DYNAMO: Towards Automated Network Attack Attribution via Density-Aware Active Learning

19/01/2024

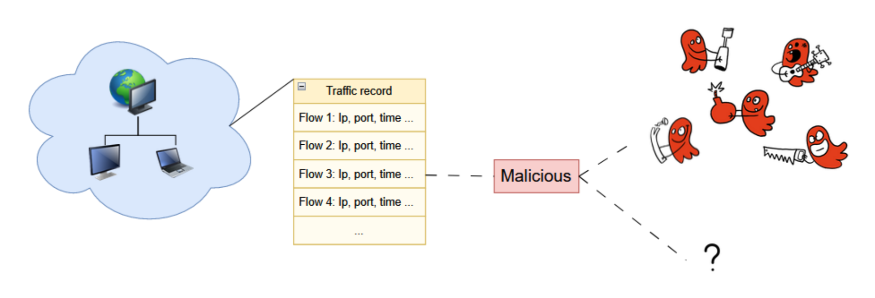
Hélène Orsini, Yufei Han

CentraleSupélec

UMR IRISA

Inria

Introduction



Introduction

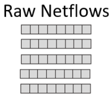
Focus

- Network traffic attribution
- Machine learning techniques

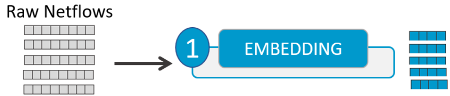
Traditional challenges

- **Annotation effort**
- **Imbalance in data volumes**
- Ever evolving traffic (concept drift)

