netMosaic

Harnessing Public Code Repositories to Develop Production-Ready ML Artifacts for Networking

Punnal Ismail Khan

UC Santa Barbara

ML Model(s) for Networks

- **Efforts in Past Decades**
  1000+ research publications, multiple products/startups, billions of dollars invested

ML Model(s) for Networks

- **Efforts in Past Decades**

  1000+ research publications, multiple products/startups, billions of dollars invested

- **Expectations**
  - Easy to develop ML models for any given problem and target environment
  - Abundance of production-ready ML models---ready for high-stake decision-making

ML Model(s) for Networks

- **Efforts in Past Decades**

  1000+ research publications, multiple products/startups, billions of dollars invested

- **Expectations**
  - Easy to develop ML models for any given problem and target environment
  - Abundance of production-ready ML models---ready for high-stake decision-making

- **Reality**
  - Availability of **public datasets** dictates choice of learning problem and environment
  - Abundance of ML artifacts with high performance in **controlled “lab” settings**

Can we Deploy Existing ML Models in Production?

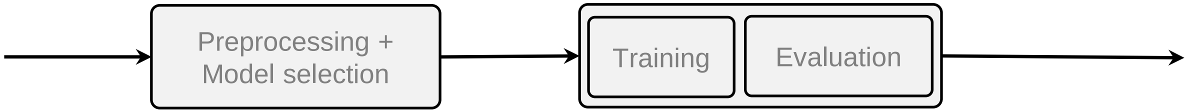
Problem	Dataset(s)	Model(s)
Detect VPN traffic	Public VPN dataset [20]	1-D CNN [61]
Detect Heartbleed traffic	CIC-IDS-2017 [54]	RF Classifier [54]
Detect Malicious traffic (IDS)	CIC-IDS-2017 [54], Campus dataset	nPrintML [32]
Anomaly Detection	Mirai dataset [44]	Kitsune [44]
OS Fingerprinting	CIC-IDS-2017 [54]	nPrintML [32]
IoT Device Fingerprinting	UNSW-IoT [56]	Iisy [63]
Adaptive Bit-rate	HSDPA Norway [49]	Pensieve [42]

Can we Deploy Existing ML Models in Production?

Problem	Dataset(s)	Model(s)	Model Generalizability Issues
Detect VPN traffic	Public VPN dataset [20]	1-D CNN [61]	Shortcut learning
Detect Heartbleed traffic	CIC-IDS-2017 [54]	RF Classifier [54]	Out-of-distribution samples
Detect Malicious traffic (IDS)	CIC-IDS-2017 [54], Campus dataset	nPrintML [32]	Spurious correlations
Anomaly Detection	Mirai dataset [44]	Kitsune [44]	Out-of-distribution samples
OS Fingerprinting	CIC-IDS-2017 [54]	nPrintML [32]	Potential out-of-distribution samples
IoT Device Fingerprinting	UNSW-IoT [56]	Iisy [63]	Likely shortcut learning
Adaptive Bit-rate	HSDPA Norway [49]	Pensieve [42]	Potential out-of-distribution samples

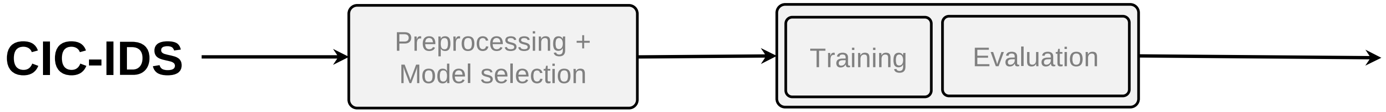
Most existing ML models fail to generalize; not ready for production deployments

How to Develop Generalizable ML Models for Networks?



Standard ML Pipeline

How to Develop Generalizable ML Models for Networks?



Is this the **right** data?

Standard ML Pipeline

How to Develop Generalizable ML Models for Networks?

Is this model **underspecified**?

F1-score: 0.99

CIC-IDS → Preprocessing + Model selection → Training | Evaluation →

Is this the **right** data?

Standard ML Pipeline

How to Develop Generalizable ML Models for Networks?

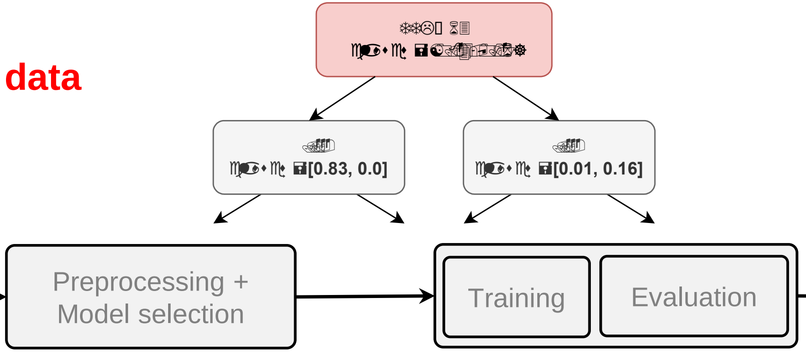
Learning shortcut

How to **collect better data** at scale?

