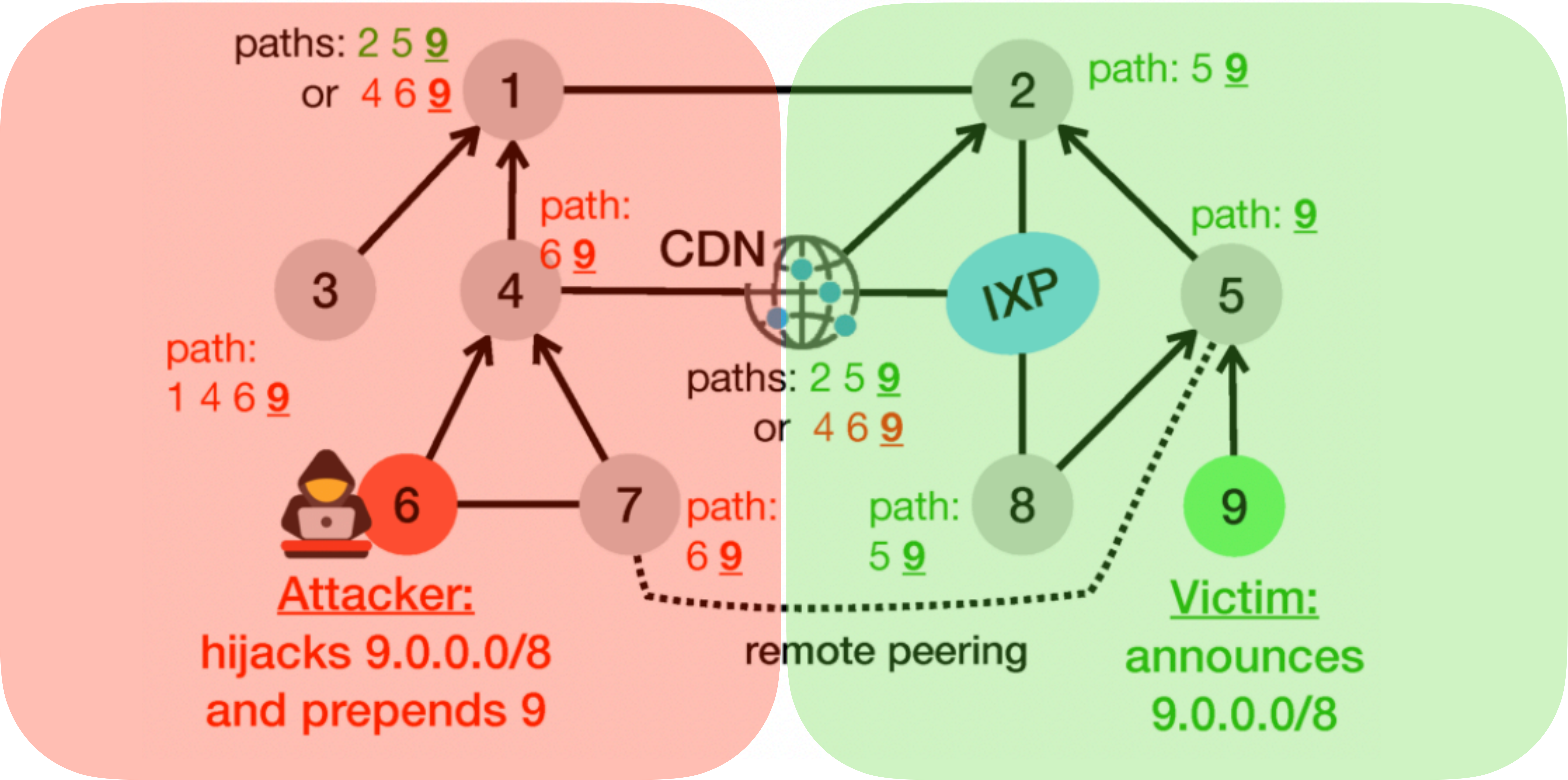
Forged-origin BGP hijack

- a BGP hijack attack
  - an attacker **announces forged AS paths** towards a victim prefix **by prepending the victim's origin AS number** to make them appear legitimate



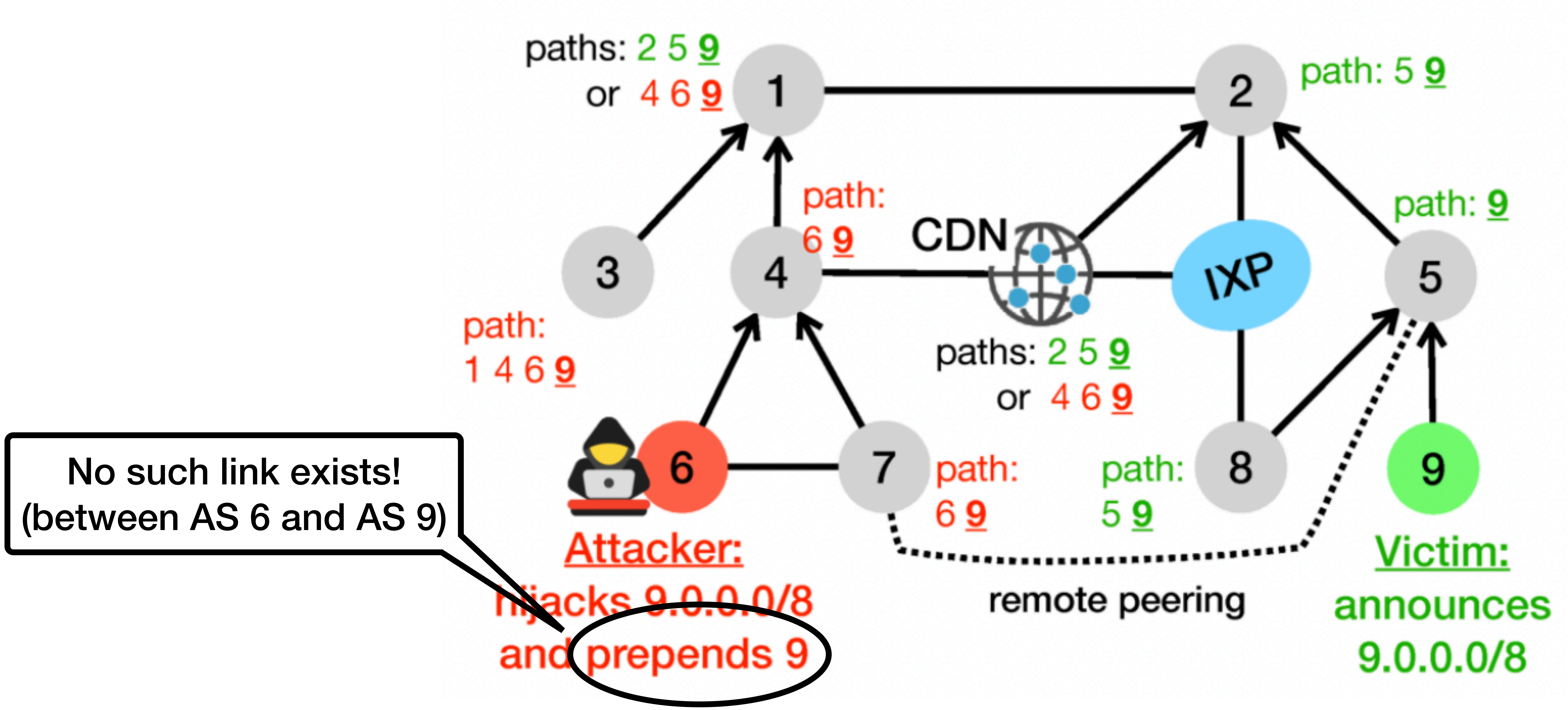
Forged-origin BGP hijack

- a BGP hijack attack
  - an attacker **announces forged AS paths** towards a victim prefix **by prepending the victim's origin AS number** to make them appear legitimate



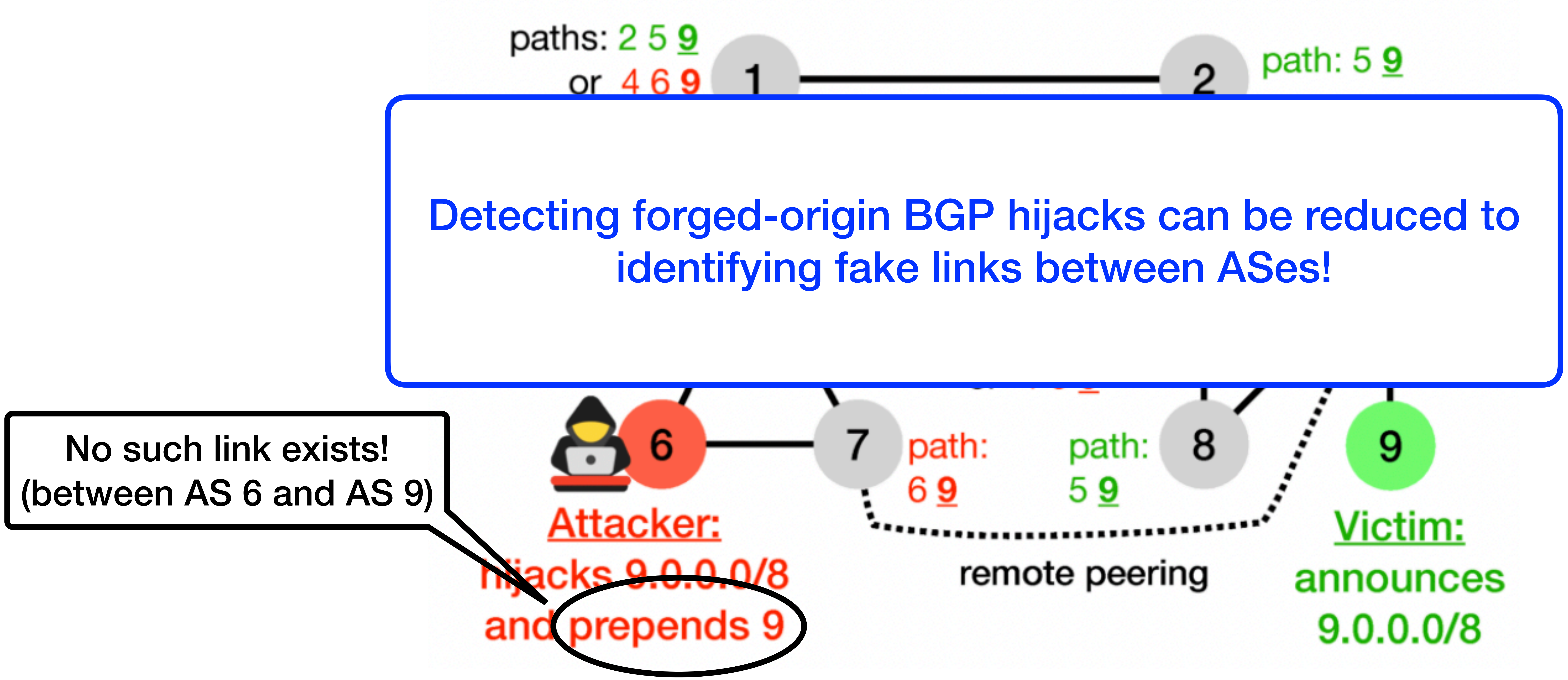
Forged-origin BGP hijack

- a BGP hijack attack
  - an attacker **announces forged AS paths** towards a victim prefix **by prepending the victim's origin AS number** to make them appear legitimate



Forged-origin BGP hijack

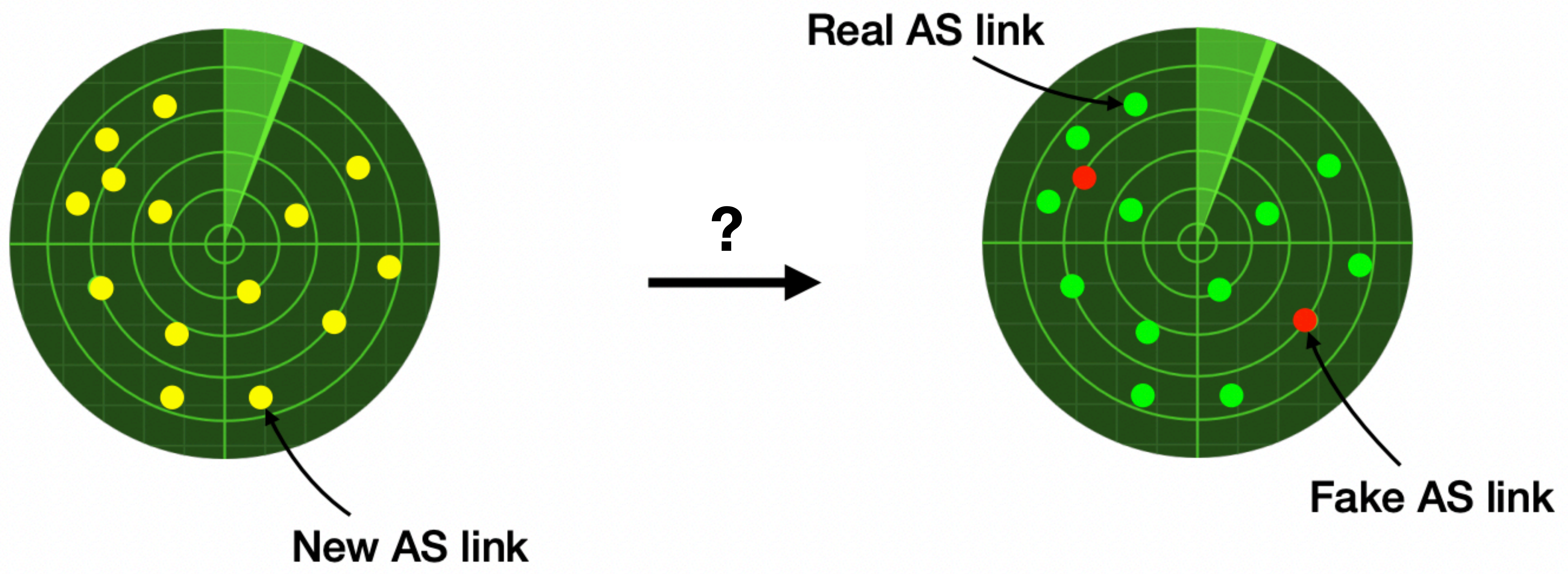
- a BGP hijack attack
  - an attacker **announces forged AS paths** towards a victim prefix **by prepending the victim's origin AS number** to make them appear legitimate



Detecting forged-origin BGP hijacks can be reduced to identifying fake links between ASes!

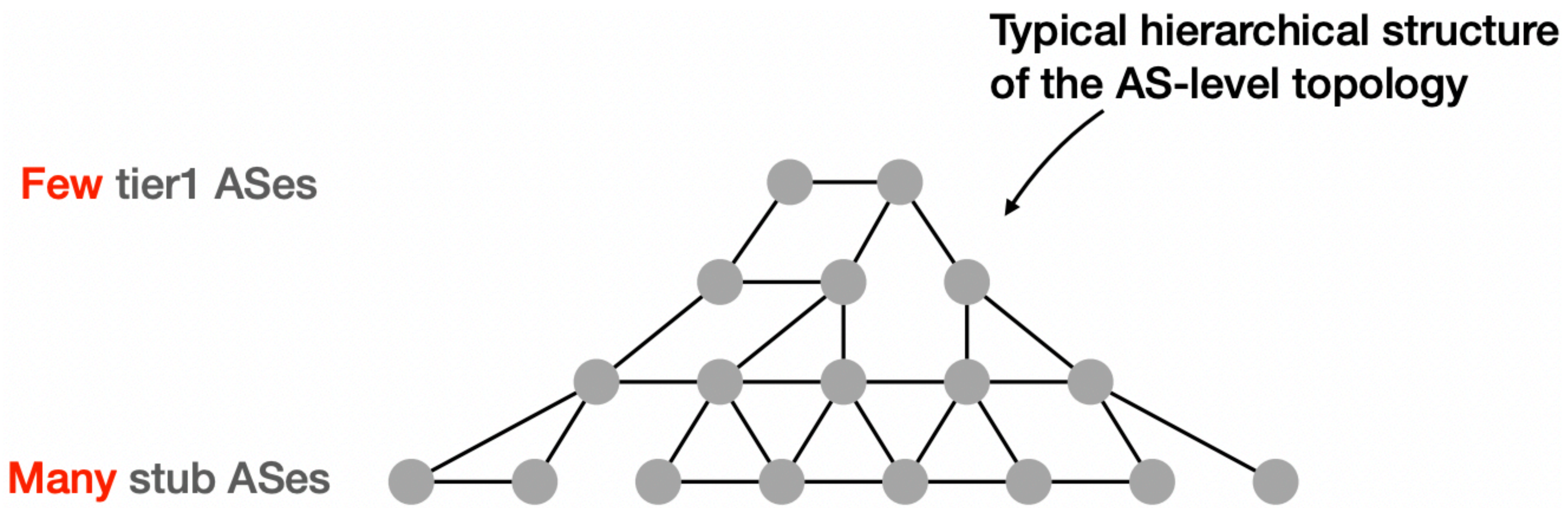
The challenge of identifying fake links

- There are many new AS links every day but no simple property that tells whether they are real or fake
  - 166 new AS links every day (median) and the vast majority are likely legitimate



Limitations of existing approaches

- Existing link prediction approaches [1, 2] does not perform well on detecting fake links
  - not suitable capture the characteristics of hierarchical AS topology



- ARTEMIS [3] can be used to detect forged-origin hijacks but it is self-operated
  - only capable of detecting hijacks targeting the AS deploying it

[1] Dimitrios Panteleimon Giakatos, Sofia Kostoglou, Pavlos Sermpezis, and Athena Vakali. Benchmarking Graph Neural Networks for Internet Routing Data, 2022.

[2] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. In *Advances in Neural Information Processing Systems*, 2018.

Requirements of a forged-origin detection system

1. must be **fast and accept real-time and historical queries**

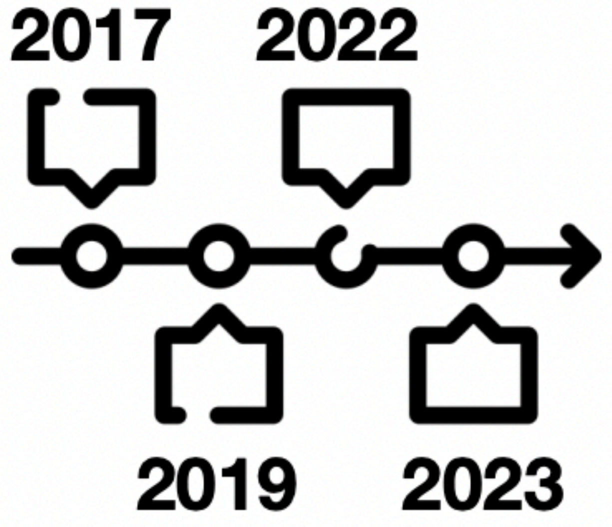
2. must be **accurate**, both for pinpointing actual hijacks and **avoiding triggering false alarms**

3. must be **robust against missing, inaccurate and polluted data**

4. must be accurate in **all attack and peering scenarios**

DFOH: A System to **D**etect **F**orged-**O**rigin BGP **H**ijacks

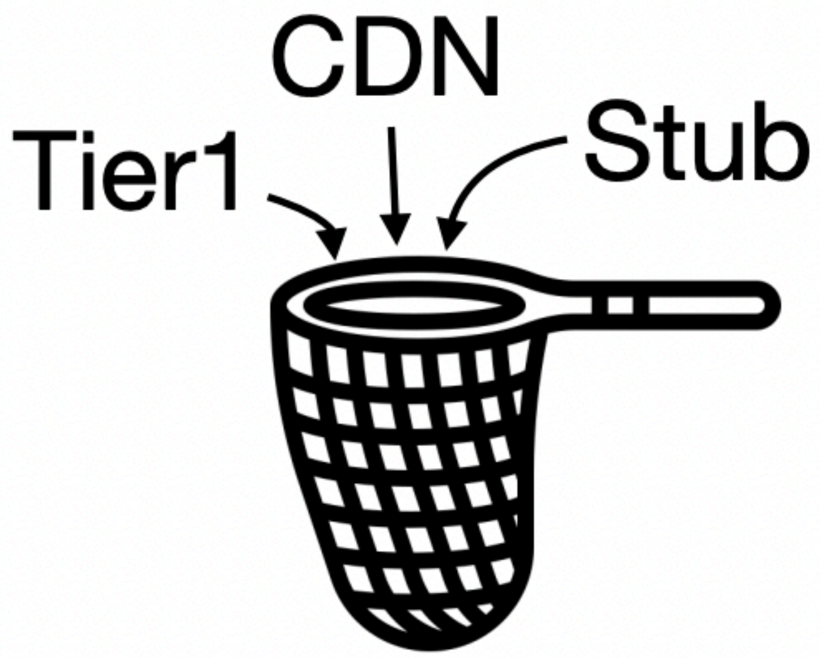
DFOH **runs in a commodity server**



DFOH **detects past hijacks**

DFOH **detects hijacks on the whole Internet**

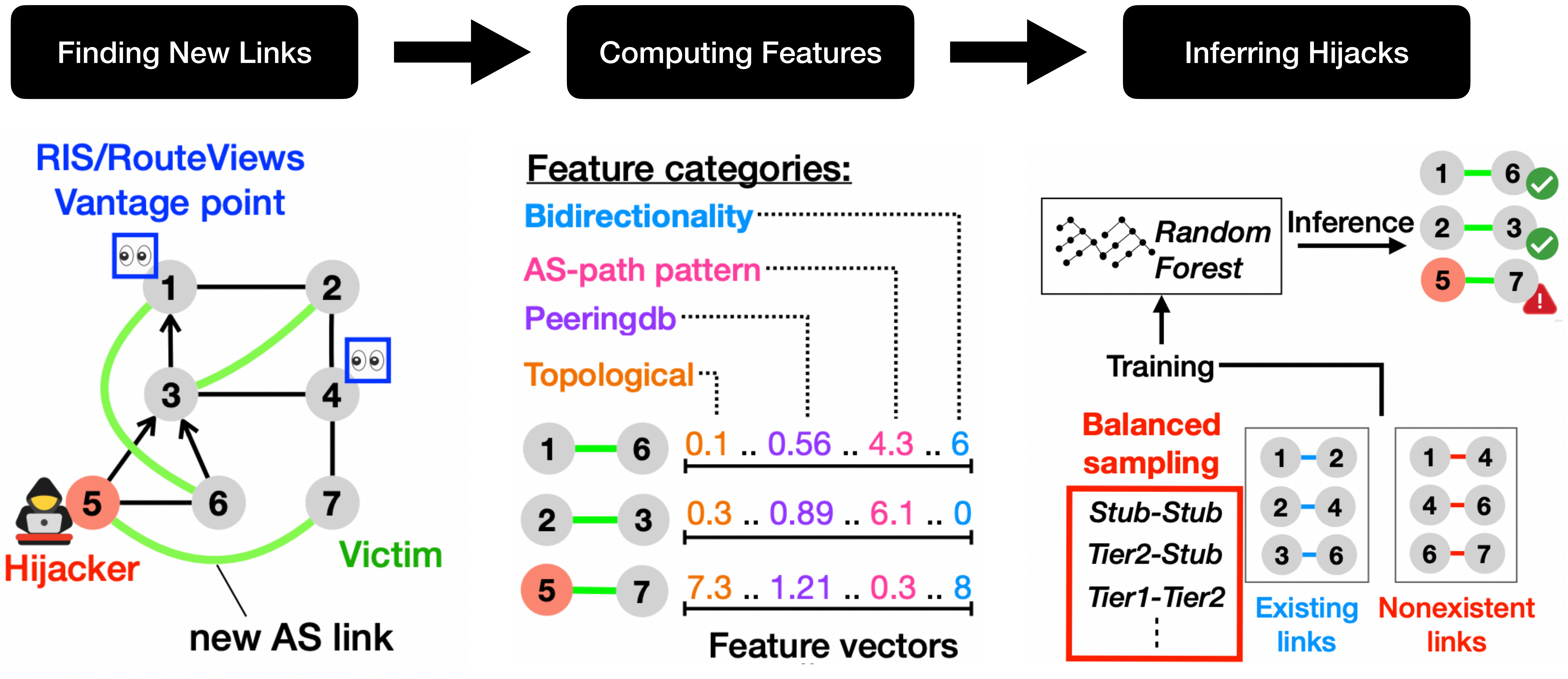
DFOH **provides near-real-time detection**