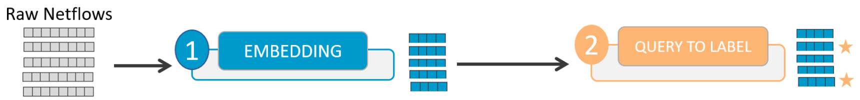
Our Proposed Framework: DYNAMO



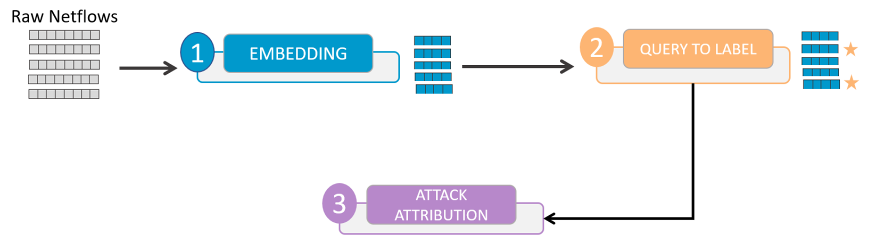
Our Proposed Framework: DYNAMO



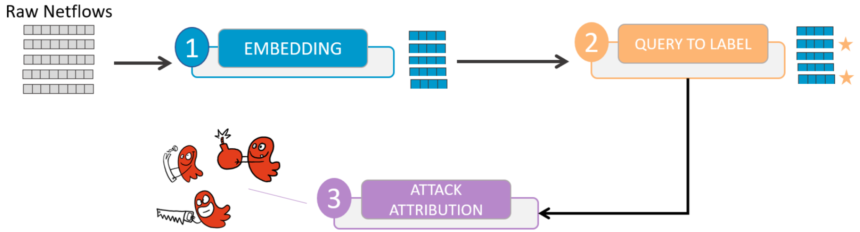
Our Proposed Framework: DYNAMO



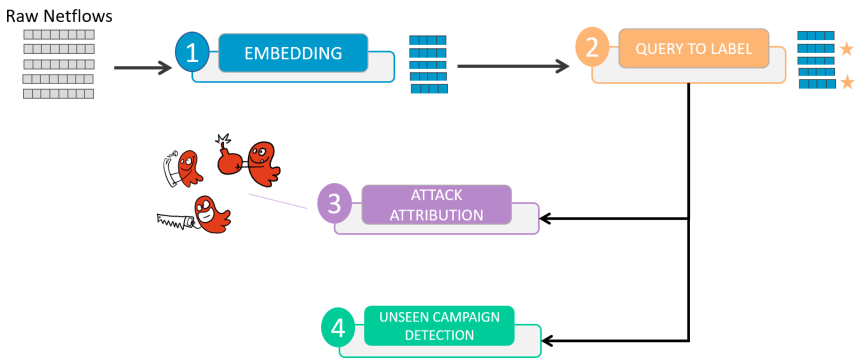
Our Proposed Framework: DYNAMO



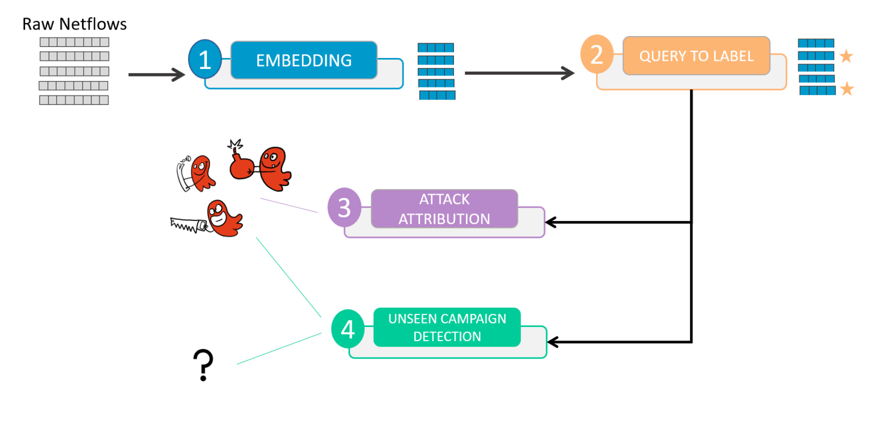
Our Proposed Framework: DYNAMO



Our Proposed Framework: DYNAMO



Our Proposed Framework: DYNAMO



Related work

Attack attribution

- Manual analysis: synthesizing and analyze report
- Machine learning-based: multi-class classification task

Query to Label: Active learning

- Uncertainty-based sampling
- Representation-based sampling

Method

1 EMBEDDING

2 QUERY TO LABEL

3 ATTACK ATTRIBUTION

4 UNSEEN CAMPAIGN DETECTION

Method

1 EMBEDDING

Self-supervised feature learning

2 QUERY TO LABEL

3 ATTACK ATTRIBUTION

4 UNSEEN CAMPAIGN DETECTION

Method

1 EMBEDDING

Self-supervised feature learning

2 QUERY TO LABEL

Density-aware active learning

3 ATTACK ATTRIBUTION

4 UNSEEN CAMPAIGN DETECTION

Method

1. **EMBEDDING**
   Self-supervised feature learning
2. **QUERY TO LABEL**
   Density-aware active learning
3. **ATTACK ATTRIBUTION**
   Attack campaigns ML Classifier
4. **UNSEEN CAMPAIGN DETECTION**

Method

1. EMBEDDING
   Self-supervised feature learning

2. QUERY TO LABEL
   Density-aware active learning

3. ATTACK ATTRIBUTION
   Attack campaigns ML Classifier

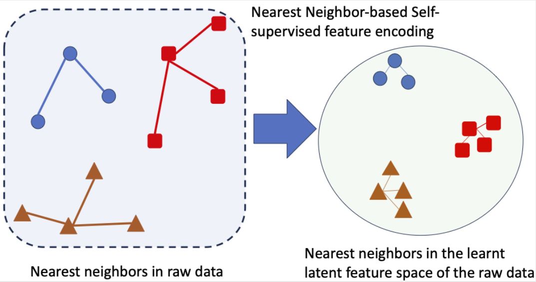
4. UNSEEN CAMPAIGN DETECTION
   Unseen attack campaigns ML detector

Nearest neighbor-based self-supervised feature encoding

1 EMBEDDING

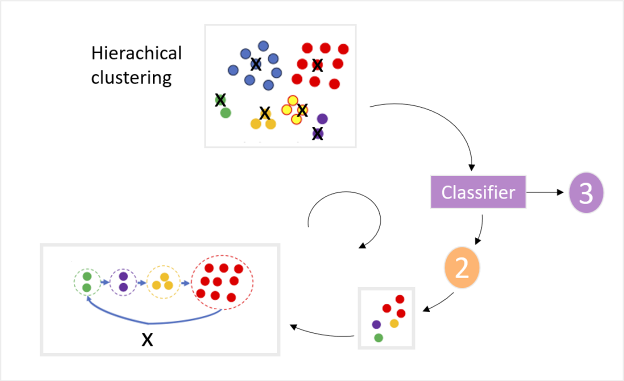
Raw feature vectors from Netflow : GraphSage method

$$\theta^* = \arg\min_{\theta} -\frac{1}{nK}\sum_{i=1}^{n}[\sum_{k=1}^{K} log(\sigma(h_\theta^T(x_i)h_\theta(x_i^{NN,k}))) - \lambda \sum_{j,x_v \notin KNN(x_i)} log(\sigma(-h_\theta^T(x_i)h_\theta(x_v)))]$$



Density-aware active learning

2 QUERY TO LABEL 3 ATTACK ATTRIBUTION



Unseen campaingn strategy

4 UNSEEN CAMPAIGN DETECTION

Pu learning

Train a classifier to distinguish between positive and negative.