Is this model **underspecified**?

F1-score: 0.99

CIC-IDS

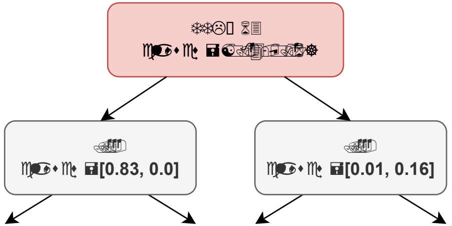


Is this the **right** data?

Standard ML Pipeline

How to Develop Generalizable ML Models for Networks?

Learning shortcut

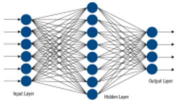


How to **collect better data** at scale?

Is this model **underspecified**?

F1-score: 0.99

CIC-IDS → Preprocessing + Model selection → Training | Evaluation →



Is this the **right** data?

Standard ML Pipeline

Answering these questions is critical for developing generalizable ML artifacts for networking

Progress in Past Years

Progress in Past Years

- **netUnicorn** [CCS’23]

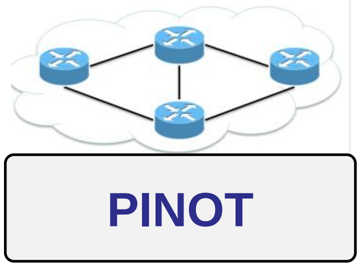
  Iteratively collect data for any problem and environment

- **Trustee** [CCS’22]

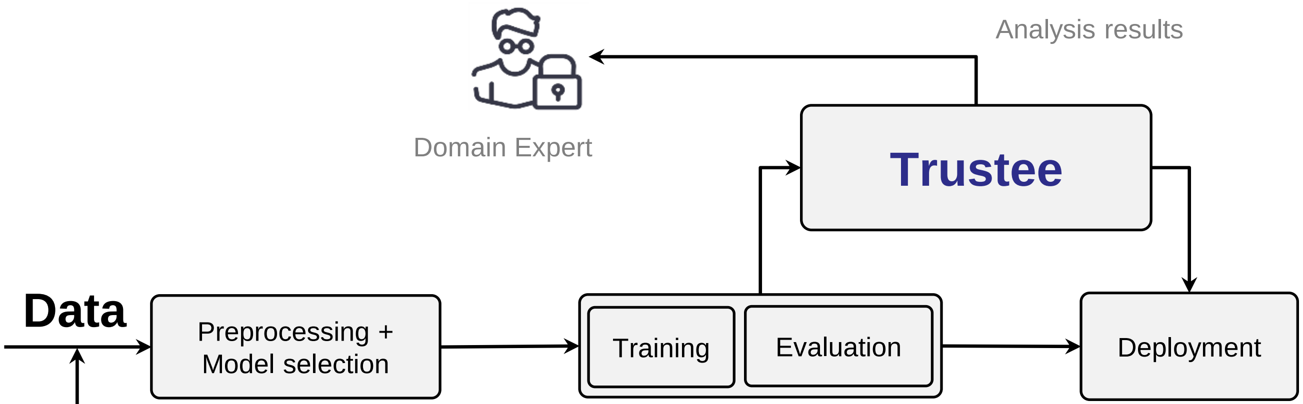
  Explain and analyze ML model’s decision making

- **PINOT** [ANRW’23]

  Transform your production network for data collection



Progress in Past Years



- **netUnicorn** **[CCS’23]**

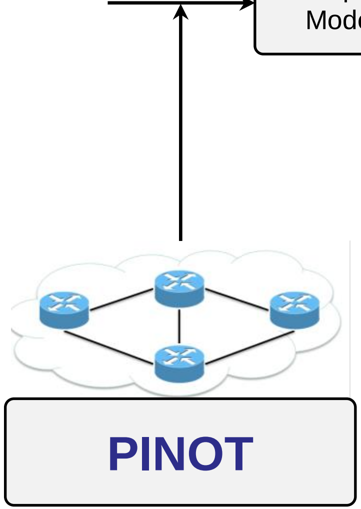
  Iteratively collect data for any problem and environment