DFOH **is accurate in every attack scenario**

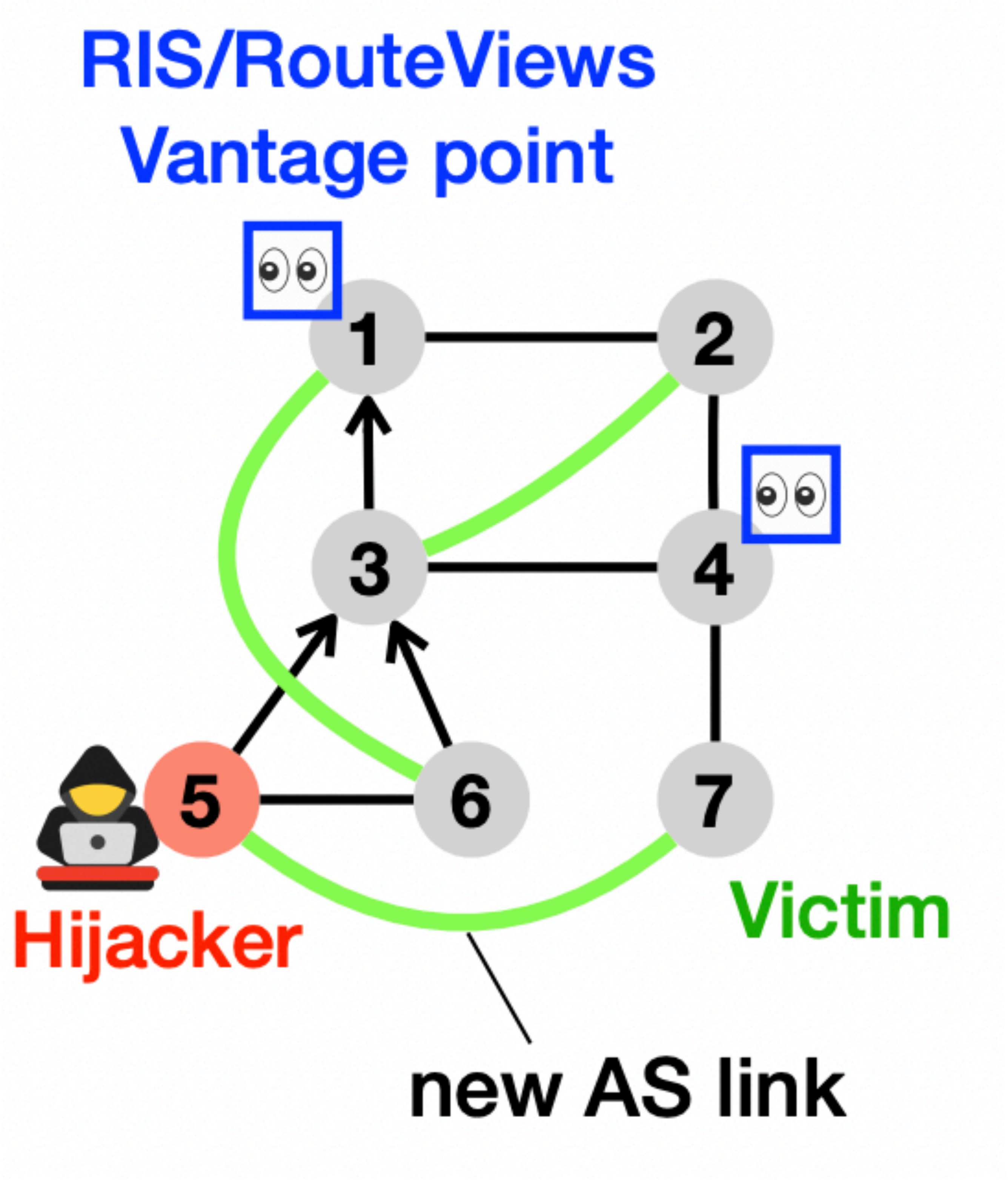
DFOH **is robust against adversarial inputs**

DFOH inference pipeline



Finding New Links

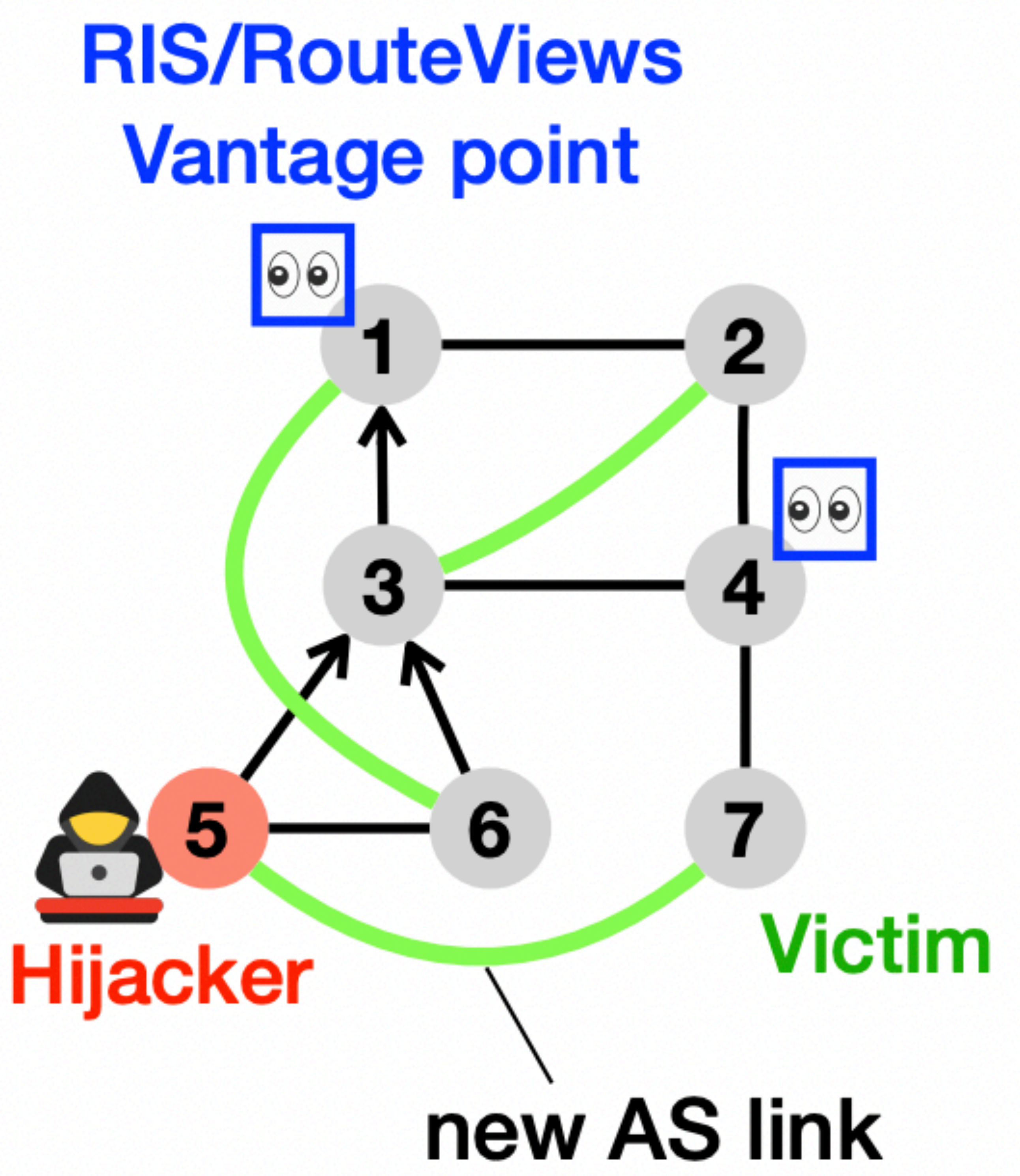
Finding New Links → Computing Features → Inferring Hijacks



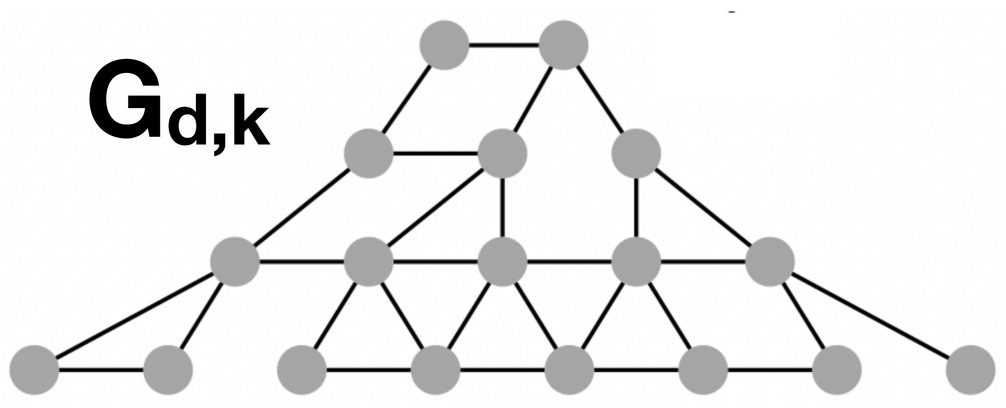
date	d-k	...	d	d+1
BGP update				
RIB				
CAIDA				

Finding New Links

Finding New Links → Computing Features → Inferring Hijacks



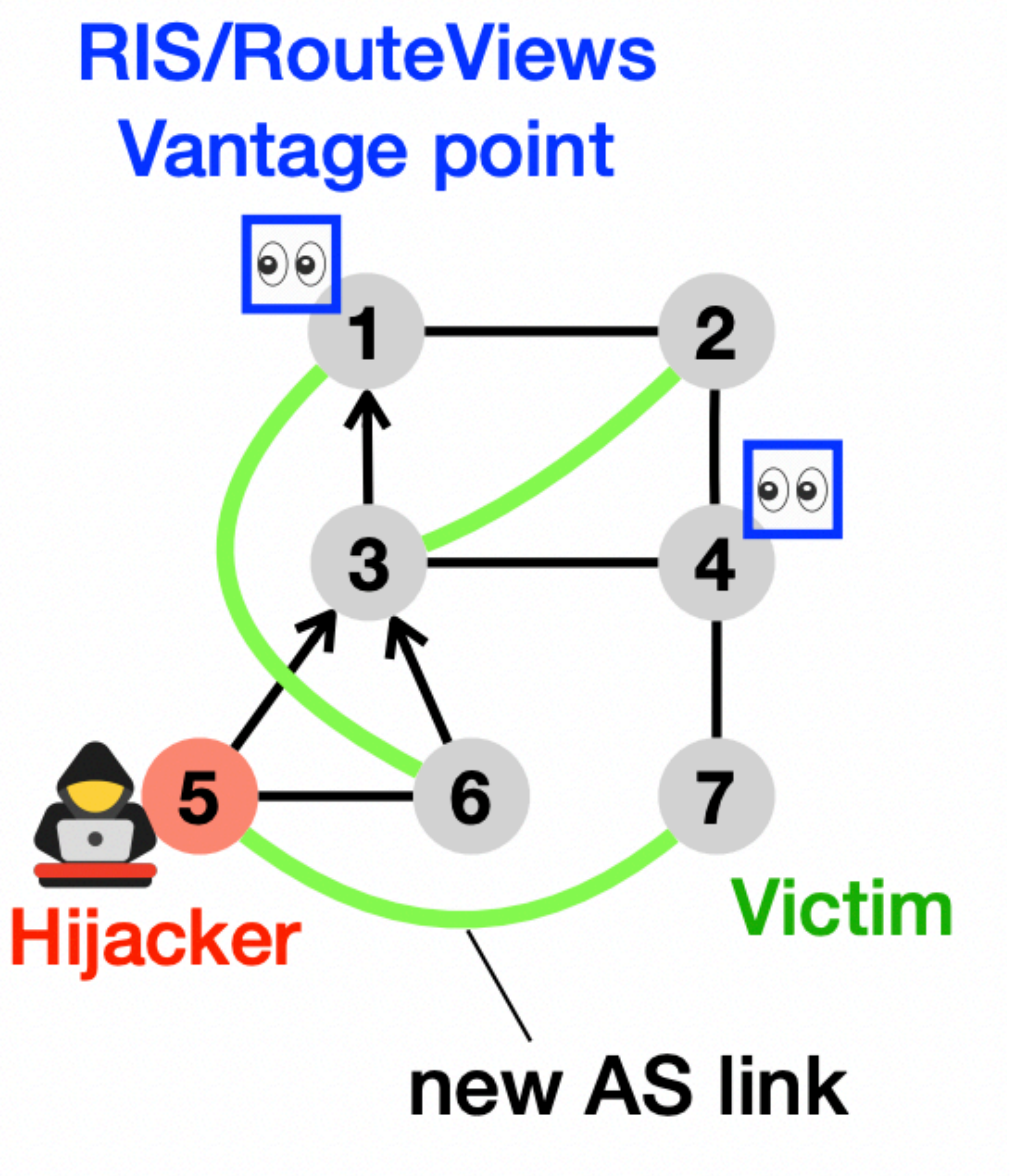
date	d-k	...	d	d+1
BGP update				
RIB				
CAIDA				