Learning phase: **Positive and Unlabelled** (*only some of the positive examples in the training data are labeled and none of the negative examples are*)

$$g_{\phi}^{pu} = \arg\min_{\phi} \frac{\pi}{n_p} \sum_{x_i \in S} [\ell(g_{\phi}^{\text{pu}}(h_{\theta}(x_i)), y_i = +1) - \ell(g_{\phi}^{\text{pu}}(h_{\theta}(x_i)), y_i = -1)] + \frac{1}{n_u} \sum_{x_i \in X_{\text{unlabeled}}} \ell(g_{\phi}^{\text{pu}}(h_{\theta}(x_i)), y_i = -1)$$

Goals

Q1 - Raw vs Embedded

Q2 - Effectiveness of DYNAMO's density-aware active learning module ?

Q3 - Effectiveness ML-based unseen campaign detection ?

Set up

Dataset: CTU13

D_{attr}^{test}	D_{ood}^{test}
D_{attr}^{train}	D_{ood}^{train}

Scenario	Flows	%
1	39933	9,23
2	18 8839	4,35
3	26 759	6,18
4	1 719	0,4
5	695	0,16
6	4 431	1,02
7	37	0,0085
8	5 052	1,17
9	179 880	41,57
10	106 315	24,57
11	8 161	1,89
12	2 143	0,50
13	38 791	8,96

Set up

Dataset: CTU13

D_{attr}^{test}	D_{ood}^{test}
D_{attr}^{train}	D_{ood}^{train}

Scenario	Flows	%
1	39933	9,23
2	18 8839	4,35
3	26 759	6,18
4	1 719	0,4
5	695	0,16
6	4 431	1,02
7	37	0,0085
8	5 052	1,17
9	179 880	41,57
10	106 315	24,57
11	8 161	1,89
12	2 143	0,50
13	38 791	8,96

Baseline 2 QUERY TO LABEL

- Select $p\%$ of D_{attr}^{train} (1k, 2k, 3k, 4k, and 5k)
- Random, UAL, and DYNAMO

Set up

Dataset: CTU13

D_{attr}^{test}	D_{ood}^{test}
D_{attr}^{train}	D_{ood}^{train}

Scenario	Flows	%
1	39933	9,23
2	18 8839	4,35
3	26 759	6,18
4	1 719	0,4
5	695	0,16
6	4 431	1,02
7	37	0,0085
8	5 052	1,17
9	179 880	41,57
10	106 315	24,57
11	8 161	1,89
12	2 143	0,50
13	38 791	8,96

Baseline

2 QUERY TO LABEL

- Select $p\%$ of D_{attr}^{train} (1k, 2k, 3k, 4k, and 5k)
- Random, UAL, and DYNAMO

Attack attribution

3 ATTACK ATTRIBUTION

- Gradient Boosting Trees, Label Spreading
- Macro F1, Balanced Accuracy

Set up

Dataset: CTU13

D_{attr}^{test}	D_{ood}^{test}
D_{attr}^{train}	D_{ood}^{train}

Scenario	Flows	%
1	39933	9,23
2	18 8839	4,35
3	26 759	6,18
4	1 719	0,4
5	695	0,16
6	4 431	1,02
7	37	0,0085
8	5 052	1,17
9	179 880	41,57
10	106 315	24,57
11	8 161	1,89
12	2 143	0,50
13	38 791	8,96

Baseline

2 QUERY TO LABEL

- Select $p\%$ of D_{attr}^{train} (1k, 2k, 3k, 4k, and 5k)
- Random, UAL, and DYNAMO

Attack attribution

3 ATTACK ATTRIBUTION

- Gradient Boosting Trees, Label Spreading
- Macro F1, Balanced Accuracy

Unseen campaign detection

4 UNSEEN CAMPAIGN DETECTION

- ISO, OCSVM, and PU
- Macro F1, AUC

Results - Attack attribution Q1

Mean ± Standard deviation of GBT trained with the full supervision method

NB	Raw		Embedding	
	Macro F1	Balanced Acc	Macro F1	Balanced Acc
29,918 (p=20%)	0.644 ∓ 0.012	0.637 ∓ 0.015	0.770 ∓ 0.011	0.74 ∓ 0.015
59,835 (p=40%)	0.687 ∓ 0.009	0.648 ∓ 0.010	0.780 ∓ 0.006	0.754 ∓ 0.007
89,754, (p=60%)	0.675 ∓ 0.006)	0.637 ∓ 0.006	0.792 ∓ 0.004	0.765 ∓ 0.005
119,671 (p=80%)	0.685 ∓ 0.006)	0.665 ∓ 0.006	0.801 ∓ 0.002	0.772 ∓ 0.003
149,589 (p=100%)	0.674 ∓ 0.006	0.670 ∓ 0.002	0.805 ∓ 0.002	0.786 ∓ 0.003

Embedded data increases both macro-F1 score and balanced accuracy.