- **Trustee** **[CCS’22]**

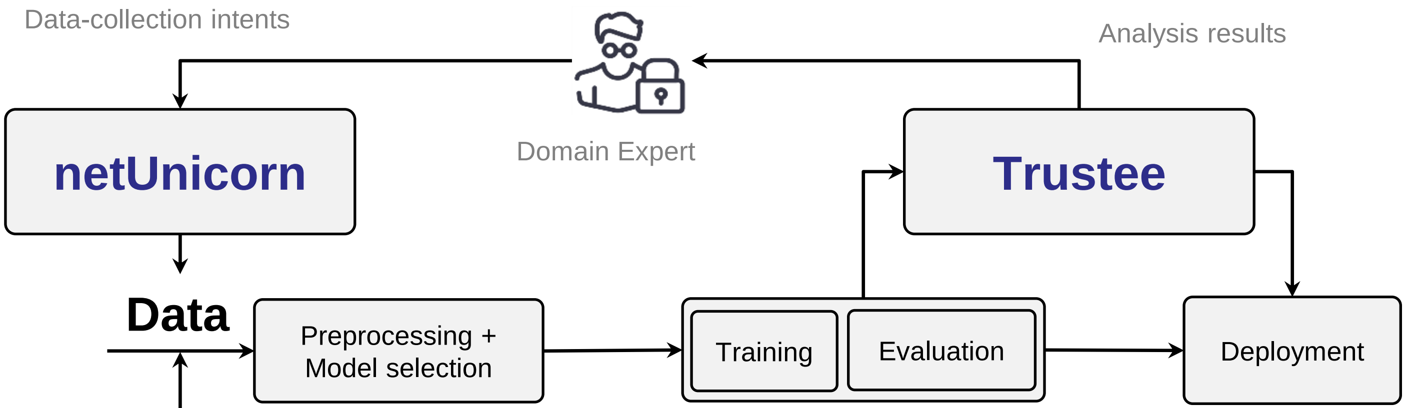
  Explain and analyze ML model’s decision making

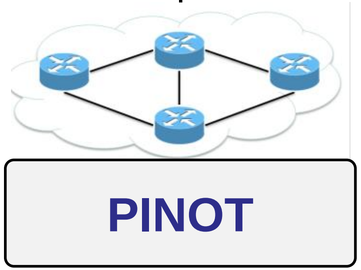
- **PINOT** **[ANRW’23]**

  Transform your production network for data collection



Progress in Past Years





- **netUnicorn [CCS’23]**

  Iteratively collect data for any problem and environment

- **Trustee [CCS’22]**

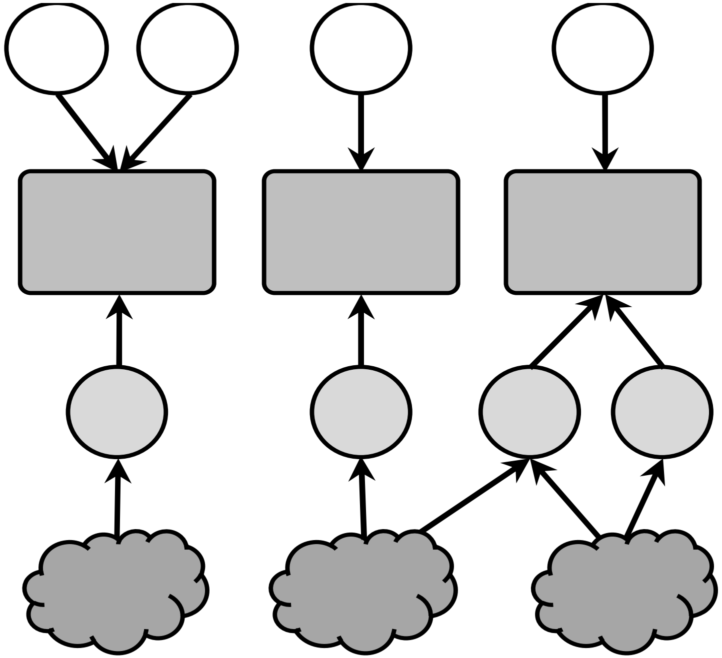
  Explain and analyze ML model’s decision making