AS topology graph

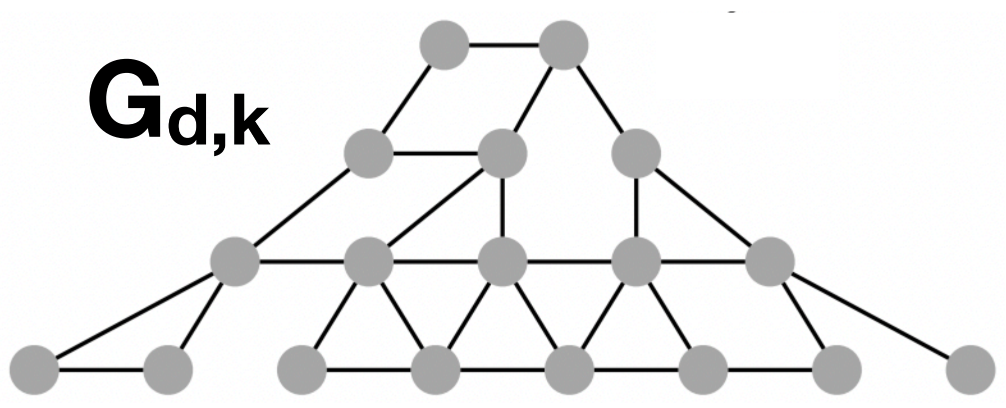
Finding New Links

Finding New Links → Computing Features → Inferring Hijacks



date	d-k	…	d	d+1
BGP update				
RIB				
CAIDA				

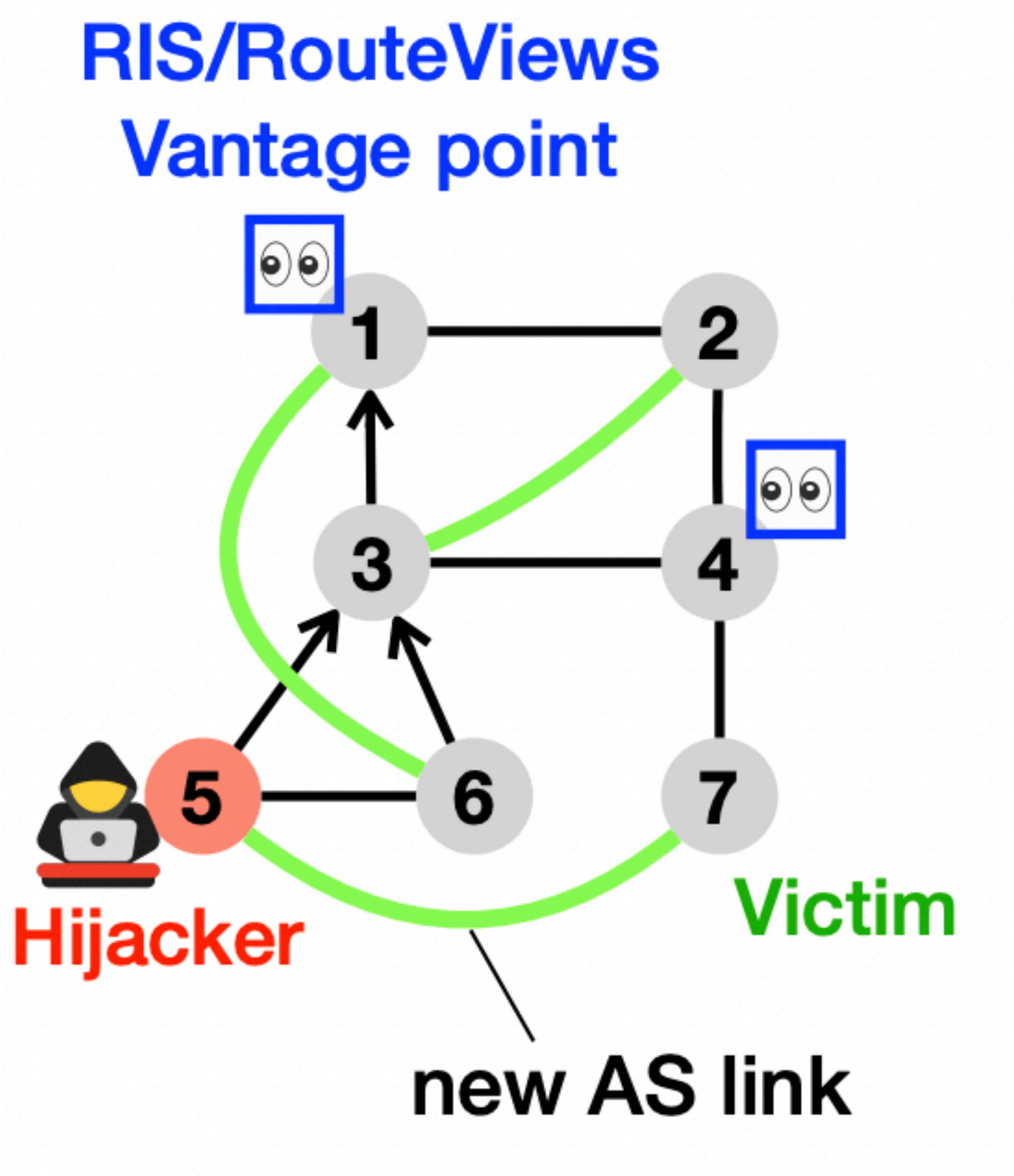
AS links



AS topology graph

DFOH inference pipeline

Finding New Links → Computing Features → Inferring Hijacks



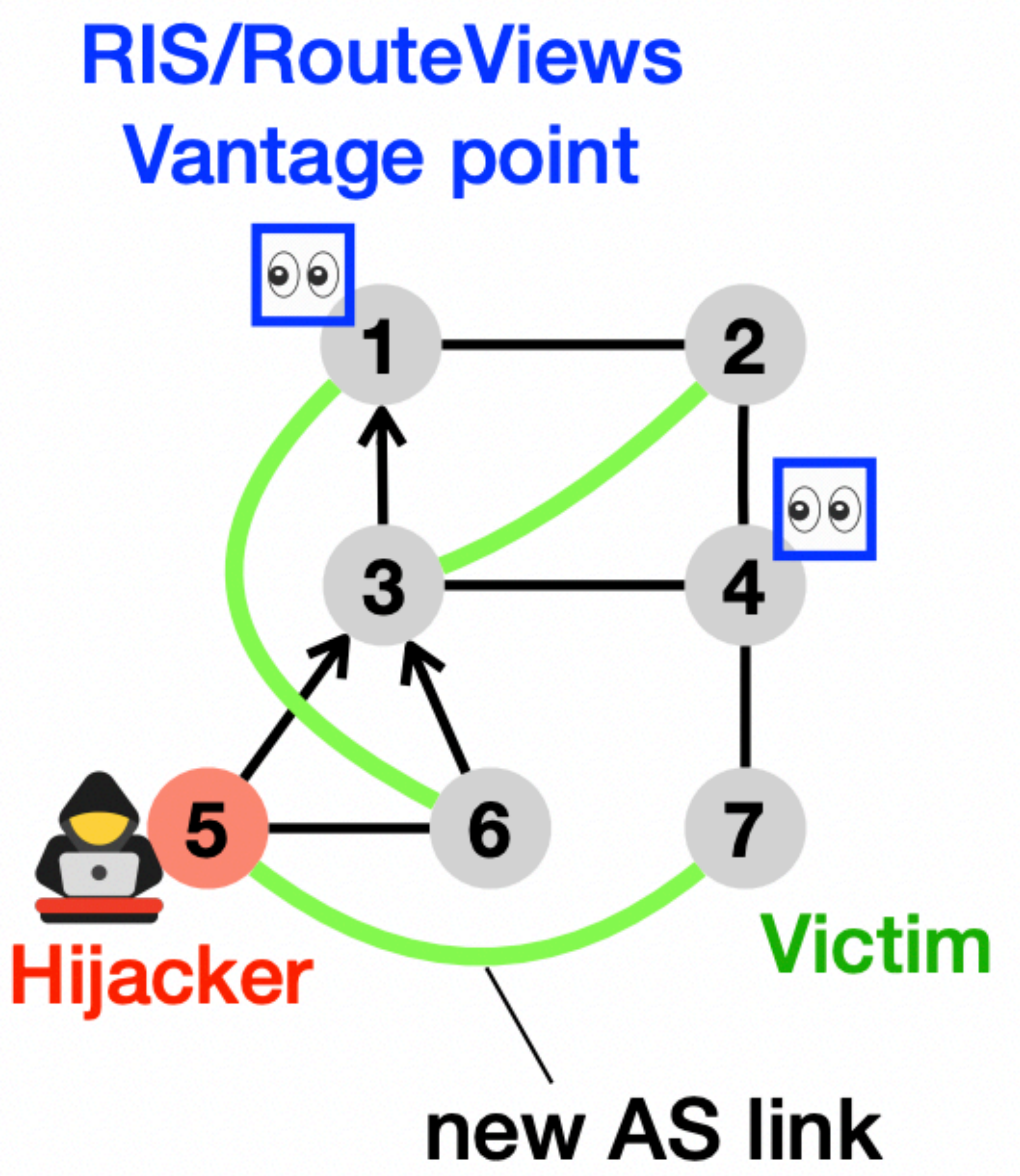
Builds an AS topology graph $G_{d,k}$ using the AS paths from
- BGP updates collected from 287 BGP vantage points (from the day ***d-k*** to the day ***d***)
- RIB of 287 BGP vantage points (at the day ***d***)
- CAIDA datasets (at the day ***d***)

Collects the BGP updates from the 287 BGP vantage points observed at the day ***d+1***

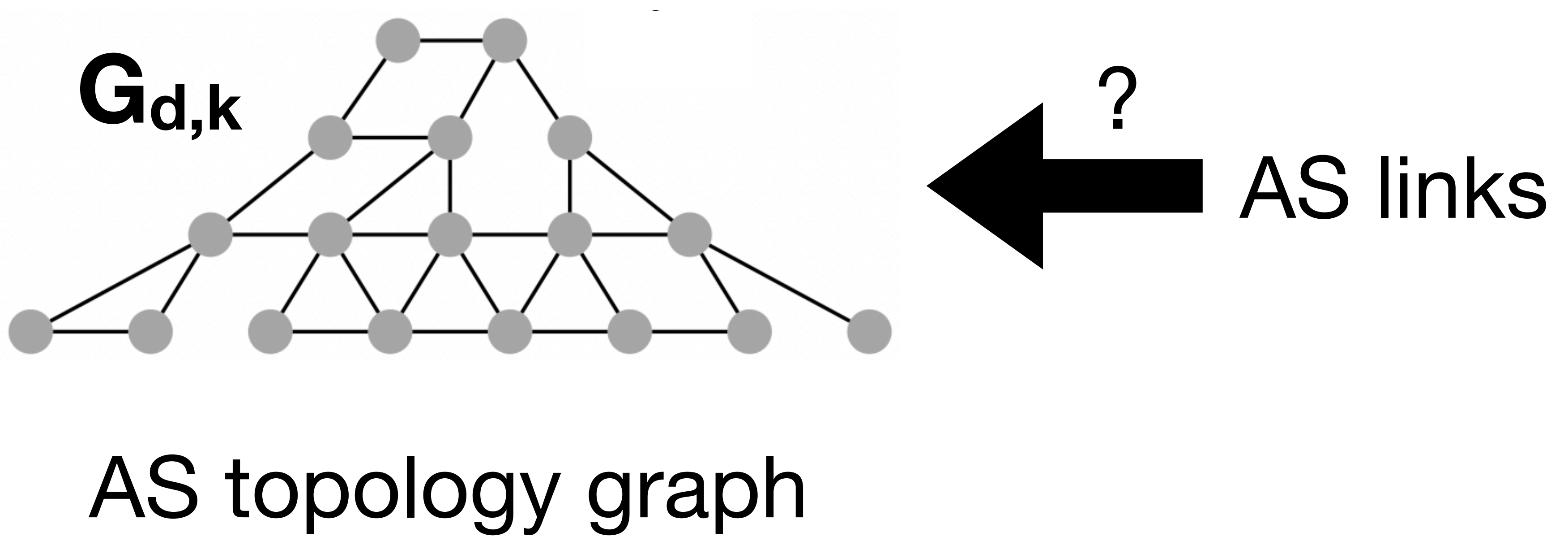
extracts AS paths and checks whether an AS link in the AS paths in the AS topology graph $G_{d,k}$

Finding New Links

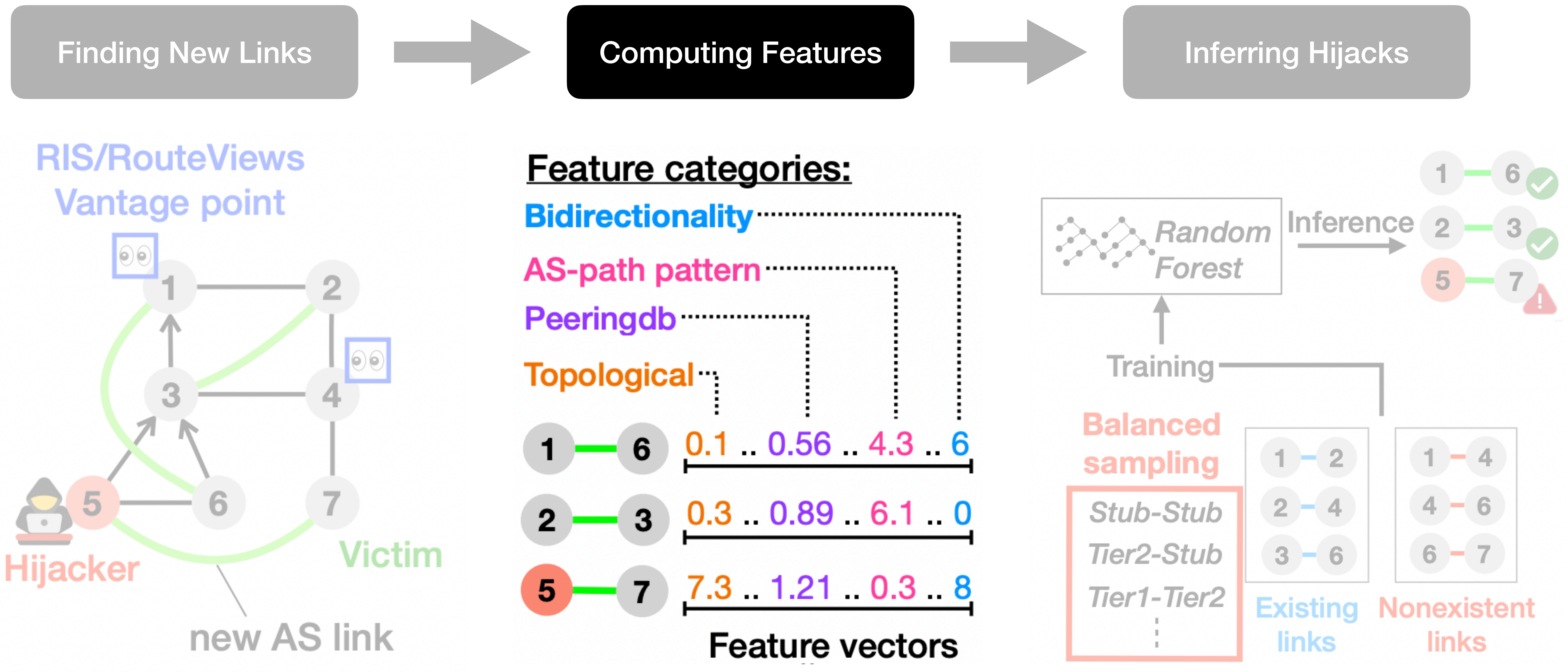
Finding New Links → Computing Features → Inferring Hijacks



date	d-k	…	d	d+1
BGP update				
RIB				
CAIDA				

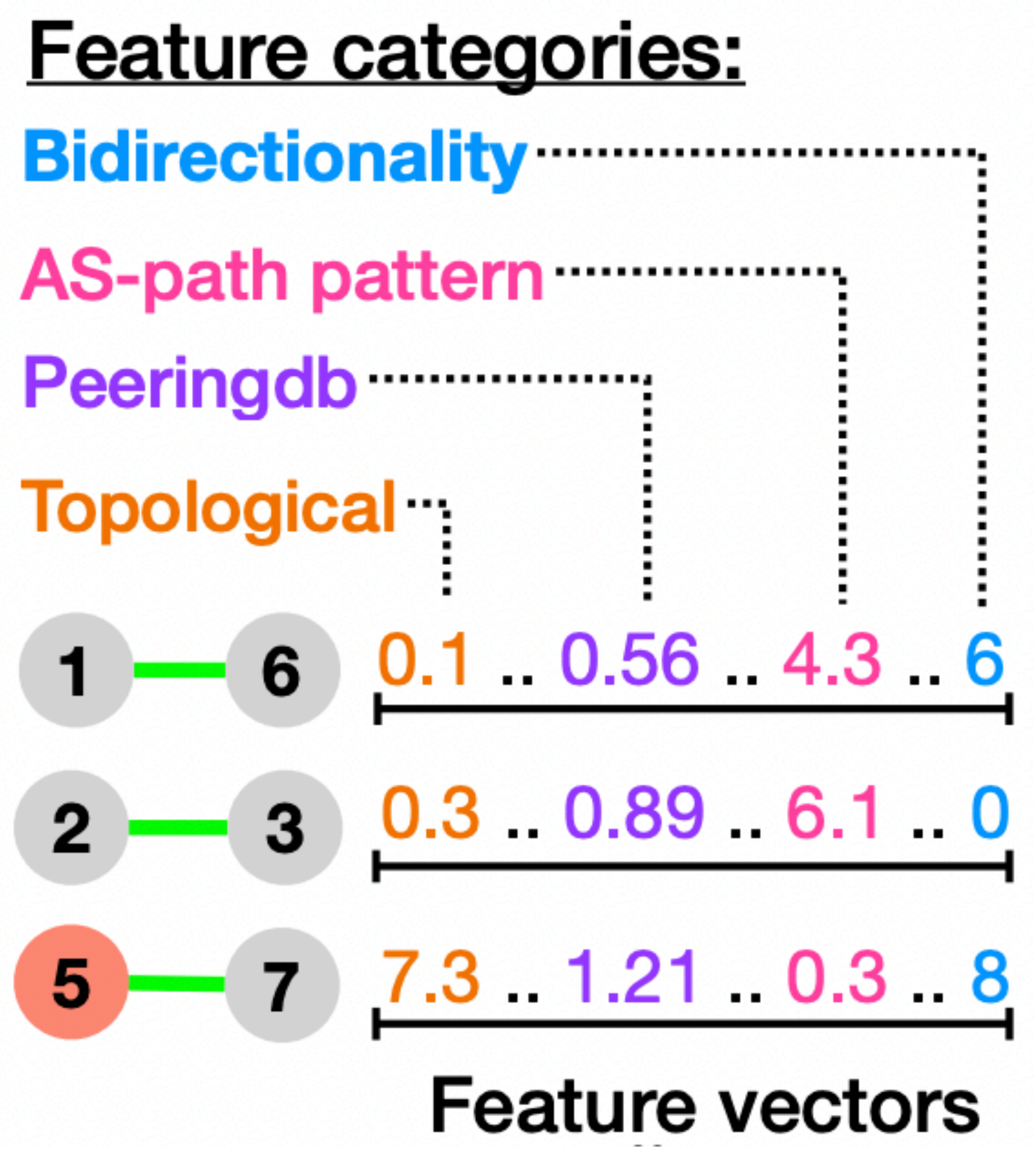


Computing Features



Computing Features

Finding New Links → **Computing Features** → Inferring Hijacks

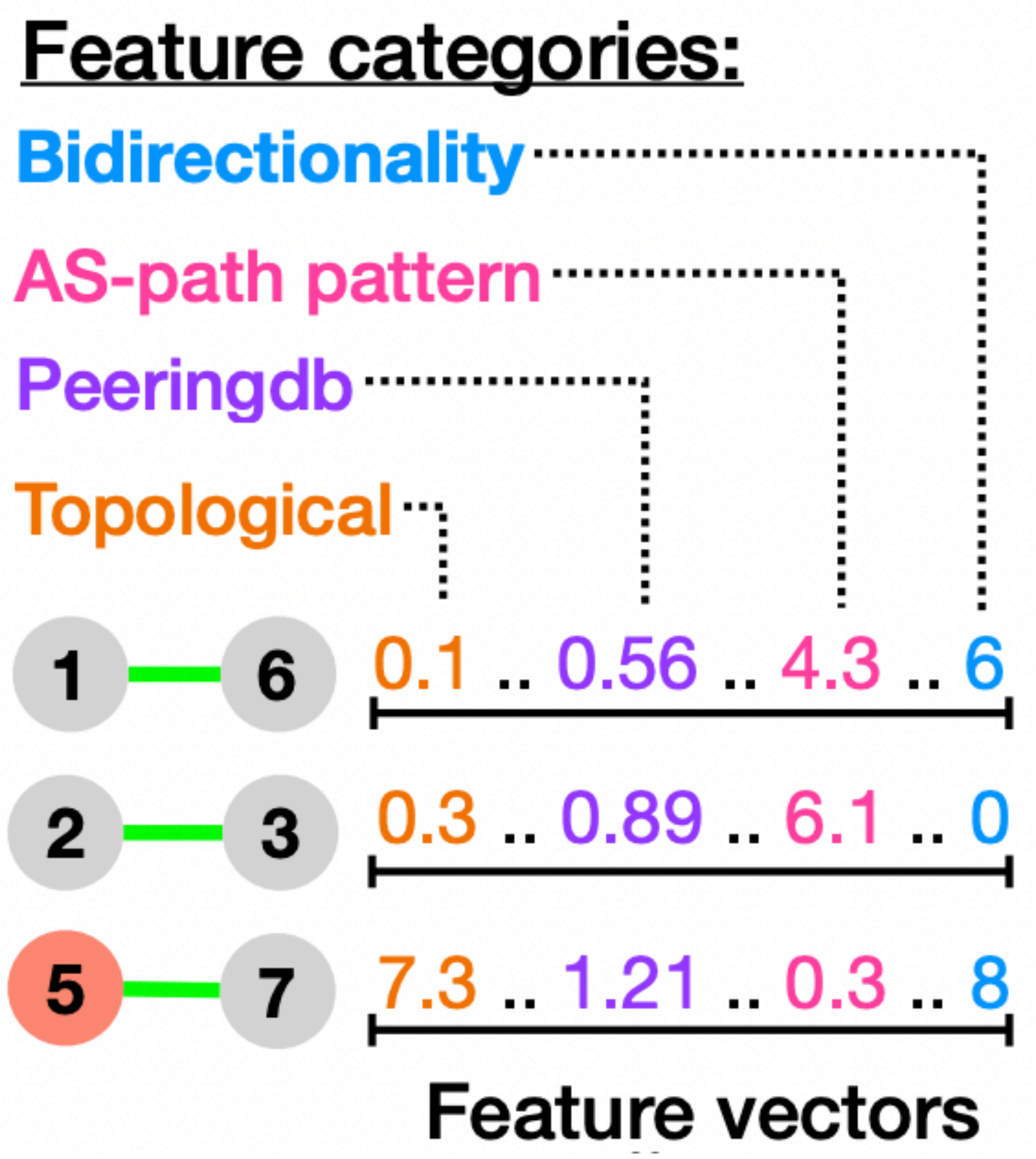


1. Topological features

- quantify the change induced by a new link on the AS topology

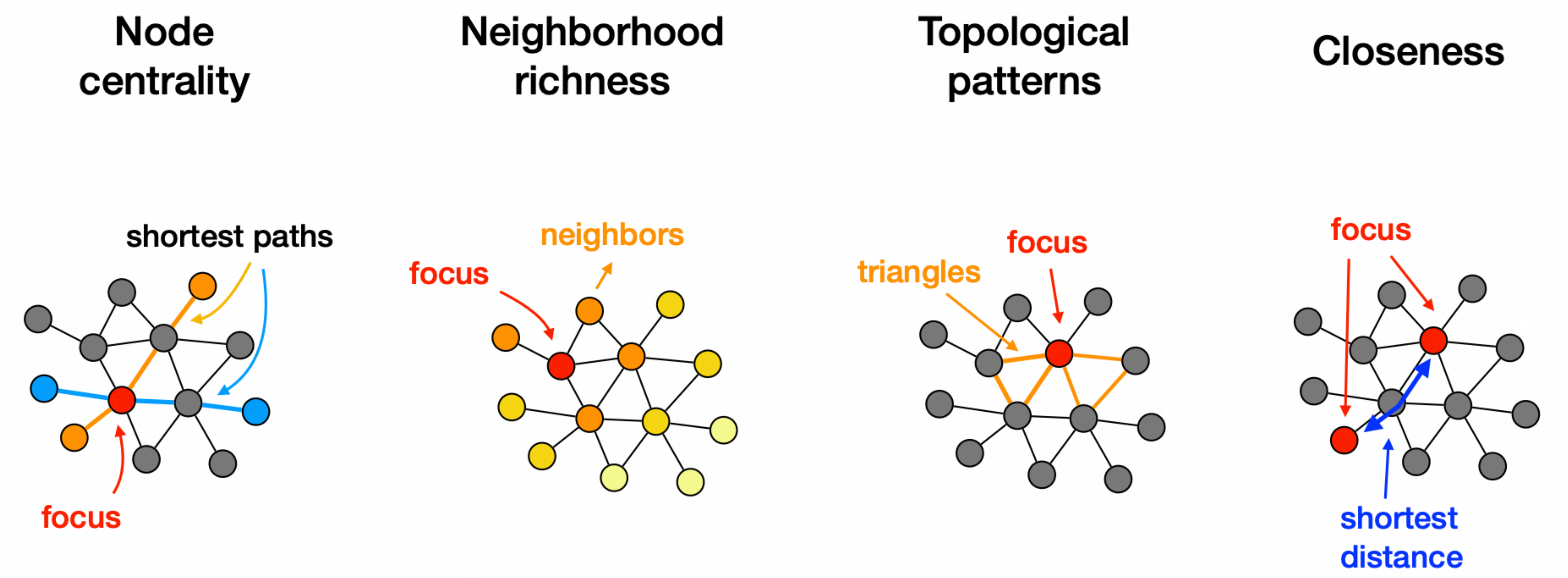
Computing Features

Finding New Links → **Computing Features** → Inferring Hijacks



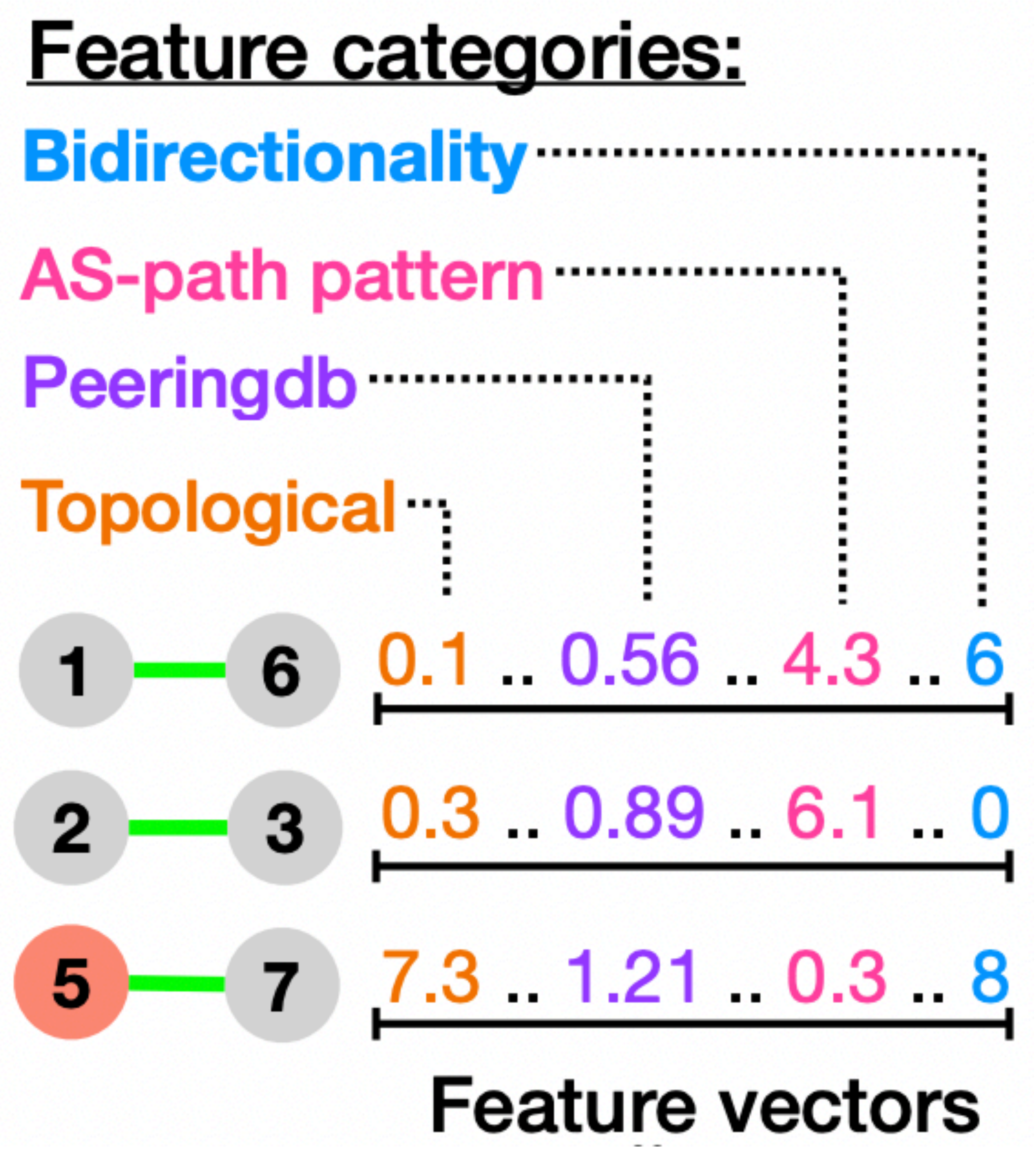
1. Topological features

- quantify the change induced by a new link on the AS topology
- a total of 11 topological features that can be divided into four categories