Results - Attack attribution Q2

Mean ± Standard deviation of Macro F1-score

	Attack attribution with the latent feature learned by the self-supervised learning module					
	Random Selection		DYNAMO		UAL	
NB	GB	LS	GB	LS	GB	LS
1000 (p=0.7%)	0.611 ∓ 0.024	0.637 ∓ 0.036	**0.695 ∓ 0.024**	0.631 ∓ 0.000	0.607 ∓ 0.016	0.574 ∓ 0.067
2000 (p=1.3%)	0.653 ∓ 0.018	0.694 ∓ 0.022	**0.745 ∓ 0.021**	0.677 ∓ 0.000	0.613 ∓ 0.016	0.608 ∓ 0.017
3000 (p=2.0%)	0.673 ∓ 0.017	0.712 ∓ 0.016	**0.764 ∓ 0.016**	0.688 ∓ 0.000	0.723 ∓ 0.013	0.654 ∓ 0.027
4000 (p=2.6%)	0.686 ∓ 0.013	0.723 ∓ 0.049	**0.781 ∓ 0.015**	0.707 ∓ 0.000	0.773 ∓ 0.002	0.689 ∓ 0.019
5000 (p=3.3%)	0.697 ∓ 0.013	0.732 ∓ 0.012	**0.791 ∓ 0.011**	0.708 ∓ 0.000	0.785 ∓ 0.009	0.702 ∓ 0.020

Dynamo outperforms UAL (GB, LS) and Random (GB)

Results - Unseen campaign Q3

Mean ± Standard deviation of Macro F1-score

	Unseen campaign detection with the latent feature learned by the self-supervised learning module								
	Random Selection			DYNAMO			UAL		
NB	ISO	OCSVM	PU	ISO	OCSVM	PU	ISO	OCSVM	PU
1000 (p=0.7%)	0.748 ∓ 0.005	0.853 ∓ 0.000	1.000 ∓ 0.000	0.913 ∓ 0.007	0.921 ∓ 0.005	**1.000 ∓ 0.000**	0.832 ∓ 0.043	0.898 ∓ 0.013	1.000 ∓ 0.000
2000 (p=1.3%)	0.754 ∓ 0.005	0.762 ∓ 0.000	1.000 ∓ 0.000	0.905 ∓ 0.010	0.913 ∓ 0.006	**1.000 ∓ 0.000**	0.817 ∓ 0.044	0.880 ∓ 0.016	1.000 ∓ 0.000
3000 (p=2.0%)	0.672 ∓ 0.005	0.696 ∓ 0.000	1.000 ∓ 0.000	0.765 ∓ 0.009	0.909 ∓ 0.008	**1.000 ∓ 0.000**	0.789 ∓ 0.018	0.860 ∓ 0.021	1.000 ∓ 0.000
4000 (p=2.6%)	0.758 ∓ 0.007	0.687 ∓ 0.000	1.000 ∓ 0.000	0.897 ∓ 0.010	0.904 ∓ 0.010	**1.000 ∓ 0.000**	0.789 ∓ 0.046	0.848 ∓ 0.048	1.000 ∓ 0.000
5000 (p=3.3%)	0.754 ∓ 0.007	0.689 ∓ 0.026	1.000 ∓ 0.000	0.891 ∓ 0.009	0.898 ∓ 0.008	**1.000 ∓ 0.000**	0.794 ∓ 0.059	0.842 ∓ 0.068	1.000 ∓ 0.000

Pu performs best

Dynamo provide better result for ISO and OCSVM

Key Takeaway

- ML-Based attack attribution challenges: scarce label data, imbalanced campaign distribution
- Self-Supervised feature encoding boosts attack attribution
- Density-aware active learning helps overcome imbalanced data issue
- Positive-Unlabeled learning outperforms for unseen campaign detection

Summary and perspectives

- DYNAMO, a weakly supervised ML-based pipeline designed for automated network attack attribution without requiring exhaustive labeling of attack campaigns
- Density-aware Machine Learning
- Positive-unlabeled weakly supervised learning

Next steps

- Adapt to new dataset (Poneypot)
- Add transfer learning environment

Thanks, Questions ?