Detecting and Mitigating Sampling Bias in Cybersecurity with Unlabeled Data

Saravanan Thirumuruganathan, Fatih Deniz*, Issa Khalil*, Ting Yu*
Mohamed Nabeel+, Mourad Ouzzani*

*Qatar Computing Research Institute, HBKU
+Palo Alto Networks

14/08/2024

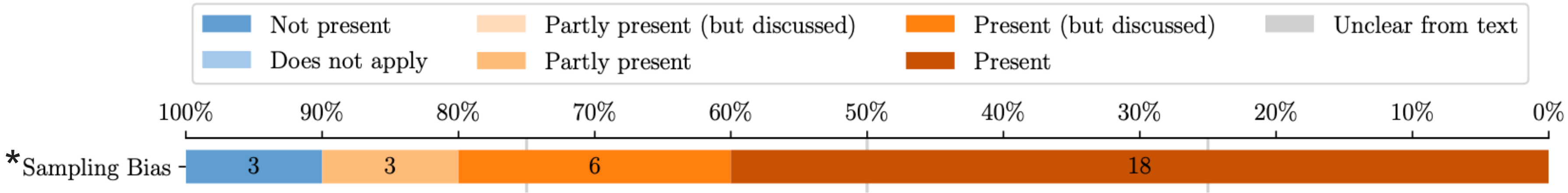
QCRI
معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute
جامعة حمد بن خليفة
HAMAD BIN KHALIFA UNIVERSITY

Outline

- Sampling Bias in Cybersecurity
- Problem Definition
- Approach Overview
- Detection Algorithms
- Mitigation Strategies
- Results

Sampling Bias in Cybersecurity

Sampling Bias: The collected data does not sufficiently represent the true data distribution of the underlying problem.

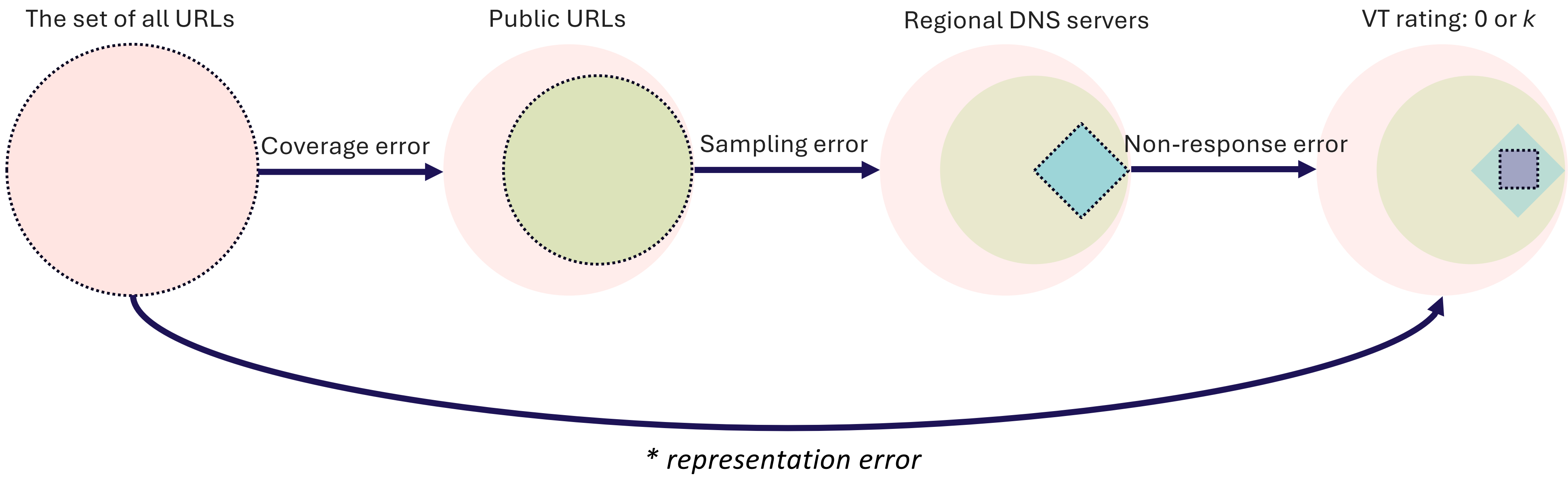


Most common causes:

- Convenience sampling
- Labelling heuristics

* Arp, Daniel, et al. "Dos and don'ts of machine learning in computer security." *31st USENIX Security Symposium (USENIX Security 22)*. 2022.