Computing Features

Finding New Links → **Computing Features** → Inferring Hijacks



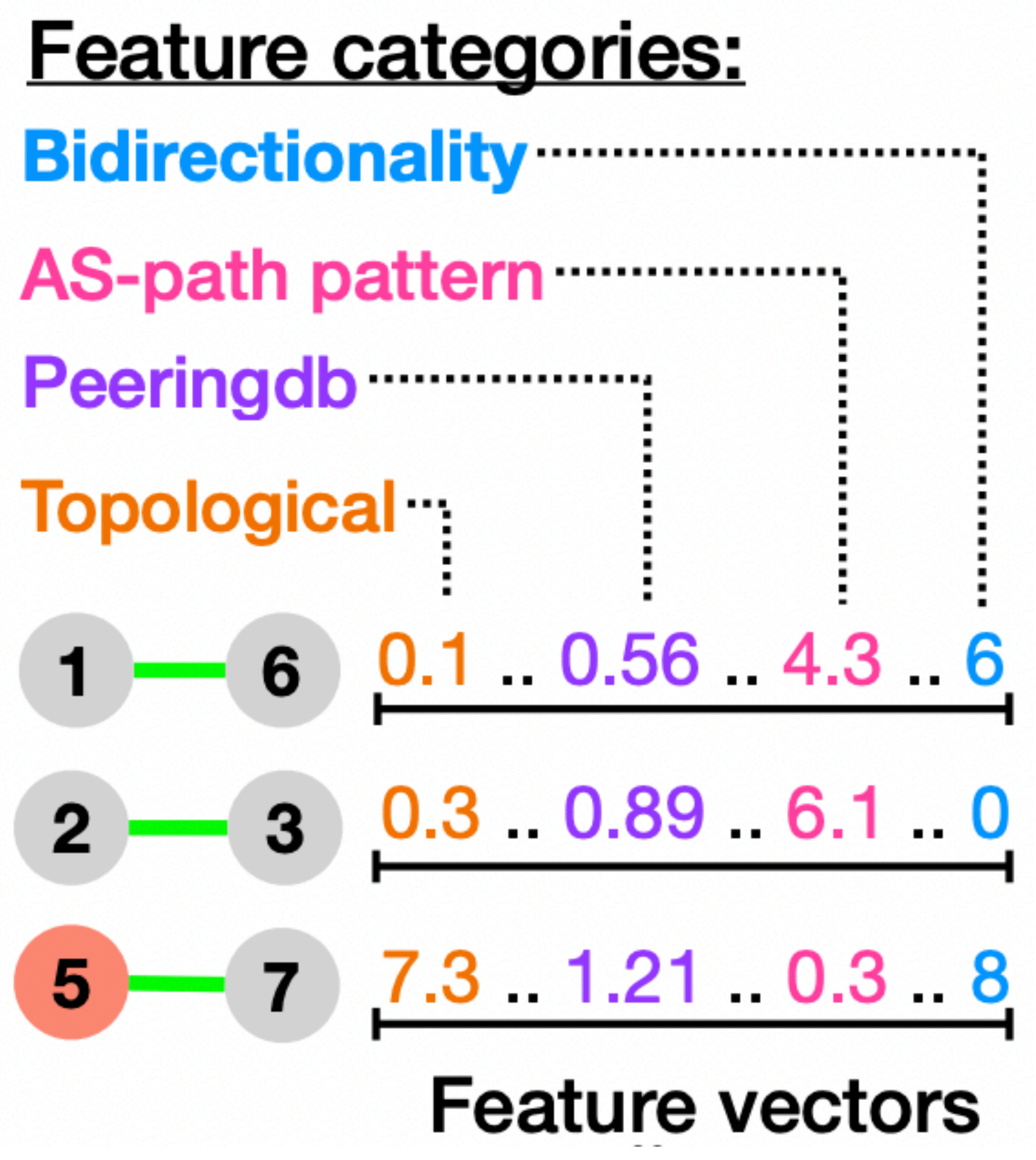
2. Peeringdb Features

- use public peering information to identify peering characteristics
- Intuitively, two ASes that exhibit similar peering characteristics have a higher chance to peer

Index	Description
1	The countries where *ASX*'s neighbors are registered
2	The IXPs to which *ASX*'s neighbors are connected to
3	The facilities to which *ASX*'s neighbors are present
4	The cities of the facilities to which *ASX*'s neighbors are present
5	The countries of the facilities to which *ASX*'s neighbors are present

Computing Features

Finding New Links → **Computing Features** → Inferring Hijacks



2. Peeringdb Features

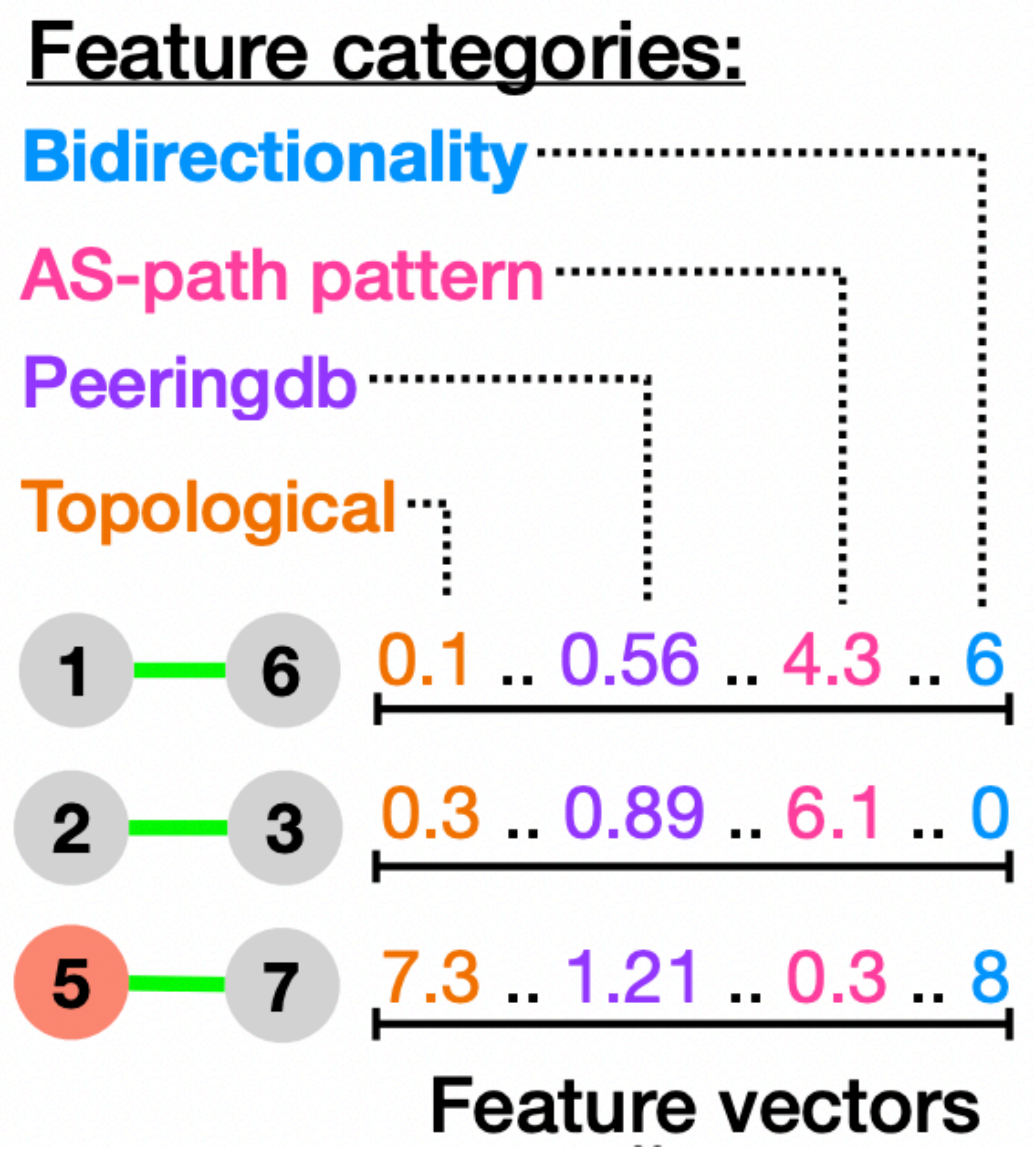
- use public peering information to identify peering characteristics

- Intuitively, two ASes that exhibit similar peering characteristics have a higher chance to peer

Index	Description
1	The countries where *ASX*'s neighbors are registered
2	The IXPs to which *ASX*'s neighbors are connected to
3	The facilities to which *ASX*'s neighbors are present
4	The cities of the facilities to which *ASX*'s neighbors are present
5	The countries of the facilities to which *ASX*'s neighbors are present

Computing Features

Finding New Links → **Computing Features** → Inferring Hijacks



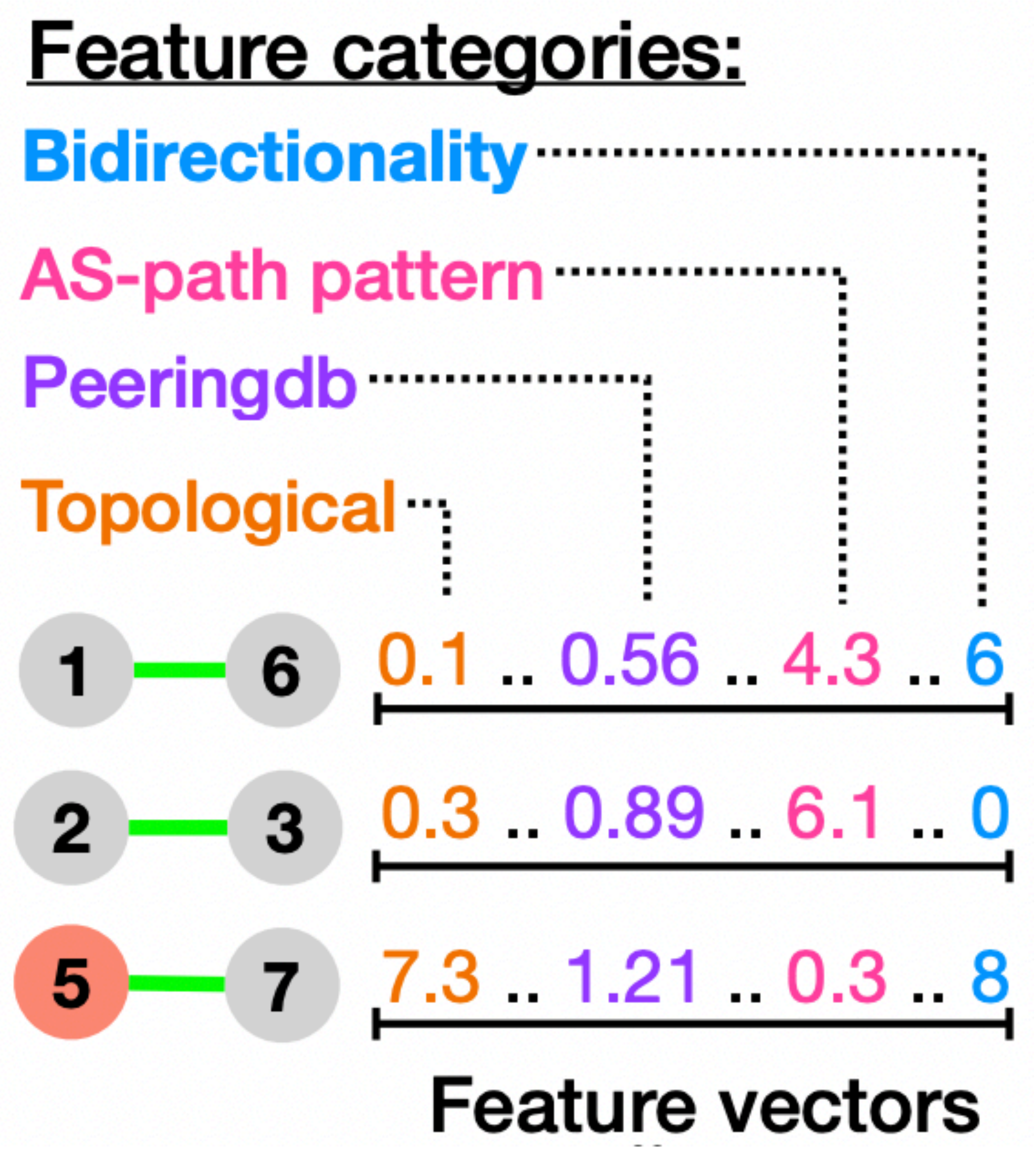
2. Peeringdb Features

- use public peering information to identify peering characteristics
- Intuitively, two ASes that exhibit similar peering characteristics have a higher chance to peer

Index	Description
1	The countries where *ASX*'s neighbors are registered
2	The IXPs to which *ASX*'s neighbors are connected to
3	The facilities to which *ASX*'s neighbors are present
4	The cities of the facilities to which *ASX*'s neighbors are present
5	The countries of the facilities to which *ASX*'s neighbors are present

Computing Features

Finding New Links → **Computing Features** → Inferring Hijacks



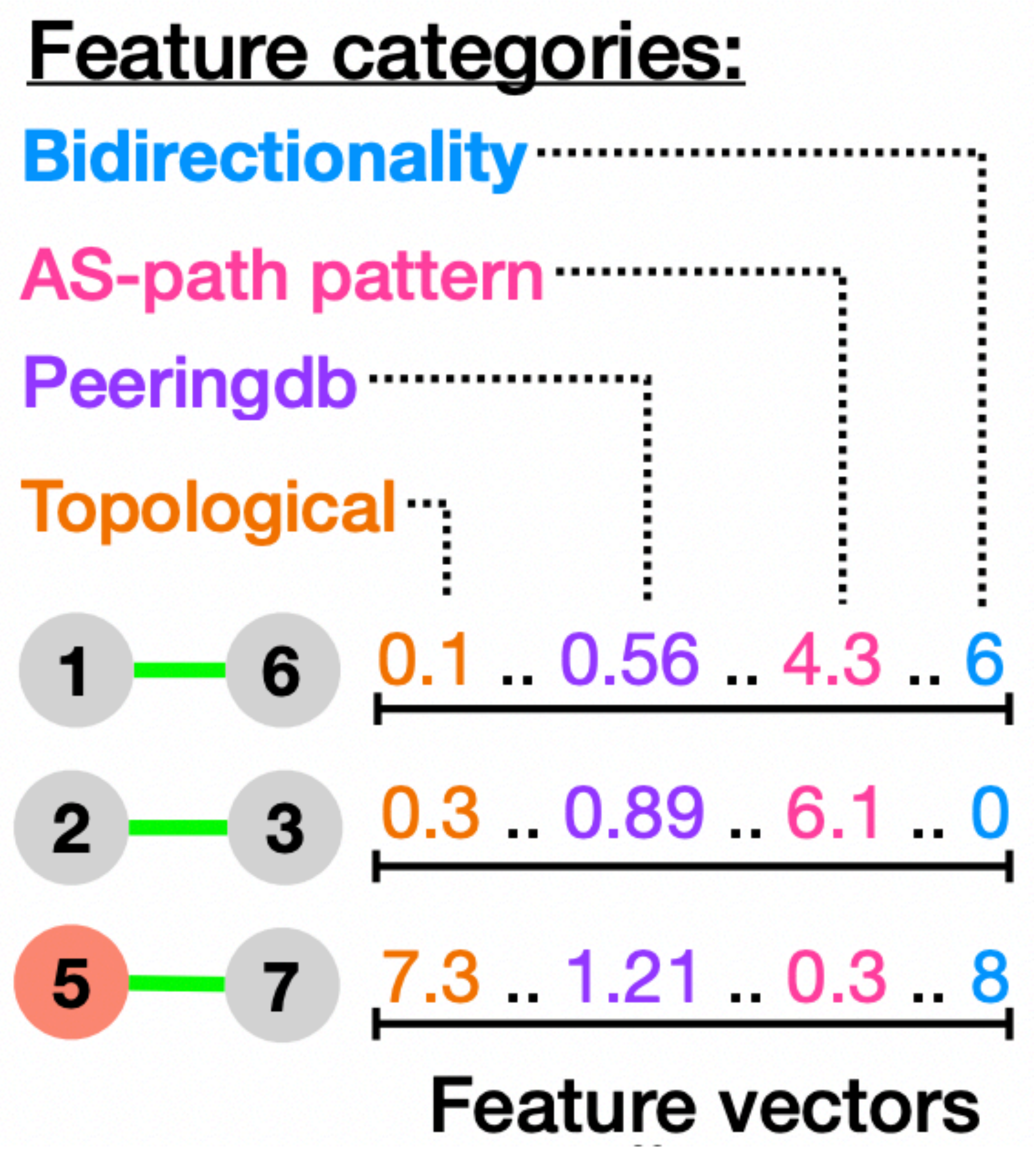
2. Peeringdb Features

- use public peering information to identify peering characteristics
- Intuitively, two ASes that exhibit similar peering characteristics have a higher chance to peer

Index	Description
1	The countries where *ASX*'s neighbors are registered
2	The IXPs to which *ASX*'s neighbors are connected to
3	The facilities to which *ASX*'s neighbors are present
4	The cities of the facilities to which *ASX*'s neighbors are present
5	The countries of the facilities to which *ASX*'s neighbors are present