- **PINOT [ANRW’23]**

  Transform your production network for data collection

netUnicorn: A Flexible Data Collection Platform

Fragmented



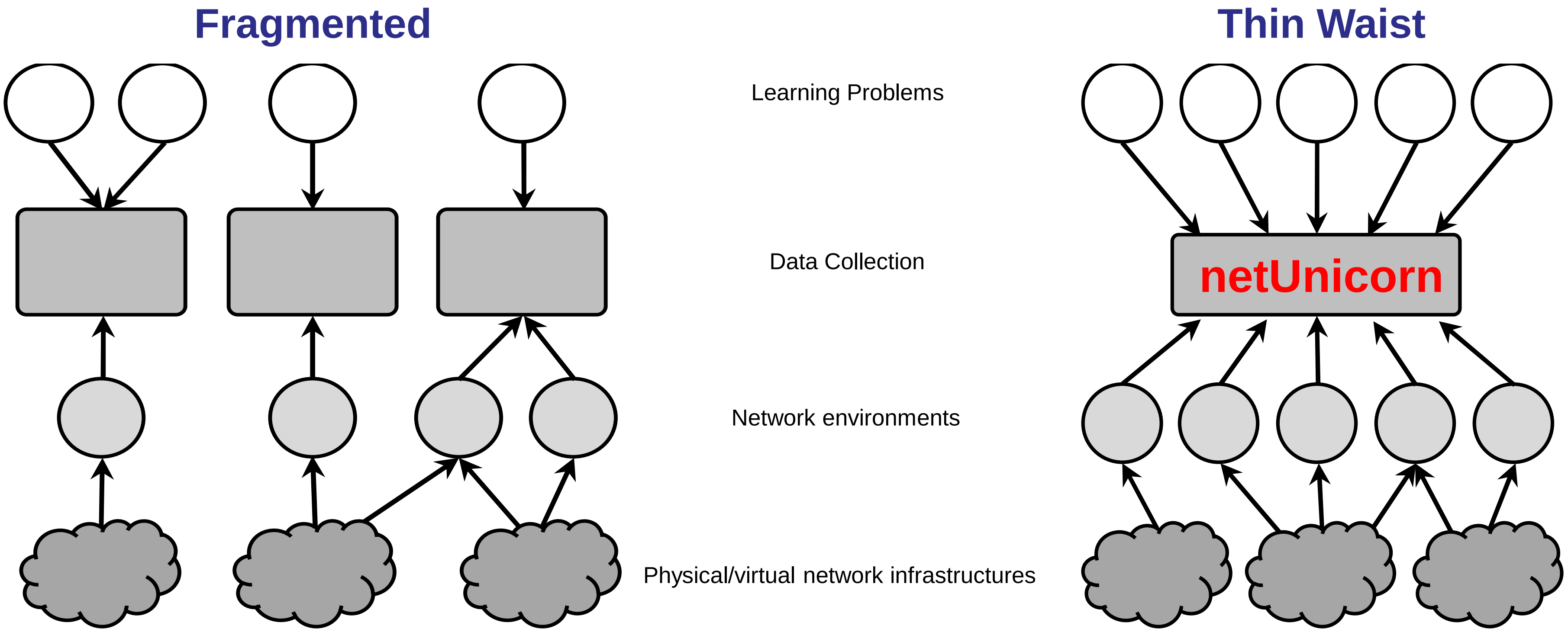
Learning Problems

Data Collection

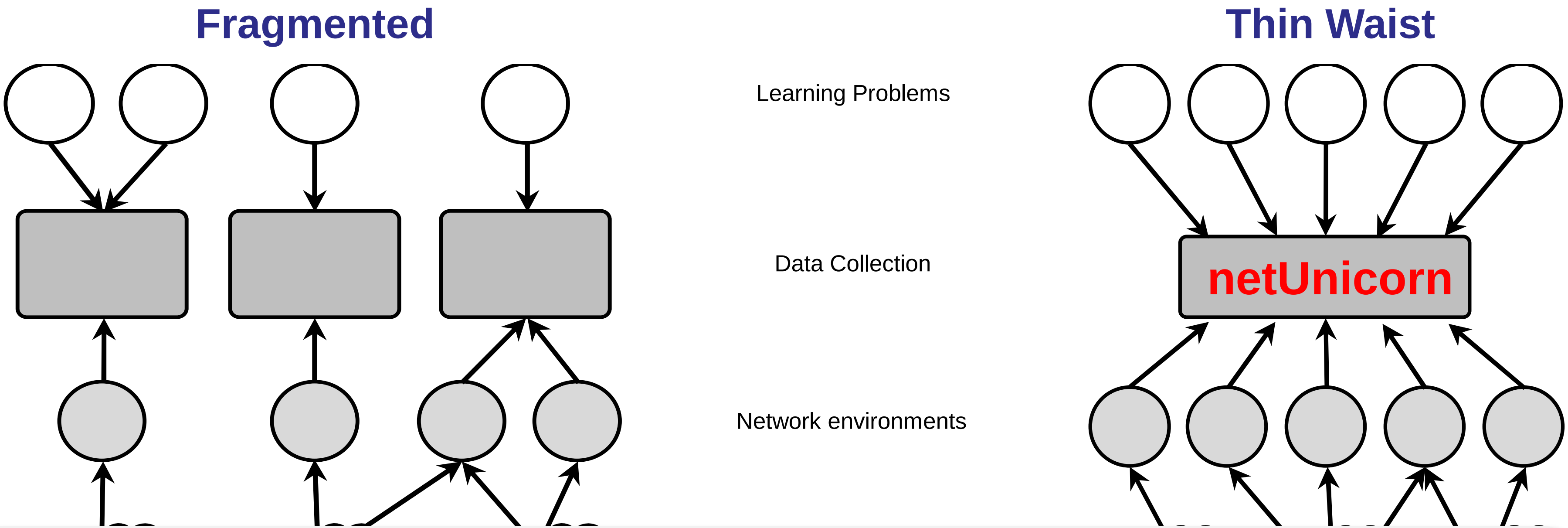
Network environments

Physical/virtual network infrastructures

netUnicorn: A Flexible Data Collection Platform

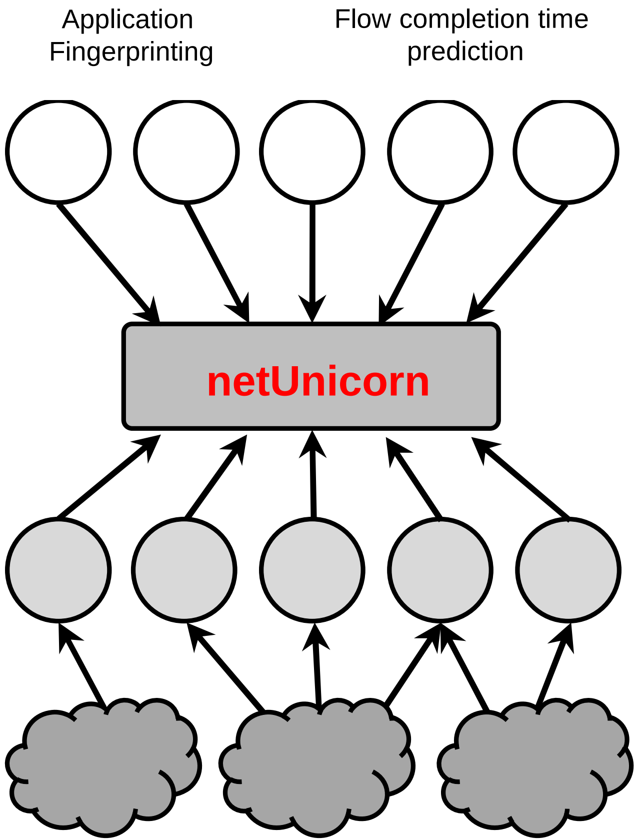