Sampling Bias in Cybersecurity



* Robert M Groves and Lars Lyberg. Total survey error: Past, present, and future. *Public opinion quarterly*, 74(5):849–879, 2010.

Sampling Bias in Cybersecurity

Concept Drift: Causes performance degradation of ML classifiers as the deployment data diverges from the training data.

Distribution Shift: A broader term that encompasses both concept drift and other shifts in data distribution, such as covariate shift or label shift.

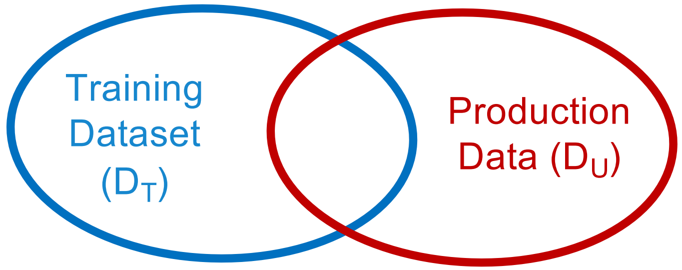
Sampling Bias: Occurs when there is a discrepancy between the training data and deployment data distributions right from the start.

Key Insight: Unlike concept/distribution drift, sampling bias exists before the classifier is deployed, and addressing it requires different strategies.

Problem Definition

Given:

- **Labeled training dataset** D_T
  $D_T = \{(x_1,y_1),\ldots,(x_n,y_n)\}$)
- **A classifier C_T** trained using D_T
- **Unlabeled deployment data** D_U
  $D_U = \{x_1,x_2,\ldots,x_m\}$



Objective:

1. Detect if **C_T** is biased or can be used on D_U.
2. If there is sampling bias, train a classifier with a higher performance on D_U than the C_T.

Overview Detection

Detection Algorithms:

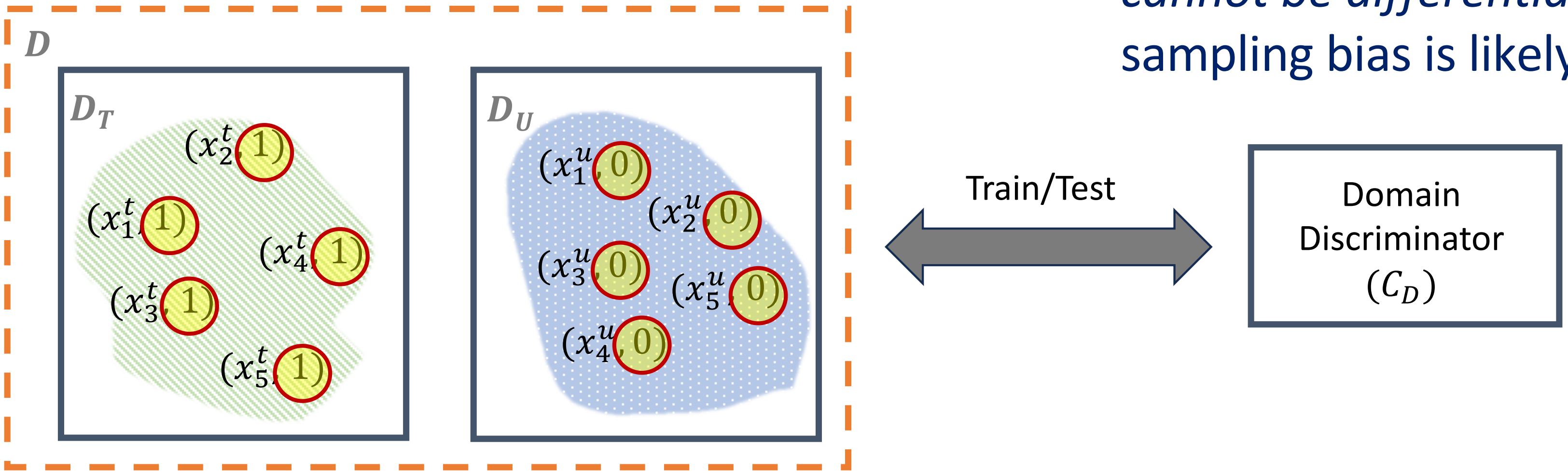
- Domain discrimination
- Distribution of kth nearest neighbor distance

Mitigation Strategies:

- Contrastive Learning for Bias Mitigation (CONL-BM)
- Bias Mitigation Using Cycle Consistency (CYC-BM)

Domain Discrimination

Intuition: If *two distributions cannot be differentiated,* sampling bias is likely minimal.