Computing Features

Finding New Links → **Computing Features** → Inferring Hijacks



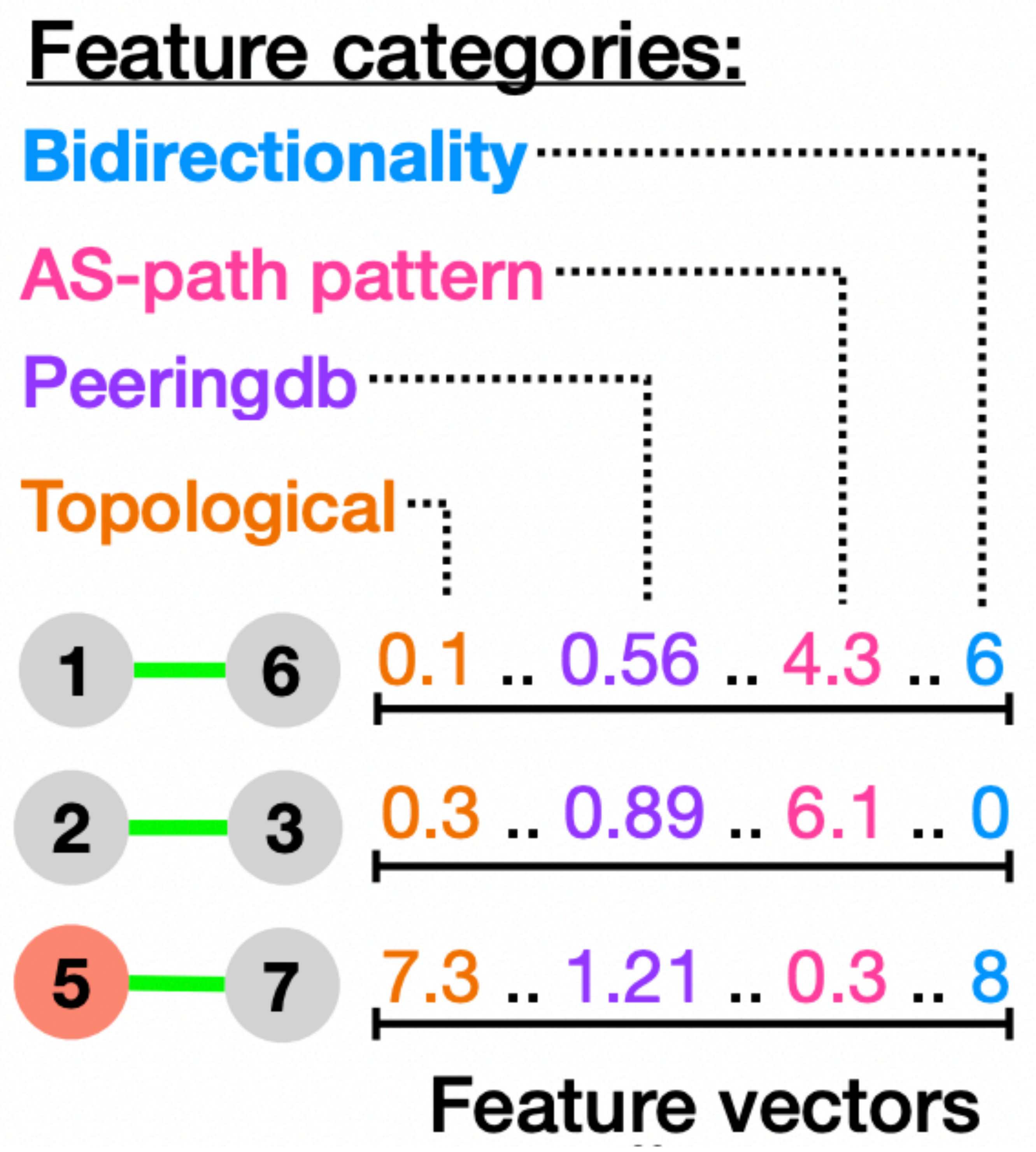
compares the peering information of the neighbors
→ protect against adversarial input
& mitigate missing peering information

- Intuitively, two ASes that exhibit similar peering characteristics have a higher chance to peer

Index	Description
1	The countries where *ASX*'s neighbors are registered
2	The IXPs to which *ASX*'s neighbors are connected to
3	The facilities to which *ASX*'s neighbors are present
4	The cities of the facilities to which *ASX*'s neighbors are present
5	The countries of the facilities to which *ASX*'s neighbors are present

Computing Features

Finding New Links → **Computing Features** → Inferring Hijacks

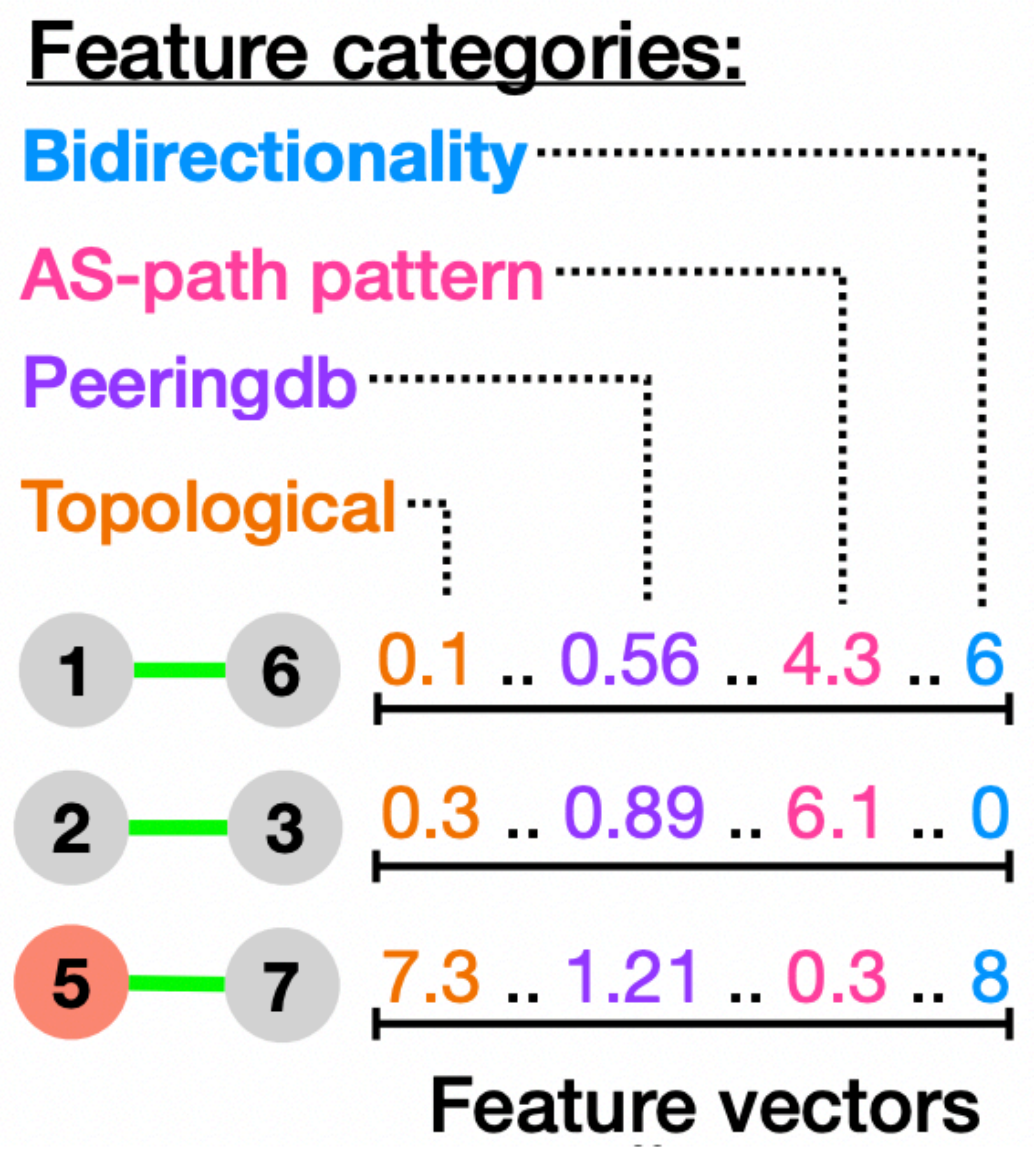


3. AS-path patterns

- examine the AS paths that include the new link and identifies suspicious sequence of ASes

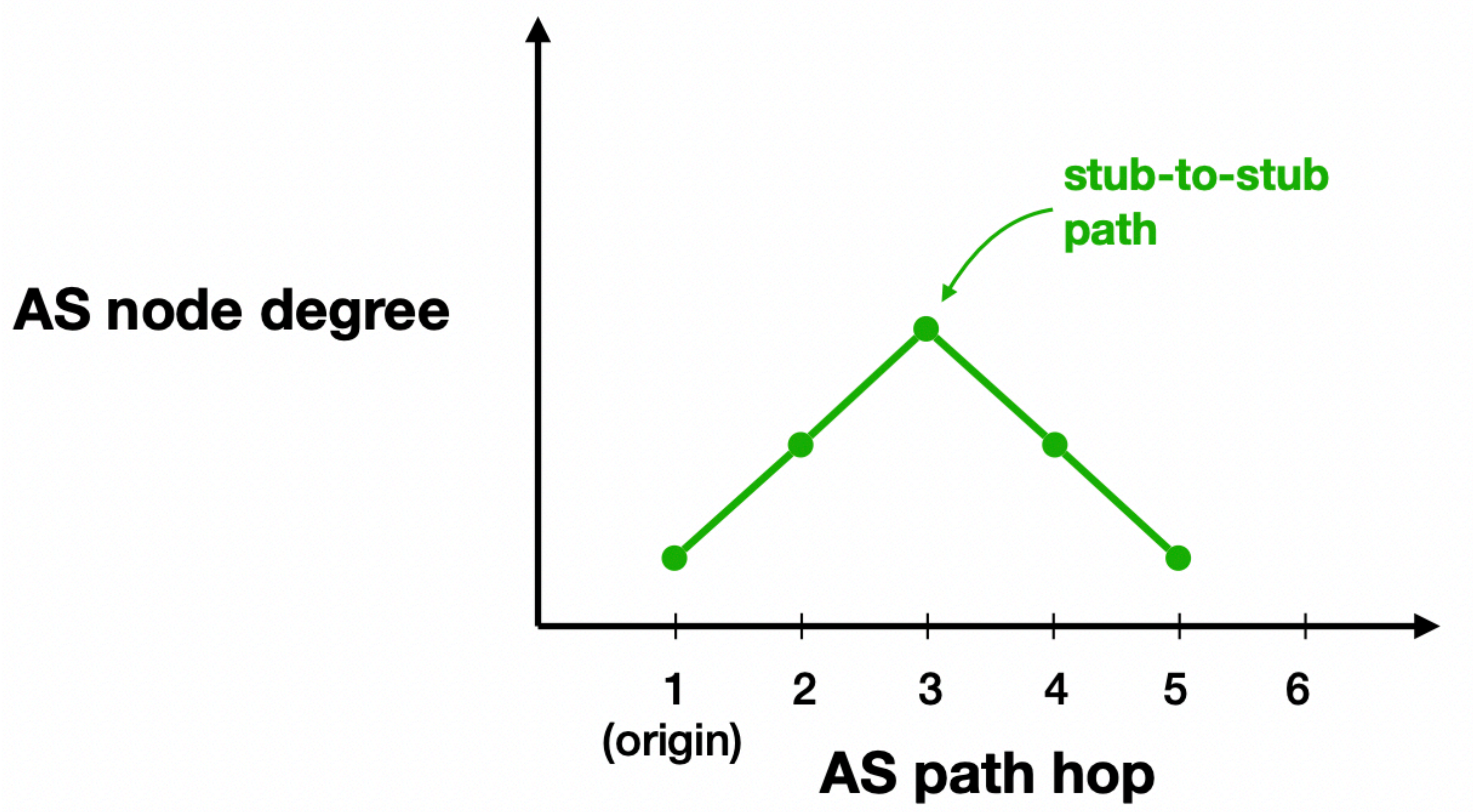
Computing Features

Finding New Links → **Computing Features** → Inferring Hijacks



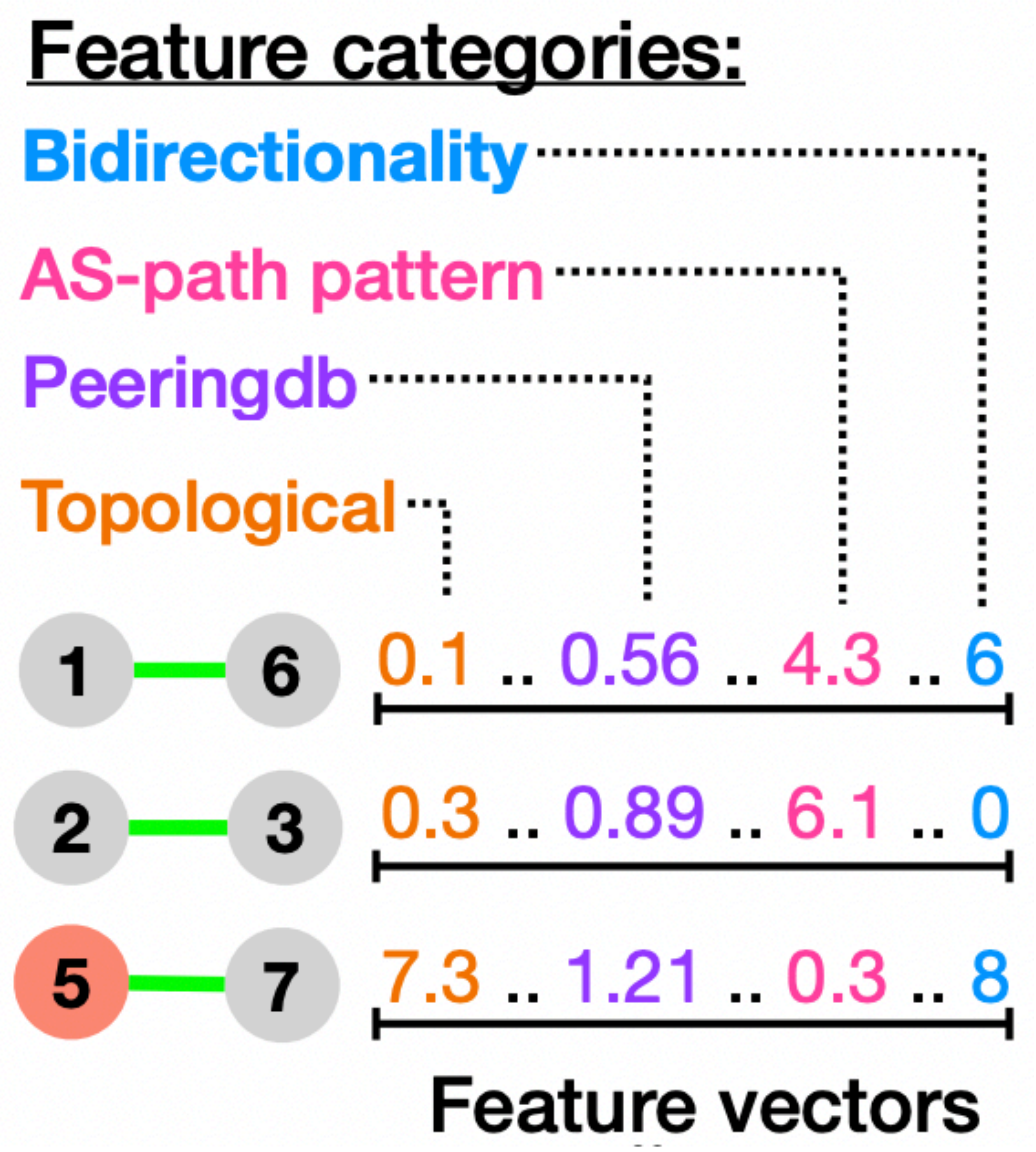
3. AS-path patterns

- examine the AS paths that include the new link to identify suspicious sequence of ASes



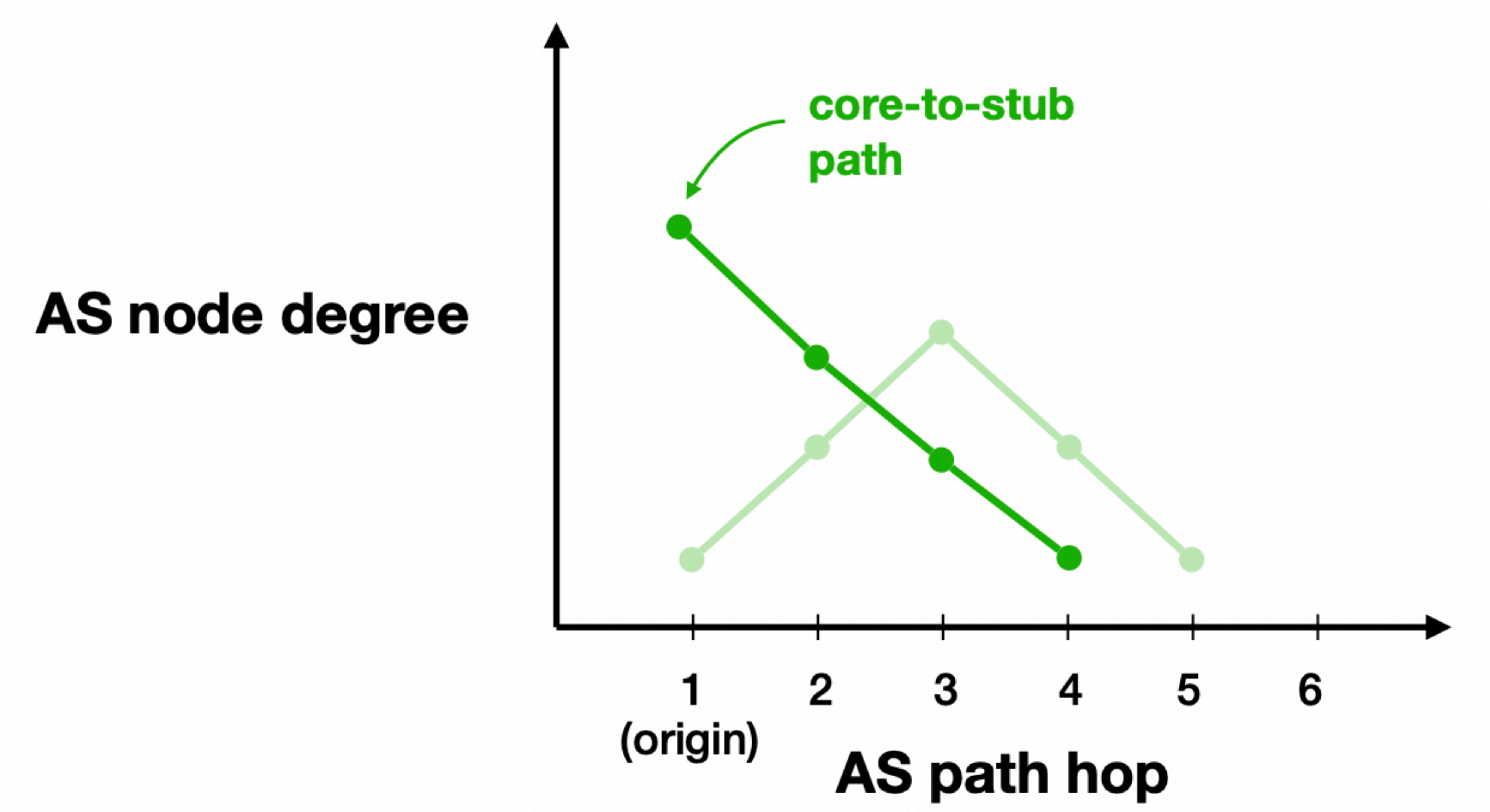
Computing Features

Finding New Links → **Computing Features** → Inferring Hijacks



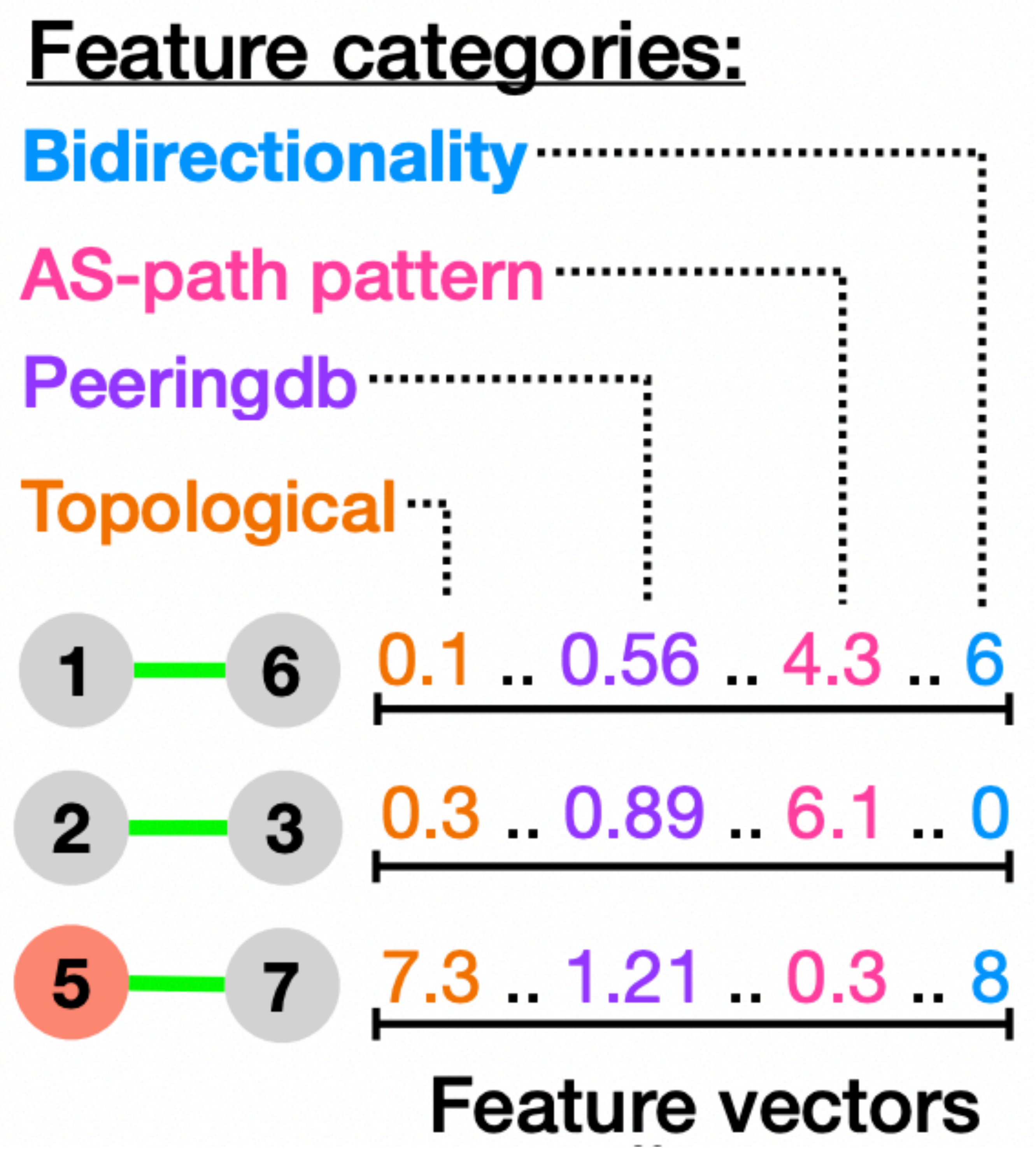
3. AS-path patterns

- examine the AS paths that include the new link to identify suspicious sequence of ASes