netUnicorn: A Flexible Data Collection Platform



Simplifies collecting data for any learning problem and target network environment

Limitation of netUnicorn



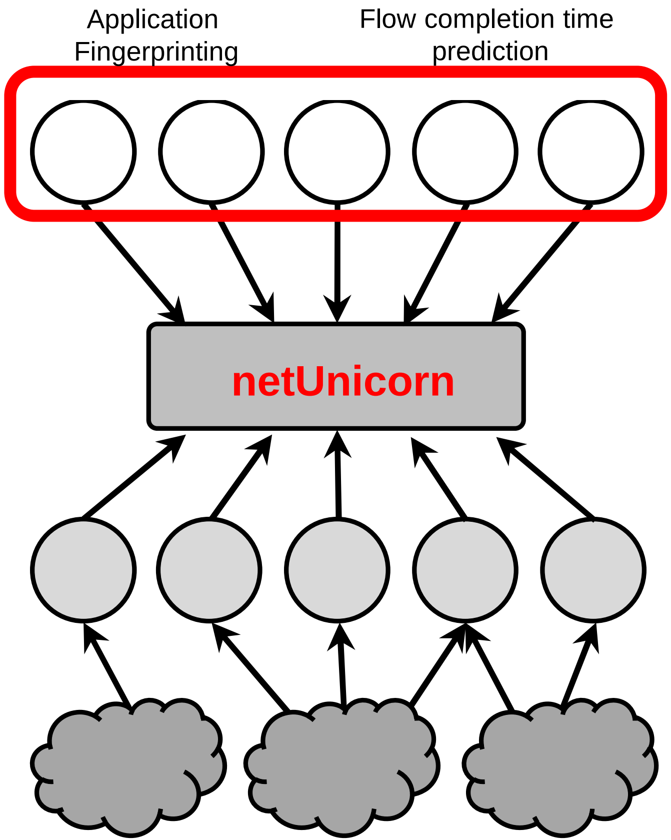
Learning Problems

Application Logic

Network environments

Physical/virtual Network infrastructures

Limitation of netUnicorn



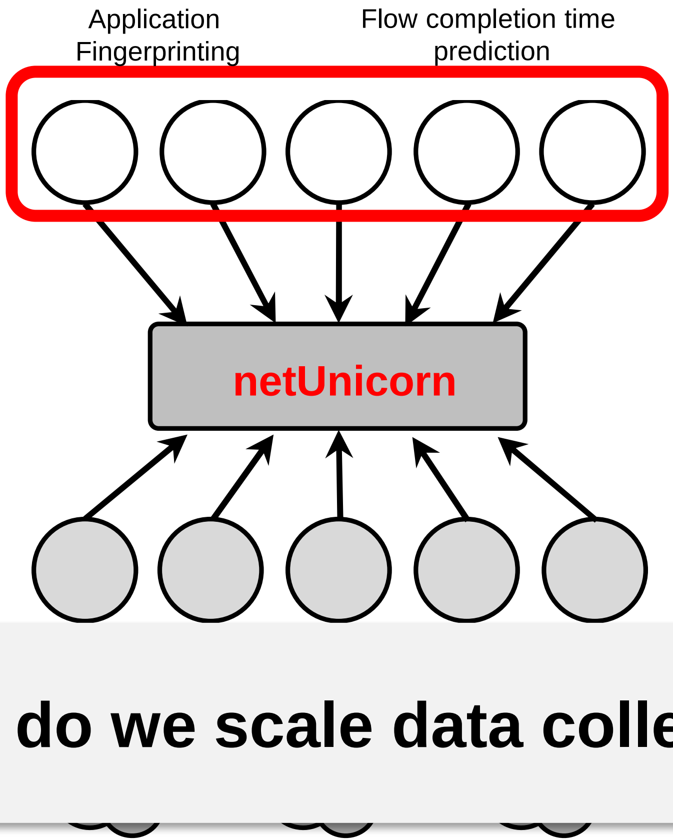
Writing application logic is manual effort