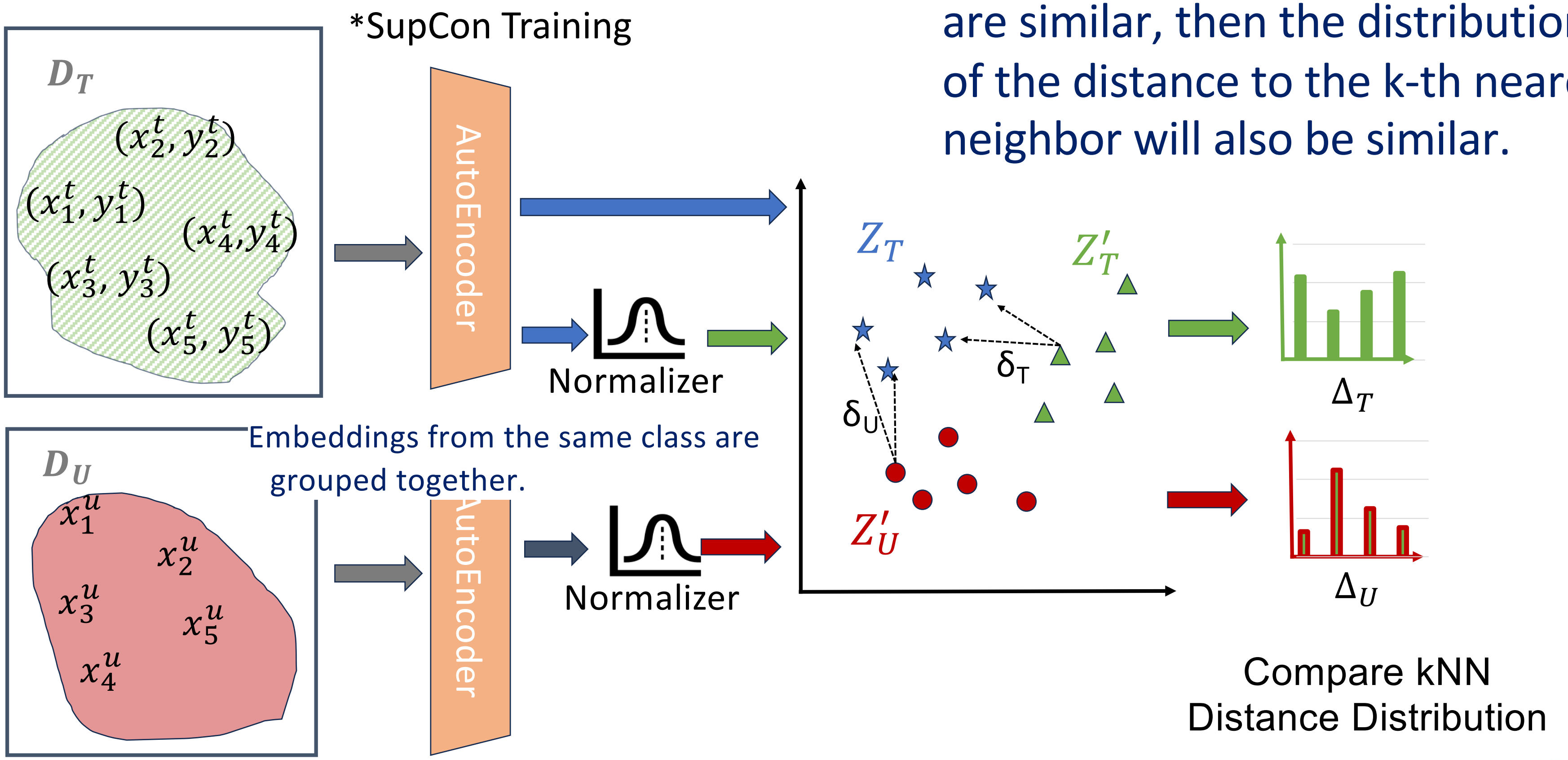


$$D_U = \{(x, 0) \quad \forall x \in D_U\}$$

Randomly split D_T into equal sized partitions D_T^1 and D_T^2
Randomly split D_U into equal sized partitions D_U^1 and D_U^2
Train classifier C_D on $D_T^1 \cup D_U^1$
acc = Accuracy of C_D on $D_T^2 \cup D_U^2$

k-NN Based Bias Detection

Intuition: If two data distributions are similar, then the distribution of the distance to the k-th nearest neighbor will also be similar.



* Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Overview Mitigation

Detection Algorithms:

- Domain discrimination
- Distribution of kth nearest neighbor distance

Mitigation Strategies:

- Contrastive Learning for Bias Mitigation (CONL-BM)
- Bias Mitigation Using Cycle Consistency (CYC-BM)

Key insight: Design a better latent space to obtain better pseudo labels.