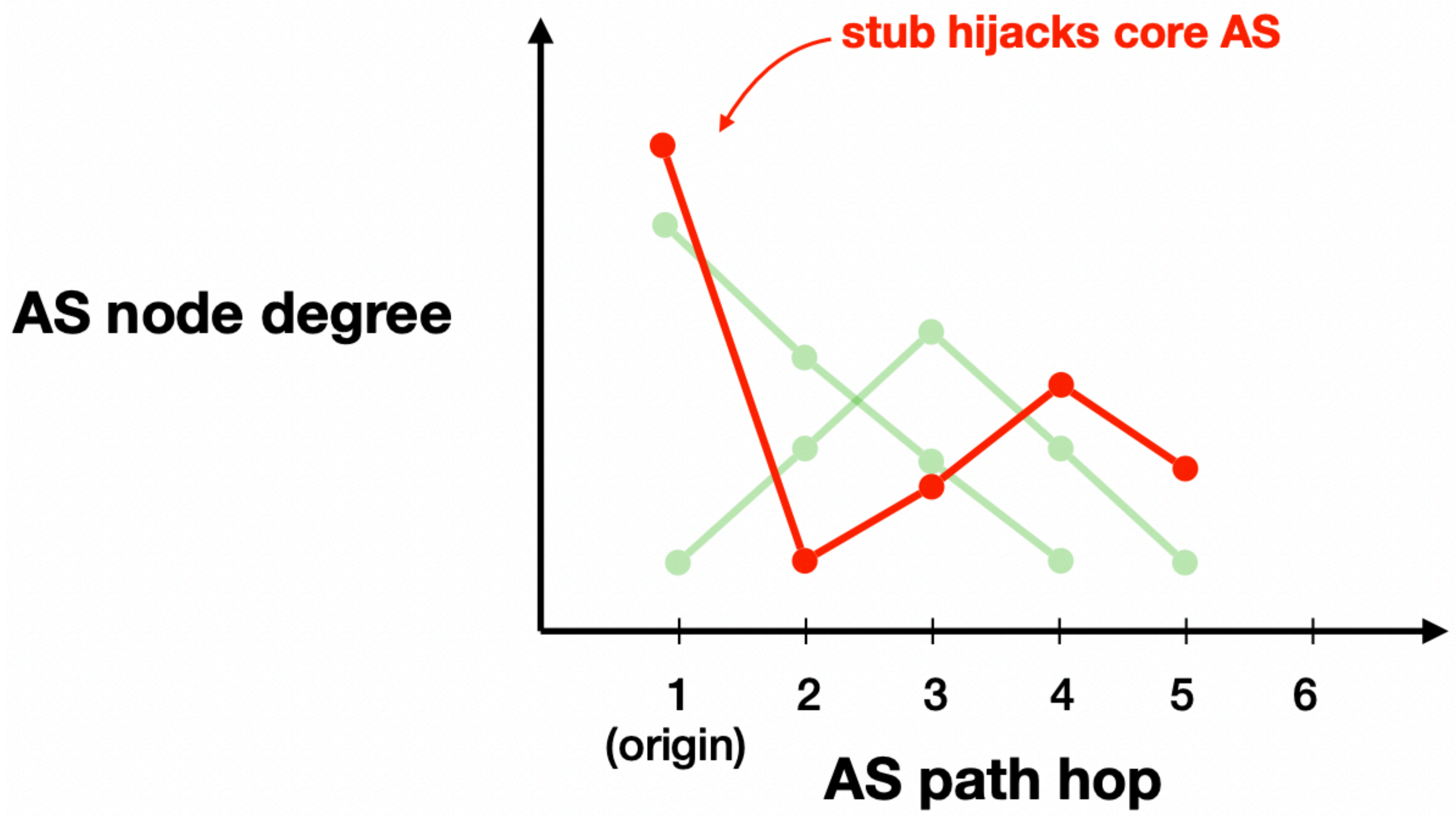
Computing Features

Finding New Links → **Computing Features** → Inferring Hijacks



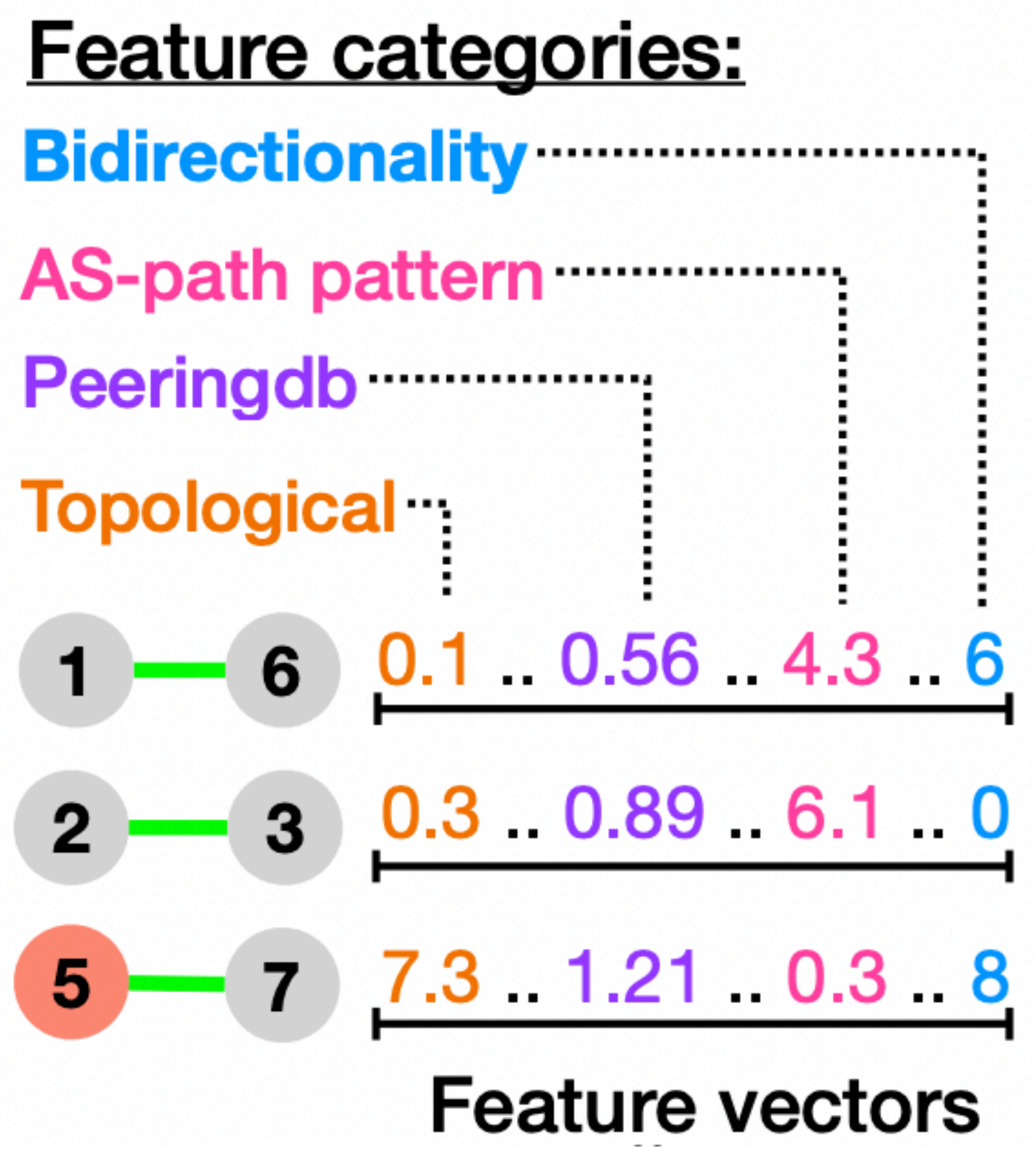
3. AS-path patterns

- examine the AS paths that include the new link to identify suspicious sequence of ASes



Computing Features

Finding New Links → **Computing Features** → Inferring Hijacks

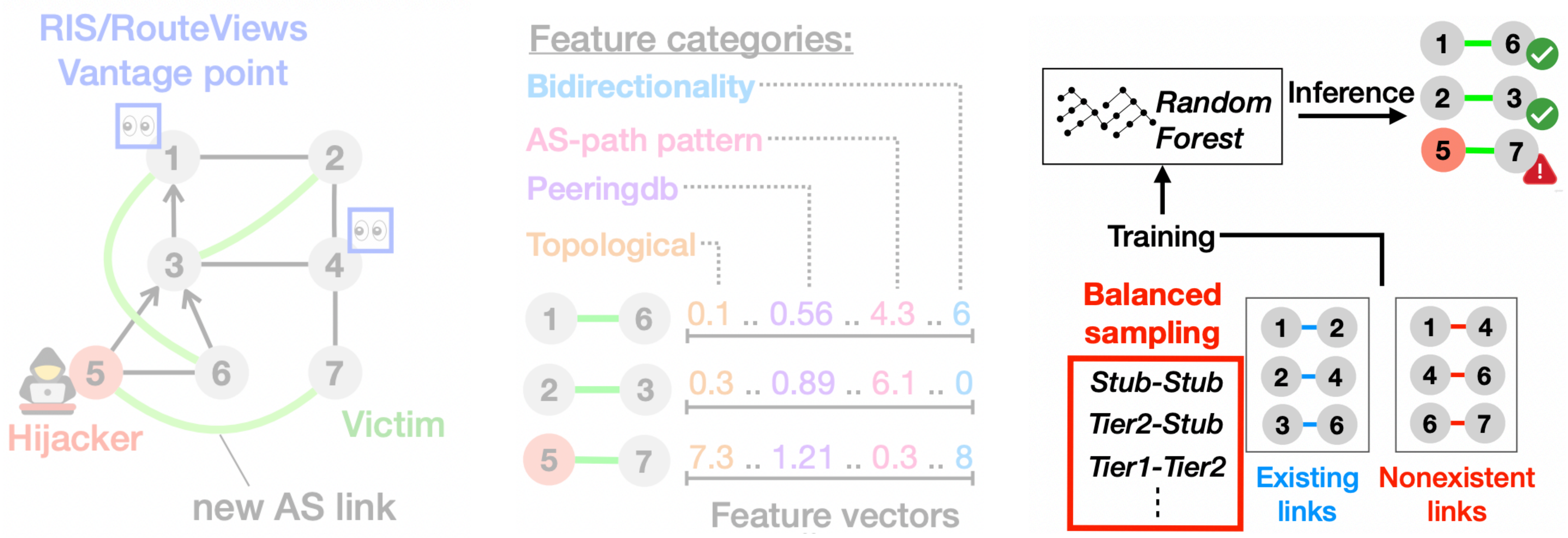


4. Bidirectionality

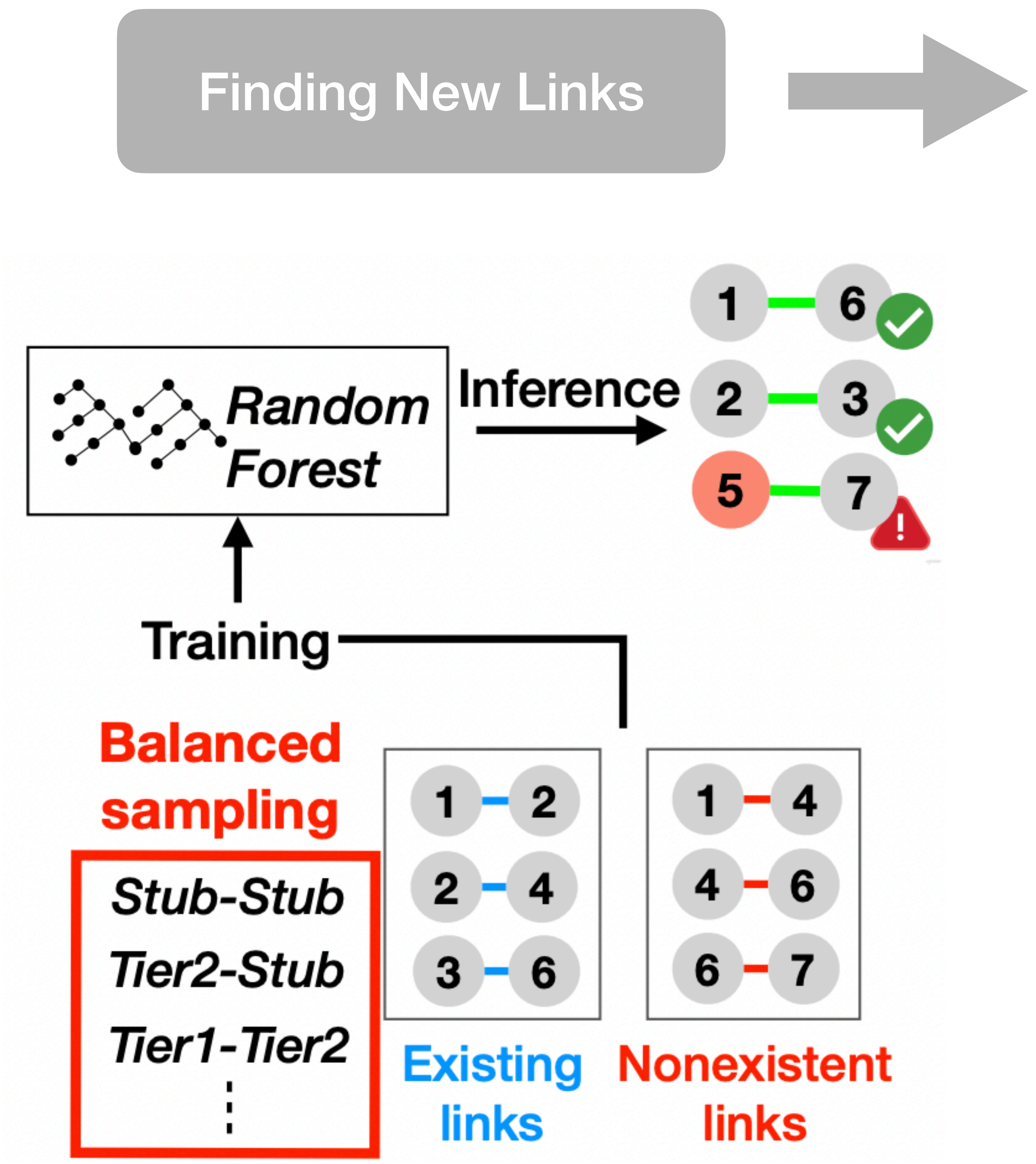
- checks whether an AS link is observed in both directions

Inferring Hijacks

Finding New Links → Computing Features → **Inferring Hijacks**



Inferring Hijacks



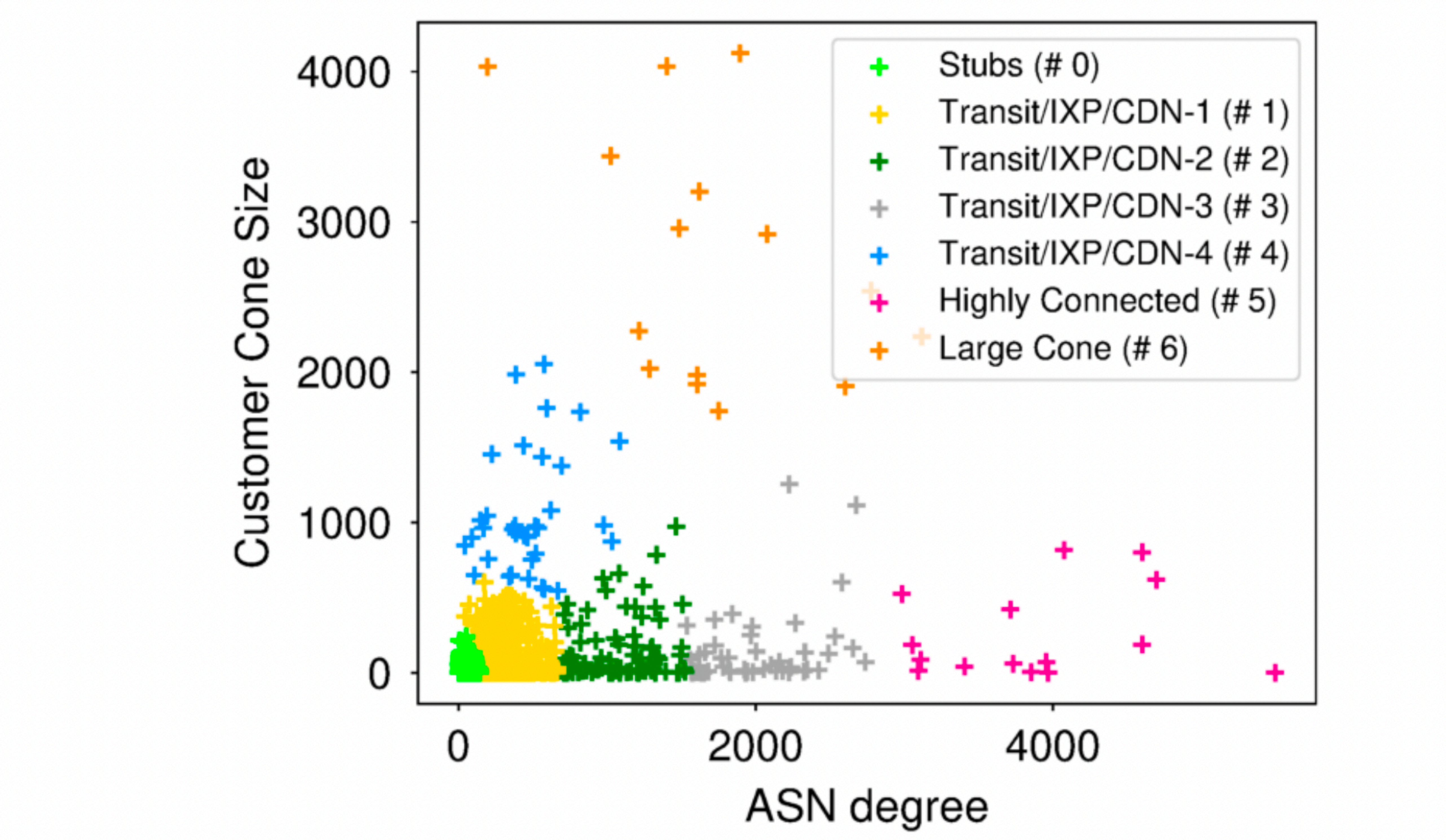
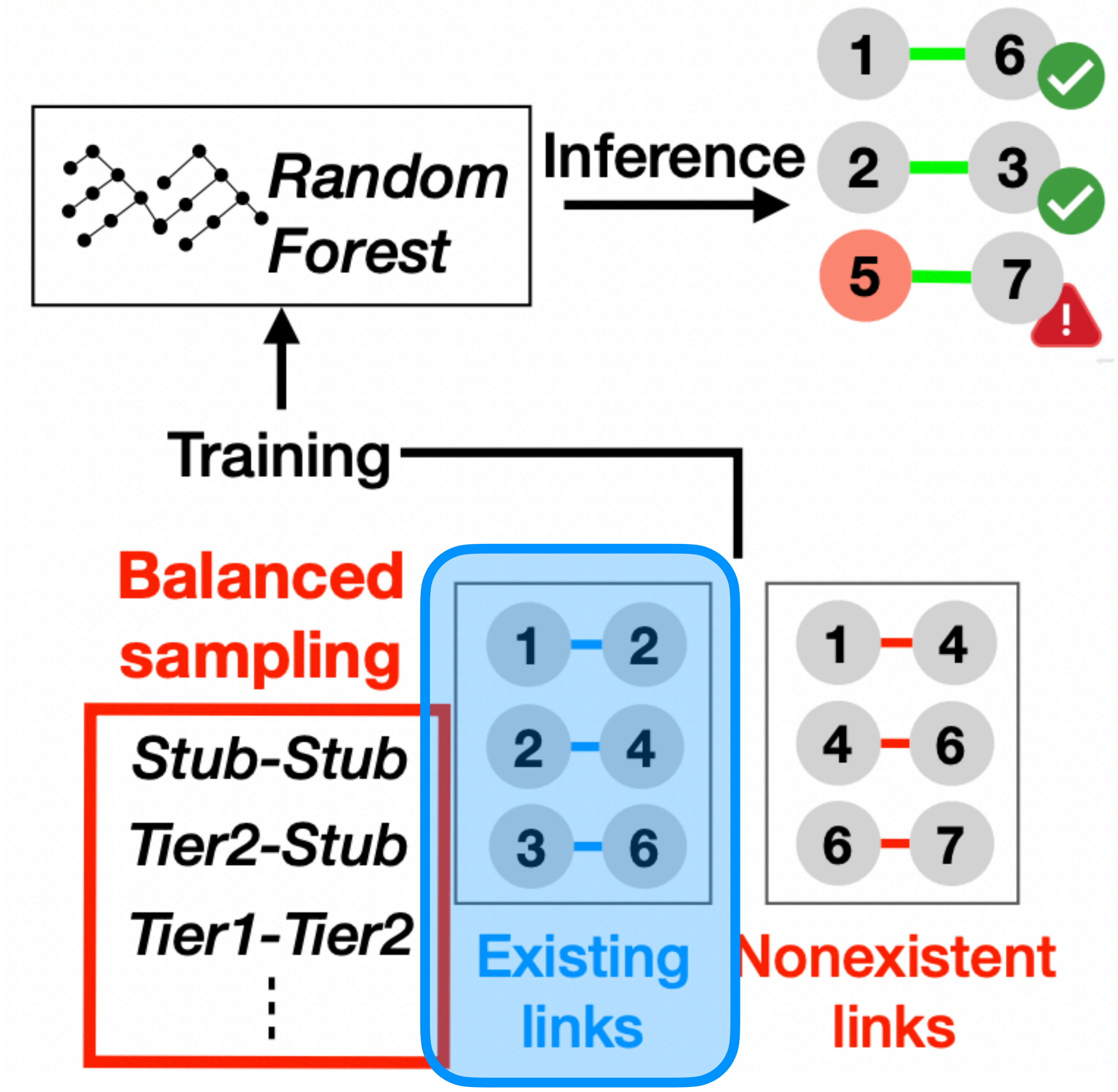


Figure 2: Computed clusters of ASes on April 30, 2022.