- Collecting data for new application is hard
- Easily breaks over time

Network environments

Physical/virtual Network infrastructures

Limitation of netUnicorn



Writing application logic is manual effort

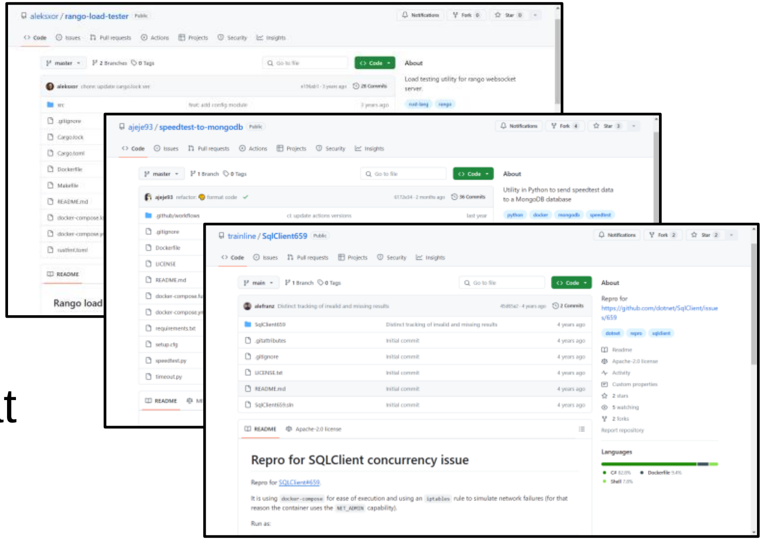
- Collecting data for new application is hard
- Easily breaks over time

Network environments

How do we scale data collection for new applications?

Opportunity: Publicly Accessible Code Repositories

- Millions of publicly accessible code repositories capture diverse application logic
  - GitHub, Bitbucket, etc.

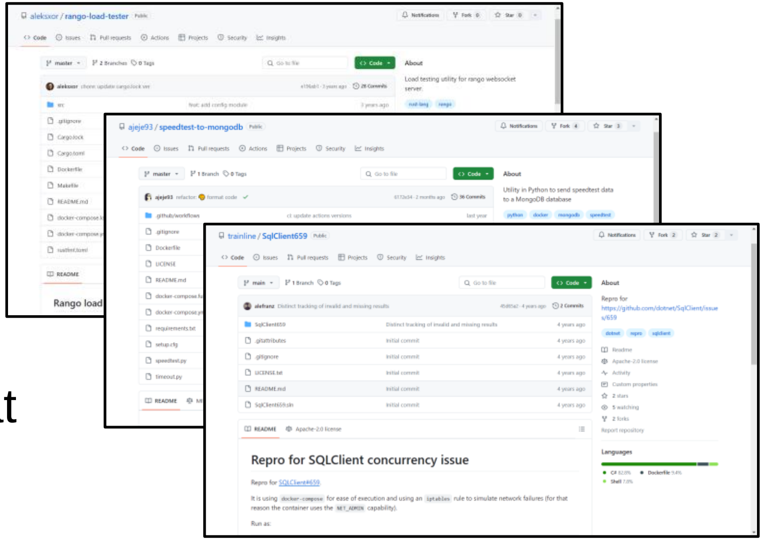


- Prior work showed around **70k GitHub repositories** with containerized applications that can generate diverse network traffic.

- We refer to these repositories as **Big Code**

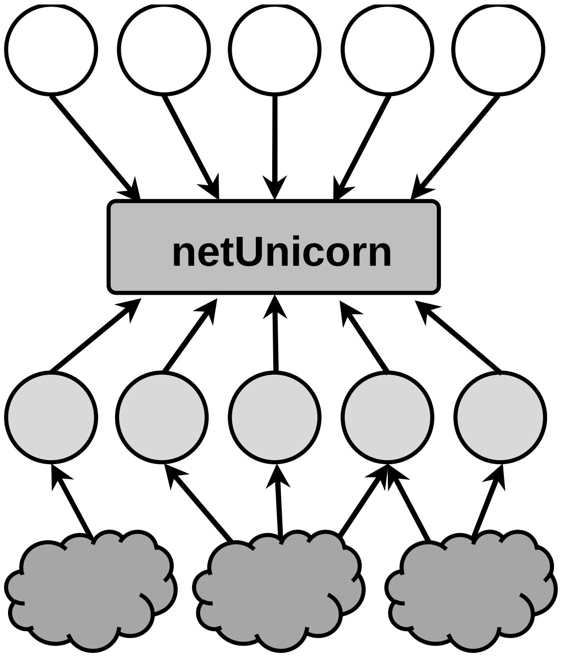
Opportunity: Publicly Accessible Code Repositories

- Millions of publicly accessible code repositories capture diverse application logic
  - GitHub, Bitbucket, etc.
- Prior work showed around **70k GitHub repositories** with containerized applications that can generate diverse network traffic.