Contrastive Learning for Bias Mitigation

Challenge: To identify positive/negative pairs without having the label information

Objective Function: Soft Nearest Neighbor Loss from D_U + CE from D_T

- $\forall\ x_i \in D_U \Rightarrow y_i$ is the (pseudo) label for x_i
- sim(·,·) measures the similarity between two data items

$$\mathcal{L}_{snn} = -\frac{1}{|B|}\sum_{i=1}^{|B|} \log \frac{\sum_{i \neq j, \hat{y}_i = \hat{y}_j} \exp(-sim(x_i, x_j)/\tau)}{\sum_{i \neq k} \exp(-sim(x_i, x_k)/\tau)}$$

Bias Mitigation Using Cycle Consistency

Challenge: Estimating pseudo-label accuracy when there is no label for D_U

Solution: Use interrelated classifiers.

- Step 1: Train C_T on D_T and obtain pseudo labels for D_U
- Step 2: Train C_U on D_U and obtain predictions for D_T

- Result: Indirectly evaluate pseudo-labeling strategy accuracy

Experimental Setup

Conducted experiments over widely used benchmark datasets from:

- Android malware
- Microsoft PE
- Intrusion Detection Systems
- Domain (URL)

Experimented with different settings:

- Sampling strategies (adversarial, benign, mixed, etc.)
- Classifiers (SVM, RF, LR, DL, etc.)

Key finding: We can successfully detect sampling bias and reclaim 90% of lost deployment f-score.

Results – Detection

We accurately detect sampling bias.

	TN-AZ	AZ-TN	Emb-UCSB	Emb-BODMAS
DomDisc	0.97	0.98	0.99	0.97
kNN-Dist	0.99	0.98	0.99	0.98
PM	0.89	0.91	0.88	0.9
CM	0.91	0.86	0.9	0.86
f-Div	0.78	0.81	0.77	0.72
ViM	0.94	0.96	0.96	0.95
MaxLogit	0.92	0.93	0.96	0.91

Results – Detection

Our detection approach is classifier agnostic.

	SVM	**RF**	**LogReg**	**DL**	**Trans**
DomDisc	0.96	0.97	0.97	0.98	0.98
kNN-Dist	0.95	0.95	0.96	0.96	0.96
PM	0.86	0.86	0.84	0.83	0.88
CM	0.89	0.88	0.86	0.83	0.87
f-Div	0.77	0.79	0.8	0.77	0.78
ViM	0.92	0.93	0.92	0.94	0.95
MaxLogit	0.92	0.91	0.91	0.92	0.93

Results – Mitigation

We mitigate over 90% of the adverse effects of sampling bias.

	TN-AZ	AZ-TN	Emb-UCSB	Emb-BODMAS
max Δ	16.9	12.9	26.1	27.3
ConL-BM	12.3	10.2	19.1	22.3
CyC-BM	14.3	10.6	21.3	22.7
DANN	9.8	8.1	14.7	16.1
SHOT	6.5	6.2	10.1	11.3
VAT	4.4	4.1	8.8	7.9
FixMatch	7.5	6.6	11.3	13.4

Results – Mitigation

Our approach works for different sampling strategies.

	Adv.	Benign	Mxd	Mxd-2	Mxd-3
max Δ	24.9	7.9	19.7	11.4	22.7
ConL-BM	18.2	6.7	16.5	12.6	11.2
CyC-BM	19.1	7.1	17.1	9.9	12.4
DANN	11.2	5.4	13.2	8.4	7.2
SHOT	7.6	3.3	7.8	5.4	5.8
VAT	5.1	4.1	7.5	4.8	6.9
FixMatch	8.3	5.1	6.2	5.6	4.4

Results – Mitigation

Our mitigation approach is classifier agnostic.

	SVM	**RF**	**LogReg**	**DL**	**Trans.**
max Δ	9.2	11.3	8.7	16.9	16.2
ConL-BM	8.8	10.2	6.4	12.1	11.9
CyC-BM	9.1	10.3	6.8	14.2	13.8
DANN	6.6	6.8	4.2	9.8	9.8
SHOT	5.1	5.6	3.9	6.6	6.4
VAT	4.4	4.7	3.8	4.3	4.1
FixMatch	6.1	6.3	5.9	7.9	7.7

Summary

- Sampling bias is a very prevalent issue in cybersecurity.
- We addressed this using two steps:
  - Detection
  - Mitigation
- We can successfully detect sampling bias and reclaim 90% of lost deployment f-score.

Thanks