Inferring Hijacks

Finding New Links



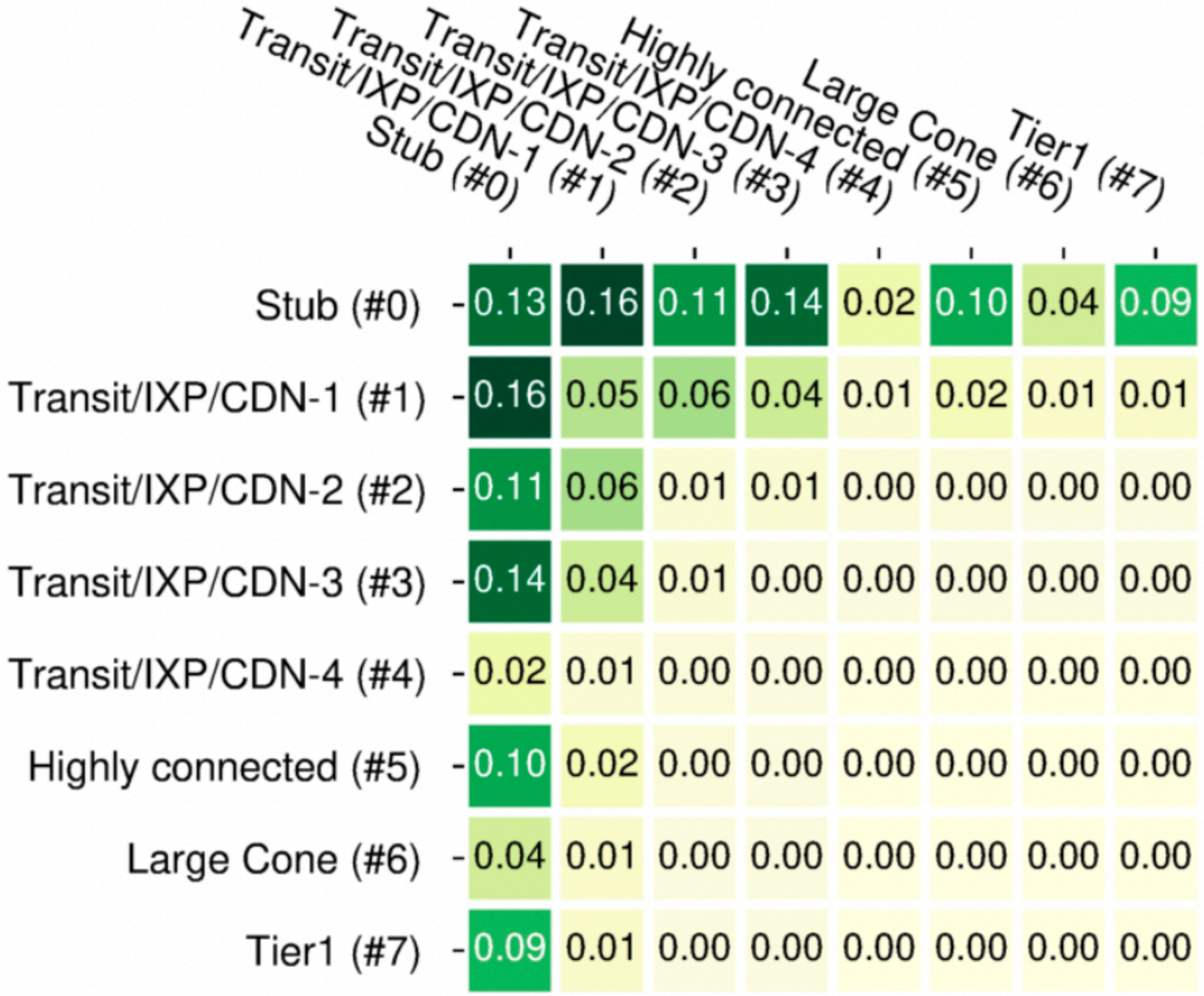
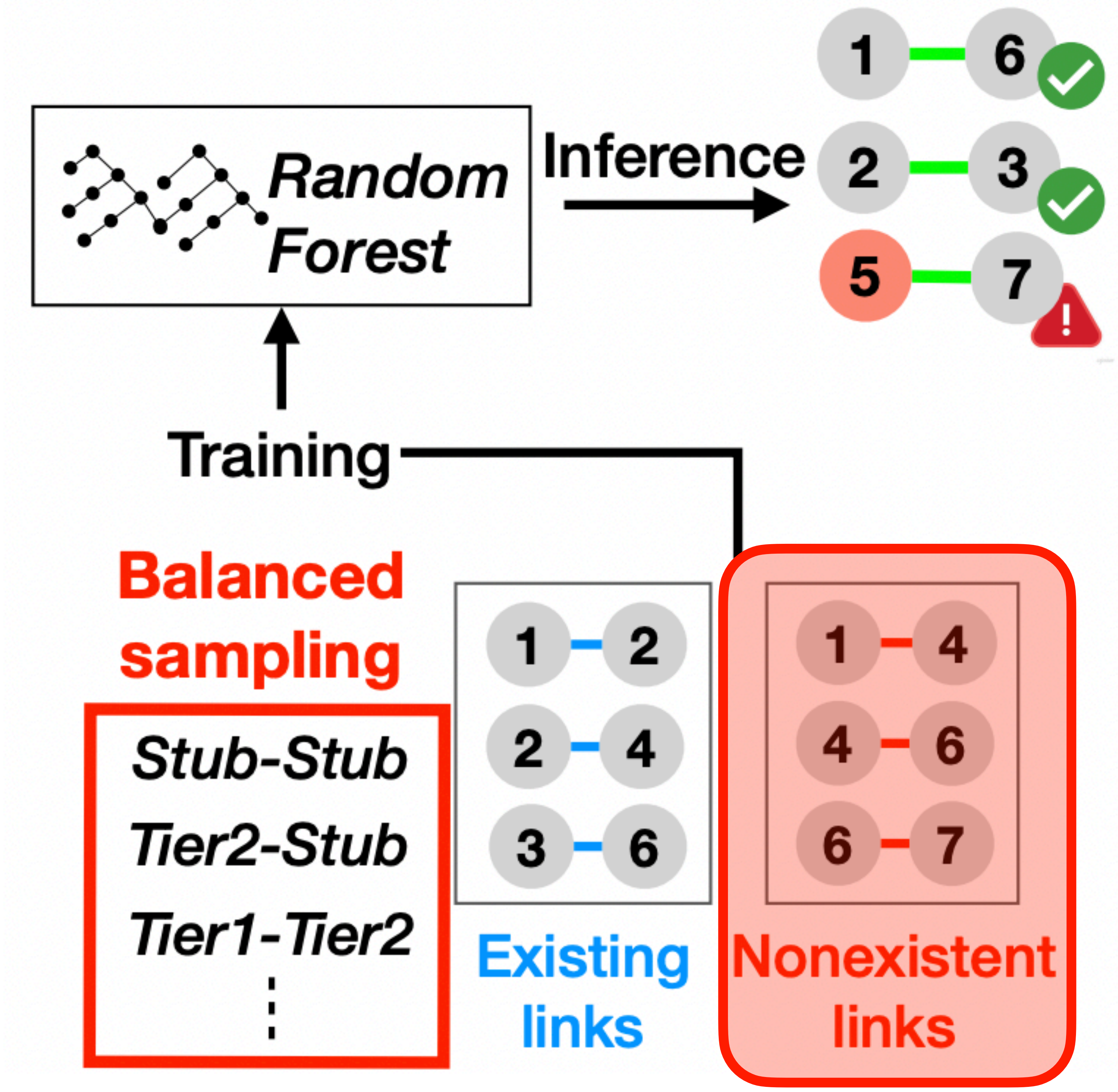


Figure 3: Link distribution within and between clusters. Each cell indicates the proportion (green means high proportion).

Inferring Hijacks

Finding New Links → Computing Features → **Inferring Hijacks**

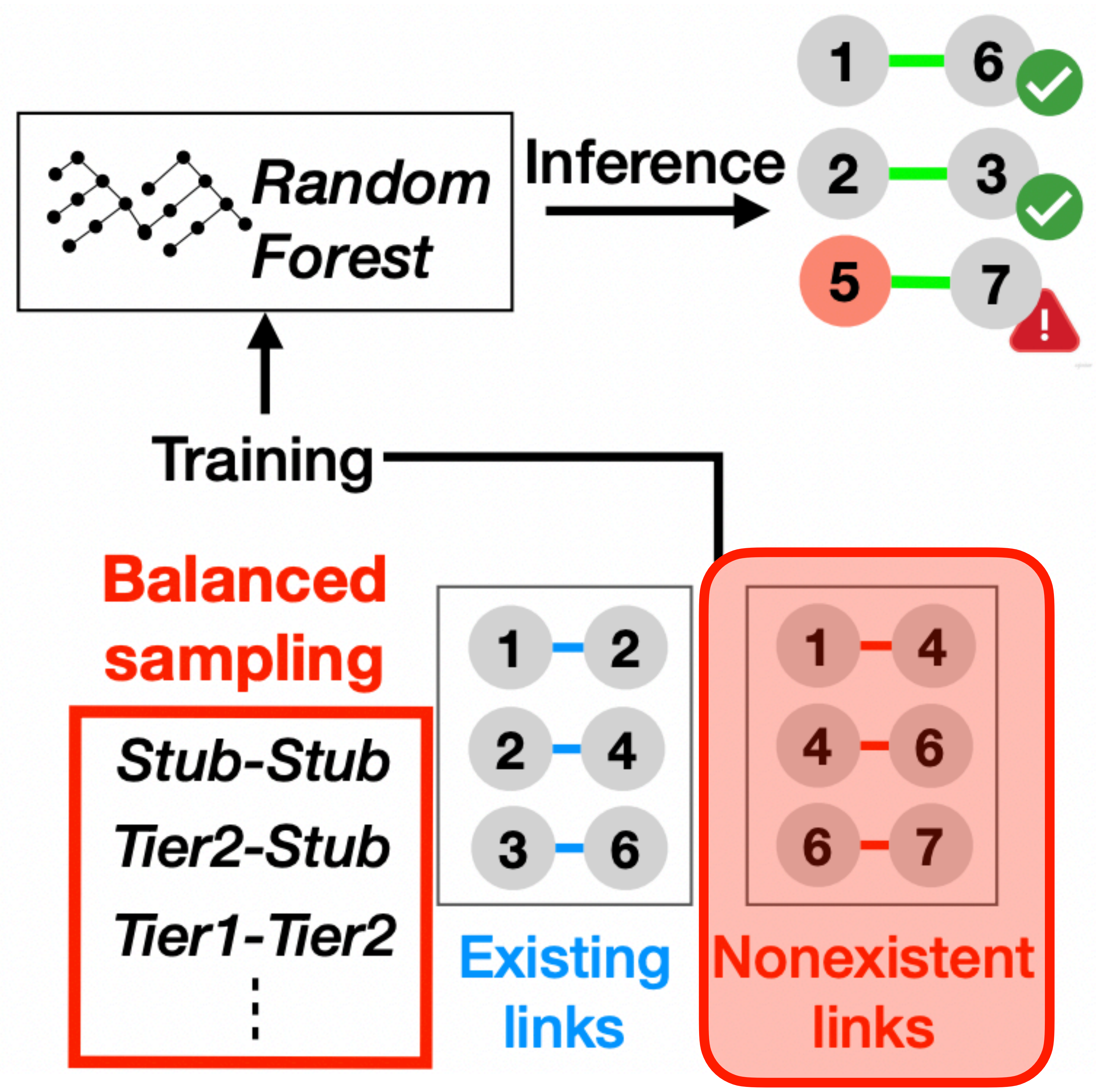


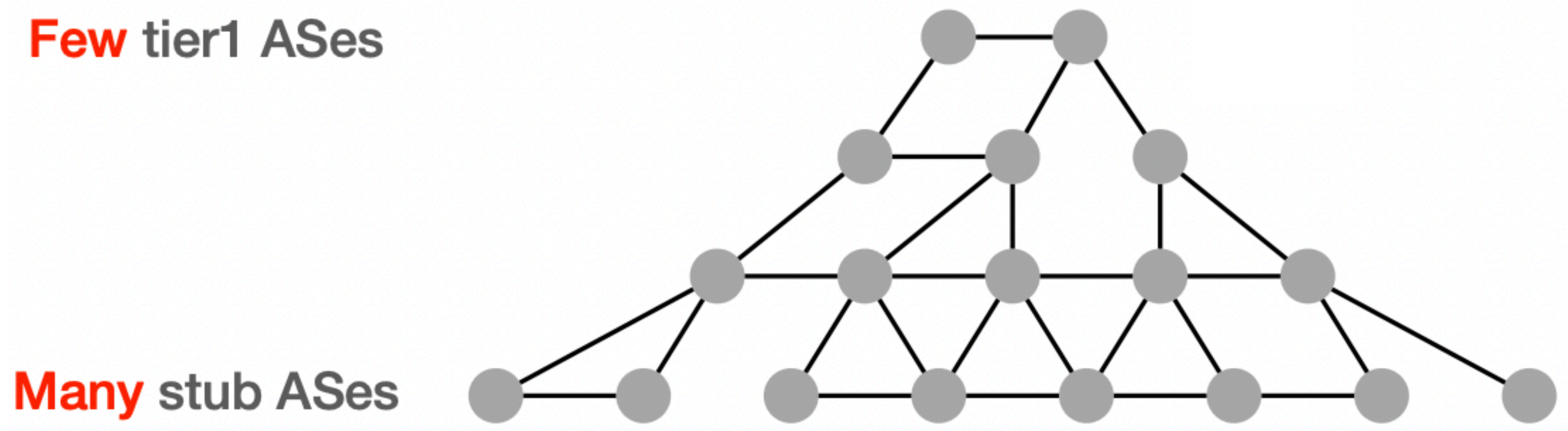
	0	1	2	3	4	5	6	7
0	0.98	0.02	0.00	0.00	0.00	0.00	0.00	0.00
1	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

(a) Random sample.

Inferring Hijacks

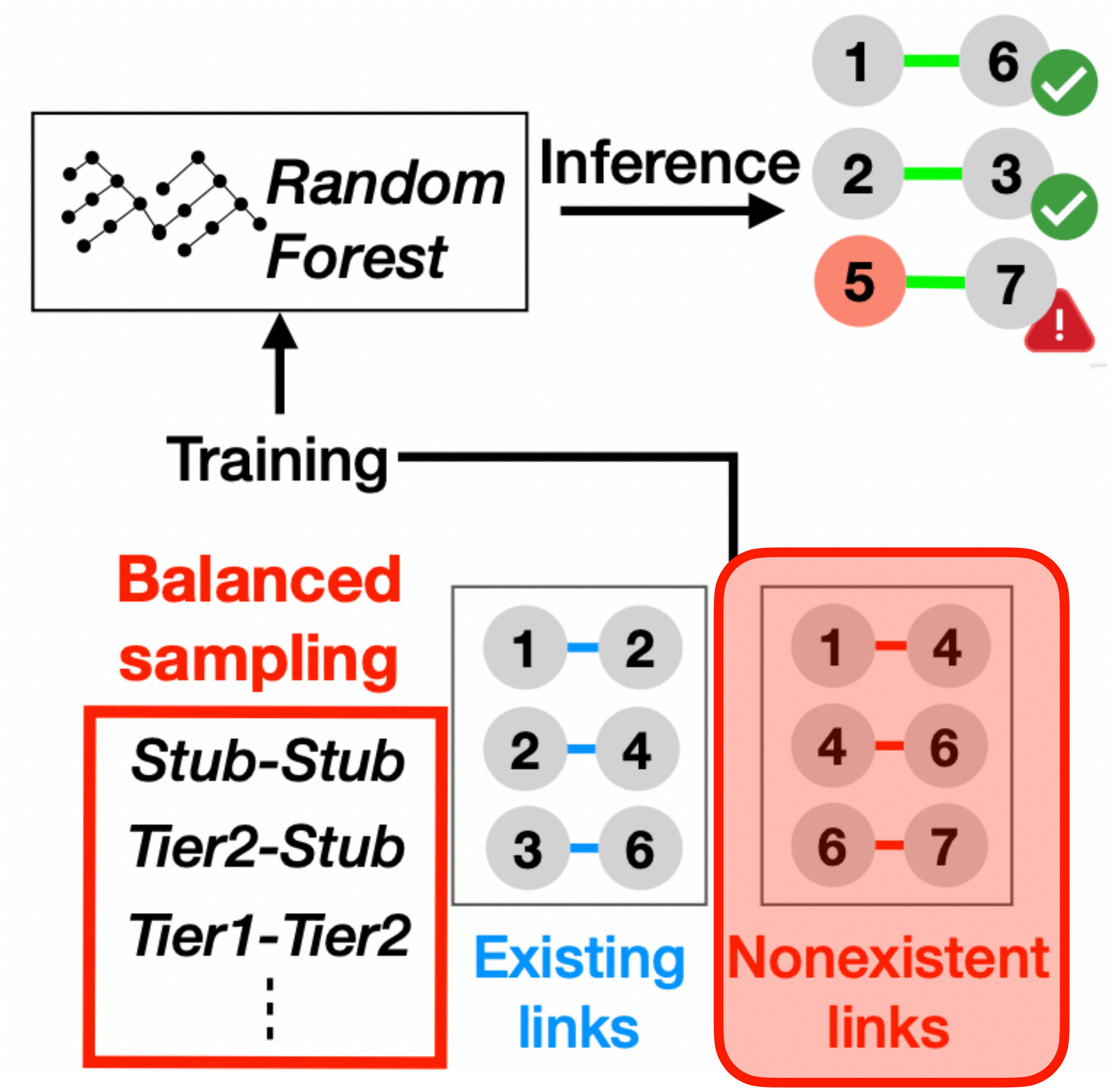
Finding New Links → Computing Features → **Inferring Hijacks**