Can we use Big Code to address netUnicorn's limitation?

Proposed Solution

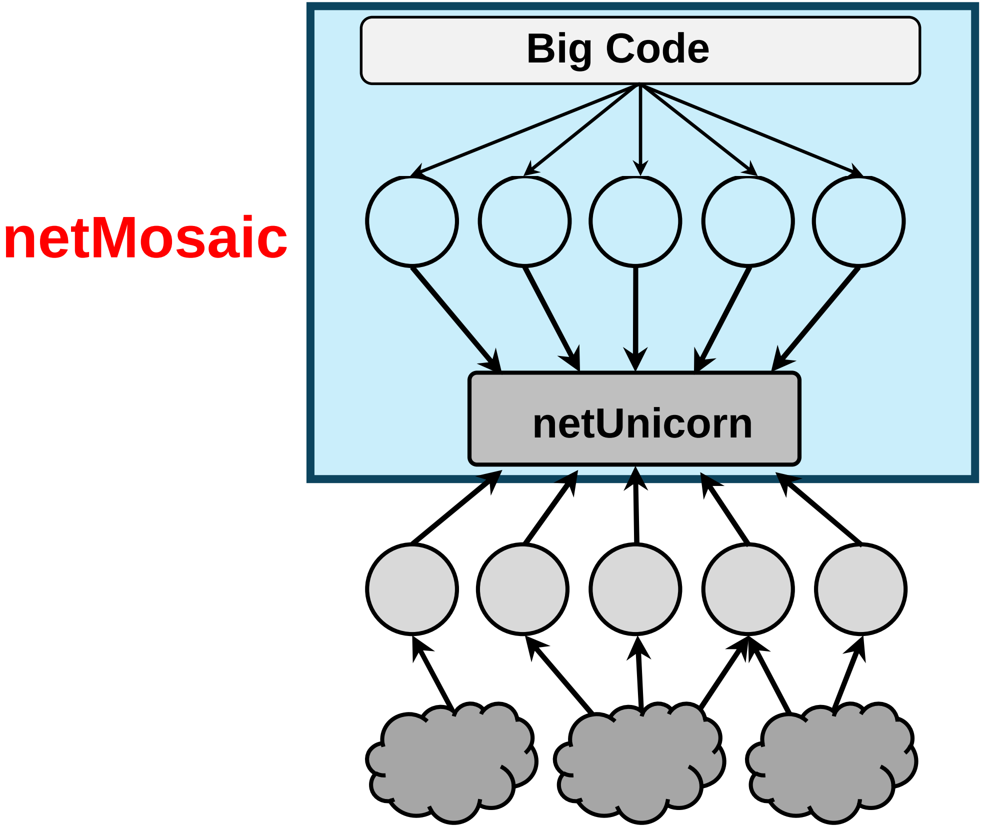


Learning Problems

Network environments

Physical/virtual Network infrastructures

Proposed Solution



Application Logic

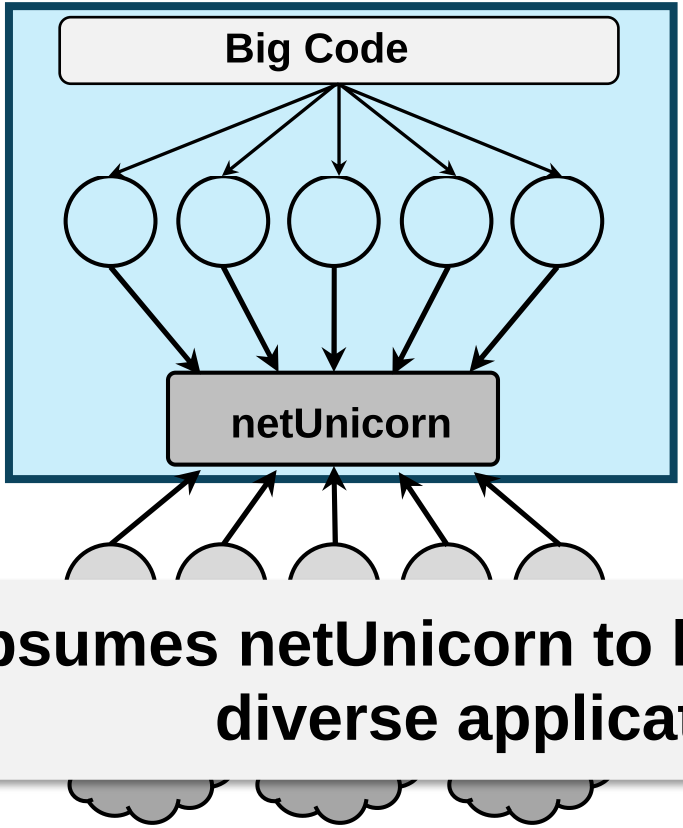
Learning Problems

Network environments

Physical/virtual Network infrastructures

Proposed Solution

netMosaic



Application Logic

Learning Problems

Subsumes netUnicorn to leverage Big Code's diverse application logic

Does it enable curating “better” datasets?