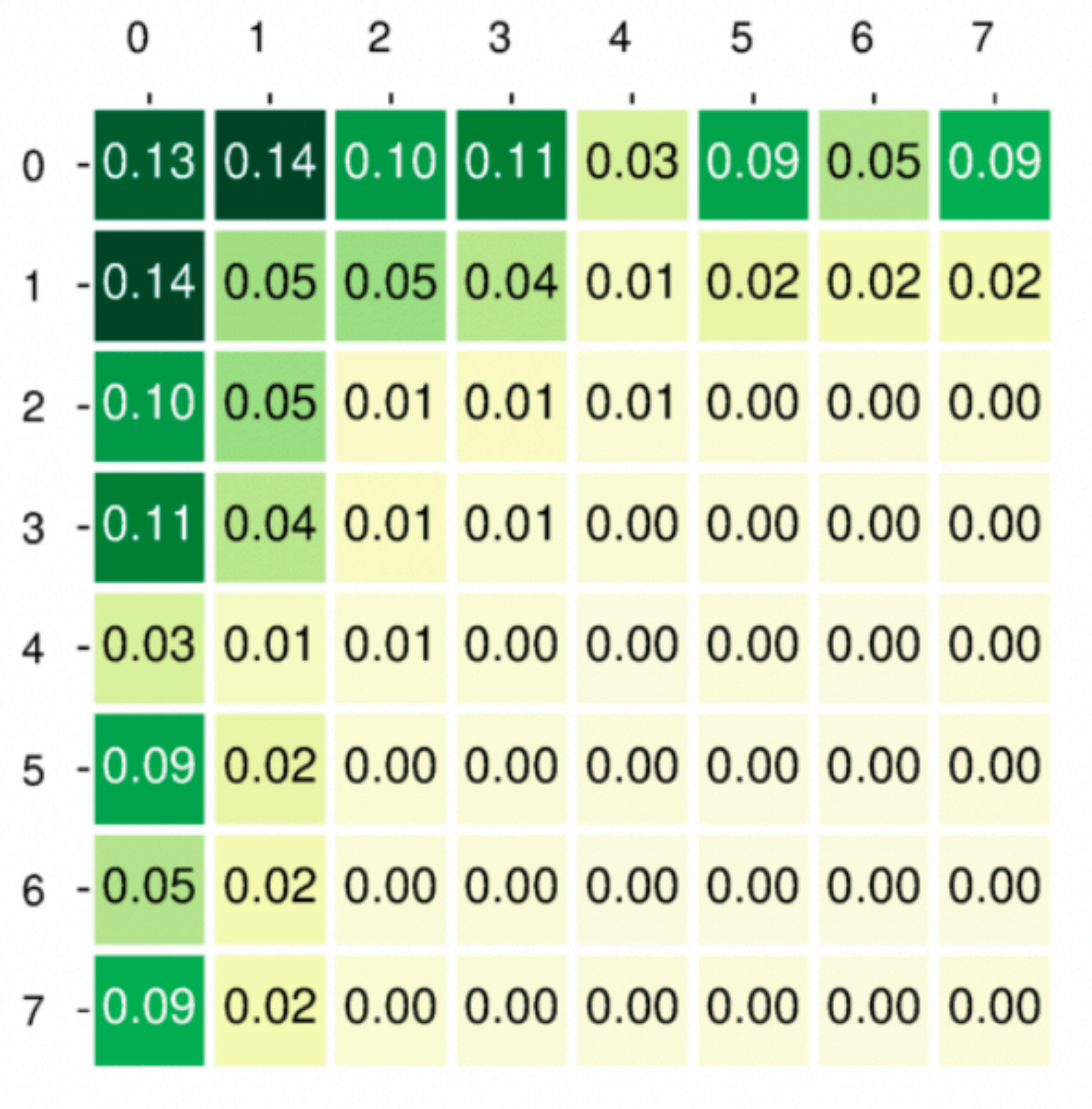




Inferring Hijacks

Finding New Links → Computing Features → **Inferring Hijacks**





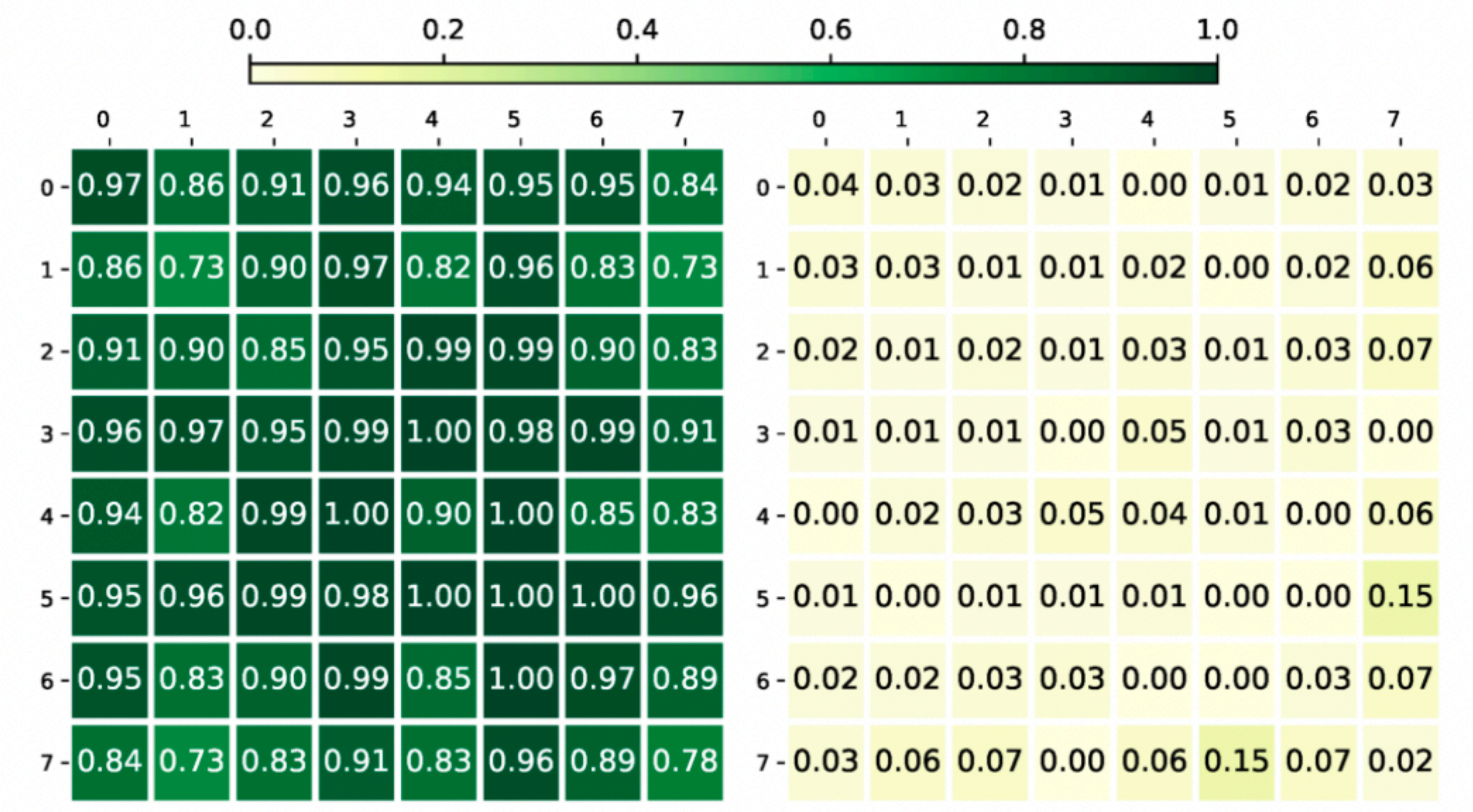
(b) DFOH's sample.

Evaluation

- Evaluated the accuracy of DFOH
  - classify 9K existing links → correctly detects 8,181 forged-origin hijacks (TPR = 0.909)
  - classify 9K nonexistent links → incorrectly inferred 171 legitimate links as forged-origin hijakcs (FPR = 0.019)
-

Evaluation

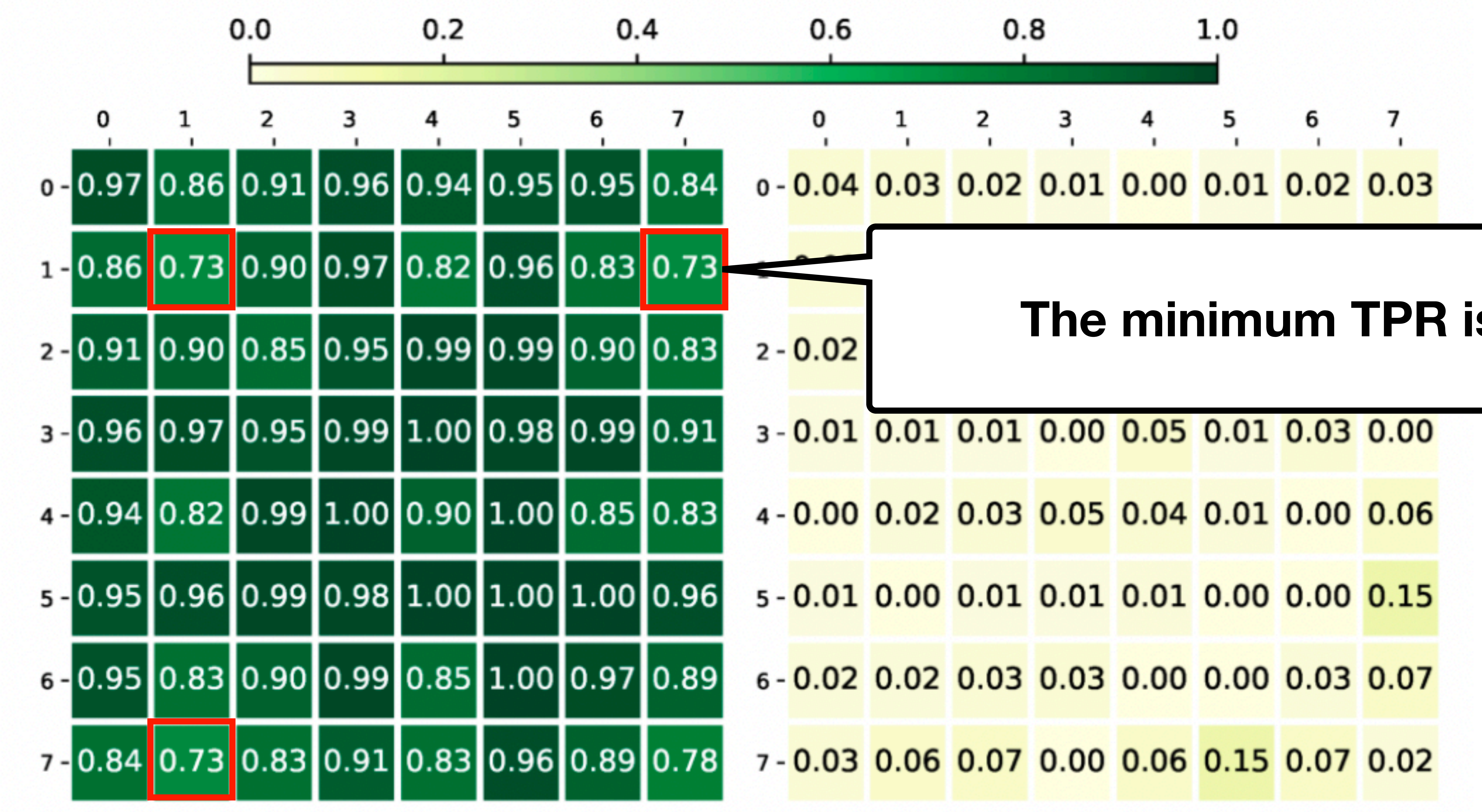
- Sample 100 links for every attack scenario



(a) TPR. (b) FPR.

Evaluation

- Sample 100 links for every attack scenario

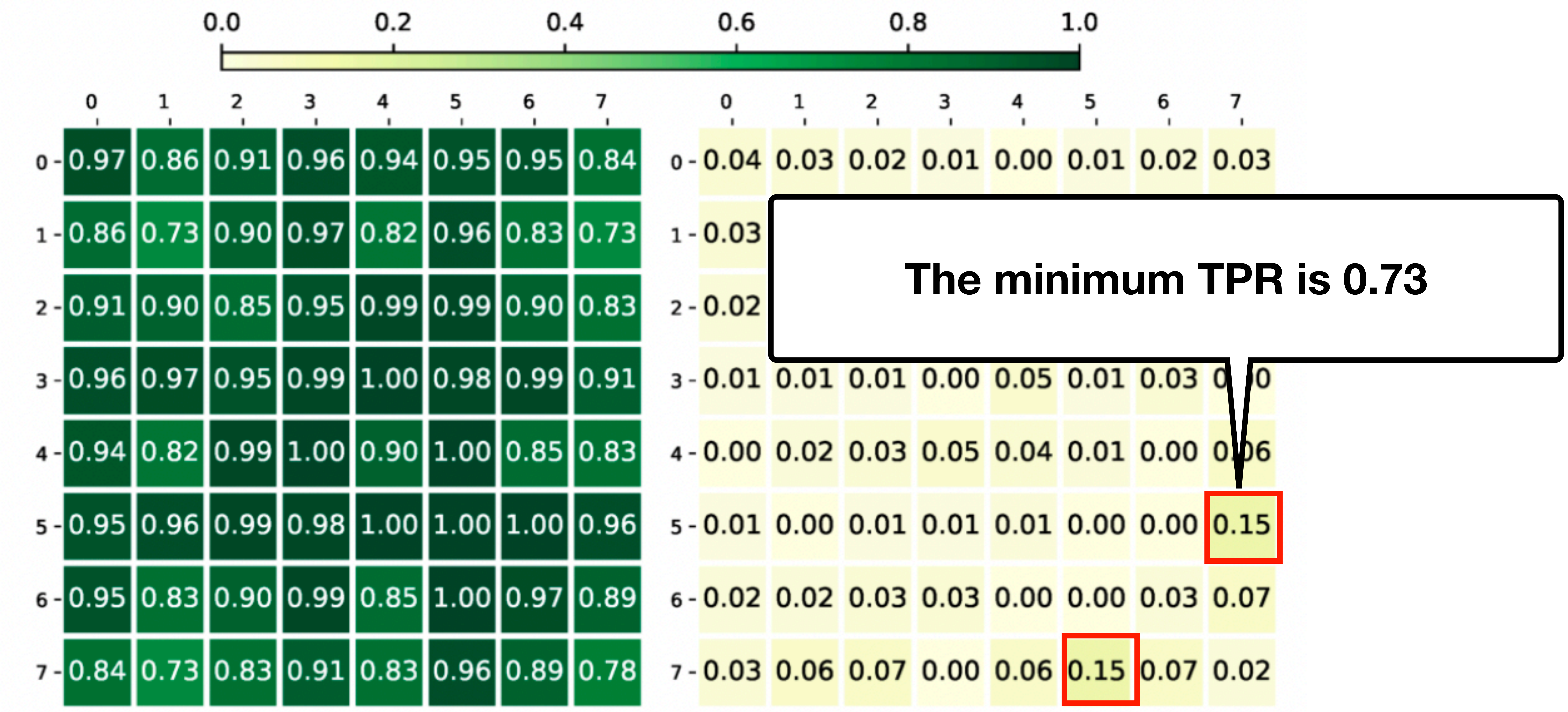


The minimum TPR is 0.73

(a) TPR. (b) FPR.

Evaluation

- Sample 100 links for every attack scenario



(a) TPR. (b) FPR.

Evaluation

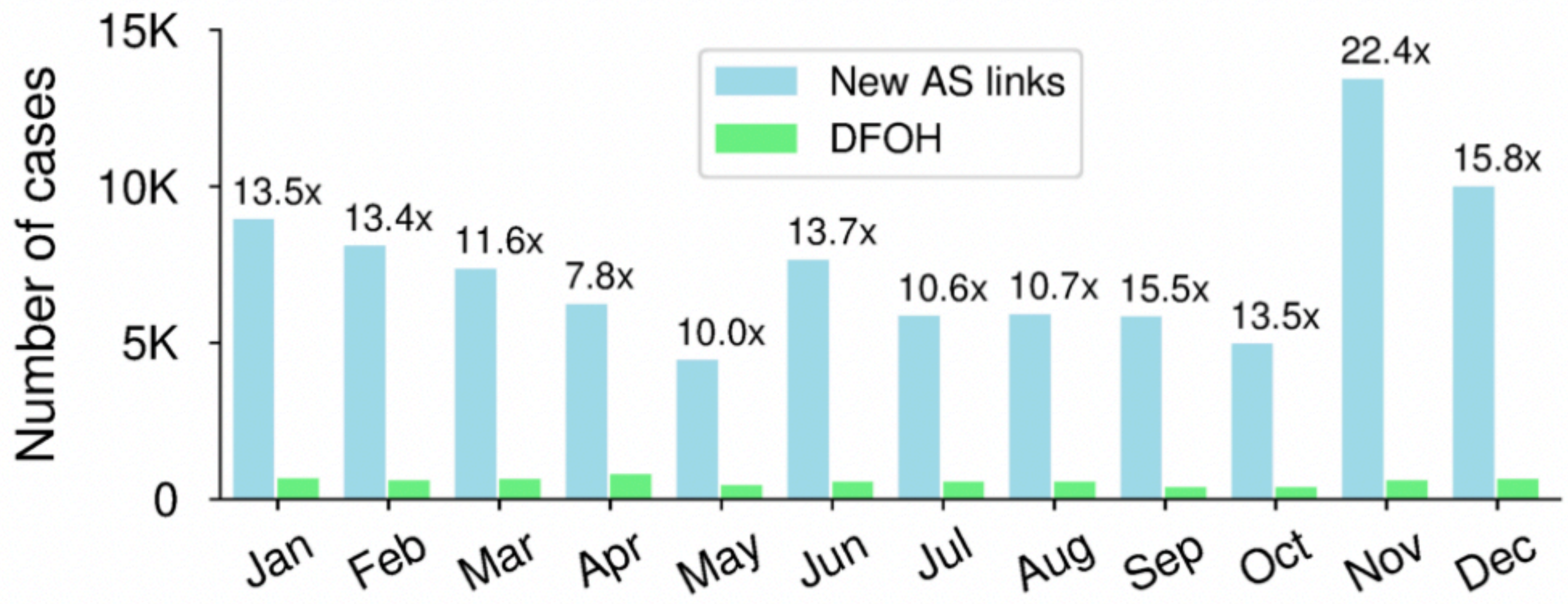


Figure 6: Number of new AS links and reported cases by DFOH for every month of 2022. We indicate the reduction factor at the top of the bars.

Evaluation

Each day, 180 new links are observed, but DFOH only classifies 17.5 of them are suspicious

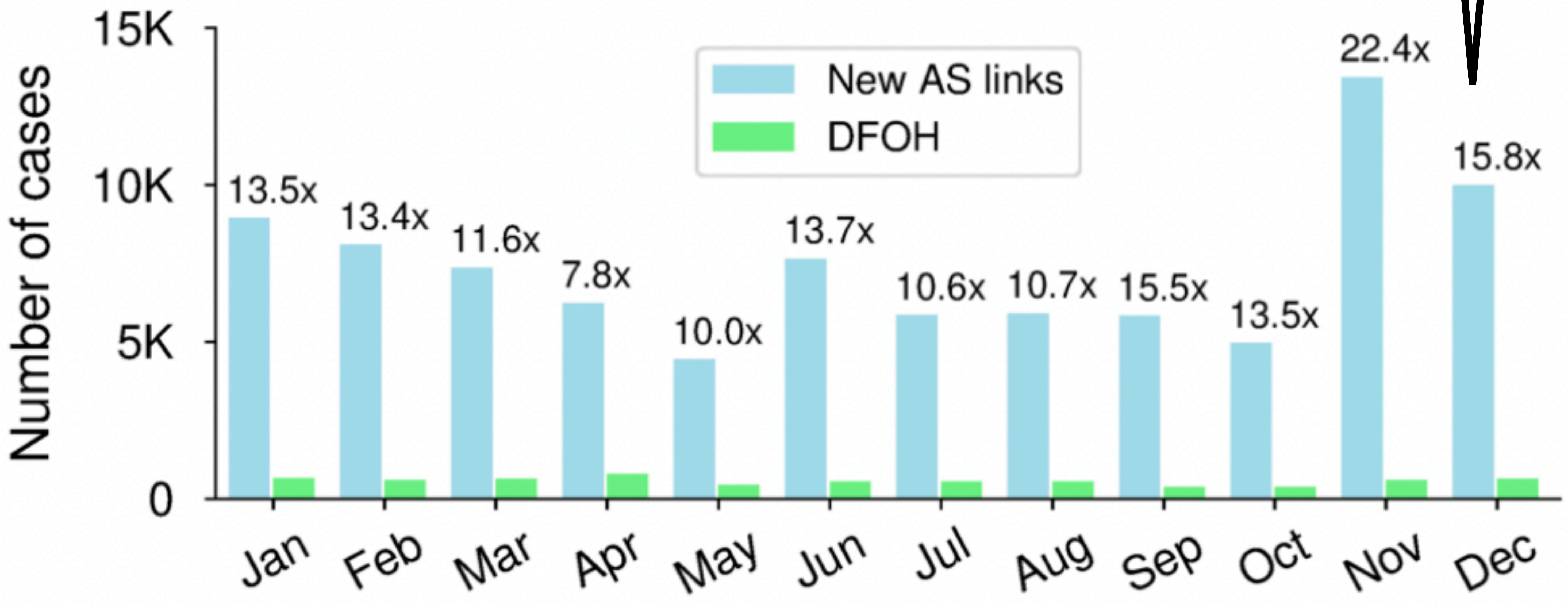


Figure 6: Number of new AS links and reported cases by DFOH for every month of 2022. We indicate the reduction factor at the top of the bars.

Conclusion

- Identify the key factors to consider when designing a forged-origin hijack detection system
- Design and present DFOH which quickly and accurately detects any forged-origin hijacks on the whole Internet
- Show the evaluation of DFOH on synthetic and real data demonstrating that DFOH is effective in defending against forged-origin hijacks

Thank you

Topological Feature

Type	Categorie	Name	Index	Description
Node-based	**Centrality Metrics**	Degree centrality	0	Fraction of nodes connected to v
		Closeness centrality	1	Average length of the shortest path between v and all other nodes
		Harmonic centrality	2	Sum of the reciprocal of the shortest path distances from all nodes to v
	Neighborhood Richness	Average neighbor degree	3	Average degree of all the neighbors of v
		Eccentricity	4	Max distance from v to all other nodes
	Topological Pattern	Number of Triangles	5	Number of triangles that include v
		Clustering	6	Fraction of possible triangles including v that exist
Pair-based	**Closeness Metrics**	Jaccard	7	Similarity between the neighbors of v_1 and v_2
		Adamic Adar	8	Closeness of v_1 and v_2 based on their shared neighbors
		Preferential attachment	9	Likelihood of v_1 and v_2 to be connected based on their degree
	Distance	Shortest Path	10	Length of the shortest path between v_1 and v_2

Topological Feature

Node-based features: Consider feature $f_i \in F_n$ and $f_i(x, G_{d,k})$ its score for node x on $G_{d,k}$, with i the feature index in Table 2. The feature value $v(f_i, d, v_1)$ is the difference induced by the new link (v_1, v_2) on the score of feature f_i for node v_1 on day d, and DFOH computes it using the following equation.

$$v(f_i, d, v_1) = f_i(v_1, G_{d,k}) - f_i(v_1, G'_{d,k})$$

$G'_{d,k} = (E'_{d,k}, V'_{d,k})$ is the graph $G_{d,k}$ that includes link (v_1, v_2), that is $E'_{d,k} = E_{d,k} \cup (v_1, v_2)$. DFOH computes the feature values for both nodes v_1 and v_2. Given that there are seven node-based features, the resulting 14-dimensional feature vector $T_{node\_based}(d, v_1, v_2)$ is the following:

$$T_{node\_based}(d, v_1, v_2) = [v(f_0, d, v_1), v(f_0, d, v_2), \\ \dots, v(f_6, d, v_1), v(f_6, d, v_2)]$$

Pair-based features: Consider feature $f_i \in F_p$ where $f_i(x, y, G_{d,k})$ is its score for the pair of nodes x, y, with i the feature index in Table 2. The feature value $v(f_i, d, v_1, v_2)$ is the difference induced by the new link (v_1, v_2) on the feature score f_i for the pair of node v_1, v_2 at day d, and DFOH computes it using the following equation.

$$v(f_i, d, v_1, v_2) = f_i(v_1, v_2, G_{d,k}) - f_i(v_1, v_2, G'_{d,k})$$

Given that there are four pair-based features, the resulting 4-dimensional feature vector $T_{pair\_based}(d, v_1, v_2)$ is:

$$T_{pair\_based}(d, v_1, v_2) = [v(f_7, d, v_1, v_2), \dots, v(f_{10}, d, v_1, v_2)]$$