- **Learning problem**

  Traffic Classification: identify traffic classes based on encrypted packets in a flow

- **Data Source**
  - **16k GitHub repositories**
  - Labeled data using port numbers
- **Curated Dataset**
  - **1.7 million flows, 54 million packets, 264 unique services**
  - Top six services: HTTPS, Redis, PostgreSQL, Eforward, MongoDB, MySQL.

Does it enable curating “better” datasets?

- **Learning problem**

  Traffic Classification: identify traffic classes based on encrypted packets in a flow
- **Data Source**
  - **16k GitHub repositories**
  - Labeled data using port numbers
- **Curated Dataset**
  - **1.7 million flows, 54 million packets, 264 unique services**
  - Top six services: HTTPS, Redis, PostgreSQL, Eforward, MongoDB, MySQL.

	netMosaic	CrossMarkets	ISCXVPN2016
Number of Flows	**1.7 Million**	46,179	9,536

Does it enable curating “better” datasets?

- **Learning problem**

  Traffic Classification: identify traffic classes based on encrypted packets in a flow

- **Data Source**
  - **16k GitHub repositories**
  - Labeled data using port numbers

- **Curated Dataset**
  - **1.7 million flows, 54 million packets, 264 unique services**
  - Top six services: HTTPS, Redis, PostgreSQL, Eforward, MongoDB, MySQL.

netMosaic is able to curate “better” datasets,
i.e., more diverse and less sparse

Does it enable developing “generalizable” model?

- **Data Source**
  - 256 GitHub repositories

- **Datasets**
  - **Source Datasets: Labeled datasets used for model training**
    - Dataset A: Default setting → Model A
    - Dataset B: Low congestion setting → Model B
  - **Target Dataset: Unlabeled dataset used for assessing generalizability**
    - Dataset C: High-congestion setting

- **Learning Models**
  - **Random Forest**, Decision Trees, Logistic Regression, MLP

Results

Performance of models trained on Dataset A (Model A) and Dataset B (Model B) and tested on unseen Dataset C.

	Model A		Model B	
	Source Dataset	**Target Dataset**	Source Dataset	**Target Dataset**
Random Forest				
Decision Trees				
Logistic Regression				
MLP				

Results

Performance of models trained on Dataset A (Model A) and Dataset B (Model B) and tested on unseen Dataset C.

	Model A		Model B	
	Source Dataset	**Target Dataset**	Source Dataset	**Target Dataset**
Random Forest	0.83			
Decision Trees	0.81			
Logistic Regression	0.23			
MLP	0.76			

Results

Performance of models trained on Dataset A (Model A) and Dataset B (Model B) and tested on unseen Dataset C.

	Model A		Model B	
	Source Dataset	**Target Dataset**	Source Dataset	**Target Dataset**
Random Forest	0.83		0.81	
Decision Trees	0.81		0.80	
Logistic Regression	0.23		0.15	
MLP	0.76		0.73	

Results

Performance of models trained on Dataset A (Model A) and Dataset B (Model B) and tested on unseen Dataset C.

	Model A		Model B	
	Source Dataset	**Target Dataset**	Source Dataset	**Target Dataset**
Random Forest	0.83	**0.24**	0.81	
Decision Trees	0.81	**0.10**	0.80	
Logistic Regression	0.23	**0.06**	0.15	
MLP	0.76	**0.07**	0.73	

Results

Performance of models trained on Dataset A (Model A) and Dataset B (Model B) and tested on unseen Dataset C.

	Model A		Model B	
	Source Dataset	**Target Dataset**	Source Dataset	**Target Dataset**
Random Forest	0.83	**0.24**	0.81	**0.52**
Decision Trees	0.81	**0.10**	0.80	**0.28**
Logistic Regression	0.23	**0.06**	0.15	**0.14**
MLP	0.76	**0.07**	0.73	**0.37**

Results

Performance of models trained on Dataset A (Model A) and Dataset B (Model B) and tested on unseen Dataset C.

	Model A		Model B	
	Source Dataset	**Target Dataset**	Source Dataset	**Target Dataset**
Random Forest	0.83	**0.24**	0.81	**0.52**
Decision Trees	0.81	**0.10**	0.80	**0.28**
Logistic Regression	0.23	**0.06**	0.15	**0.14**
MLP	0.76	**0.07**	0.73	**0.37**

Using training data collected under more realistic network conditions could **improve model generalizability**

Summary and Outlook

- **Lessons learned**
  - Our system simplifies collecting data for disparate applications under different network conditions leveraging **Big Code** and **netUnicorn**
  - Prototype implementation demonstrates ability to curate **better datasets** and **generalizable ML models**

Summary and Outlook

- **Lessons learned**
  - Our system simplifies collecting data for disparate applications under different network conditions leveraging **Big Code** and **netUnicorn**
  - Prototype implementation demonstrates ability to curate **better datasets** and **generalizable ML models**

- **What's next?**
  - Leverage model explainability tools (e.g., Trustee)
  - Scale data collection for more repositories
  - Improve data quality: address class imbalance issues, filter noisy samples