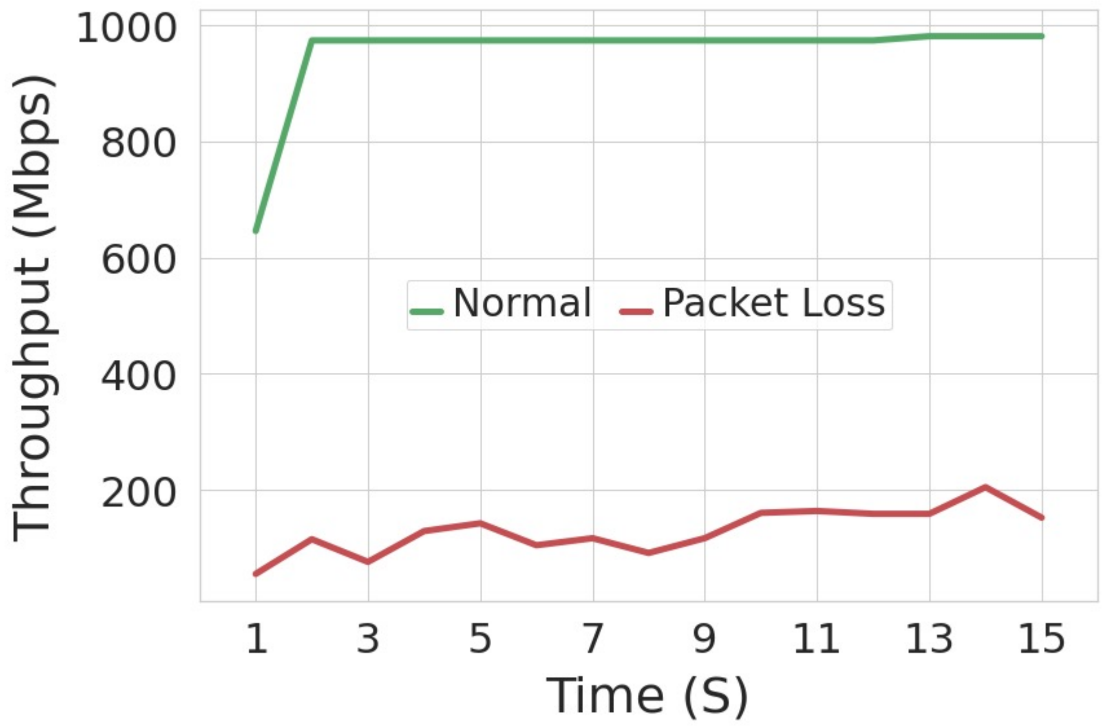
Learning Transfers via Transfer Learning

Md Arifuzzaman and Engin Arslan
University of Nevada, Reno

Performance Issues

- Data transfers often experience performance issues due to various reasons
  - *I/O interference, network congestion, packet loss, misconfigured end hosts, etc.*

- Performance issues can have severe impacts on application performance
  - *Packet loss of 0.1% leads to 5-7x decrease in throughput*
  - *Real-time streaming services affected negatively by increased network jitter*



Performance Issues

- Finding the root causes of anomalies is key to quickly recover from anomalies
- Different anomaly types can result in similar outcomes such as throughput decrease

Transfer Condition	Throughput
Normal	612 Mbps
Delay Jitter	285 Mbps
Packet Duplication	327 Mbps
Packet Reorder	428 Mbps
Packet Loss	154 Mbps
Packet Corruption	198 Mbps

Related Work: Anomaly Detection

- Statistical Model
  - Separate anomalous and normal transfers
  - Example: Change-point detections, principal component analysis
- Supervised Model
  - Binary classification models
- Unsupervised Model
  - Time-series analysis, clustering

Limitations

- No support for anomaly diagnosis
- Can be sensitive to data drift

Related Work: Anomaly Diagnosis

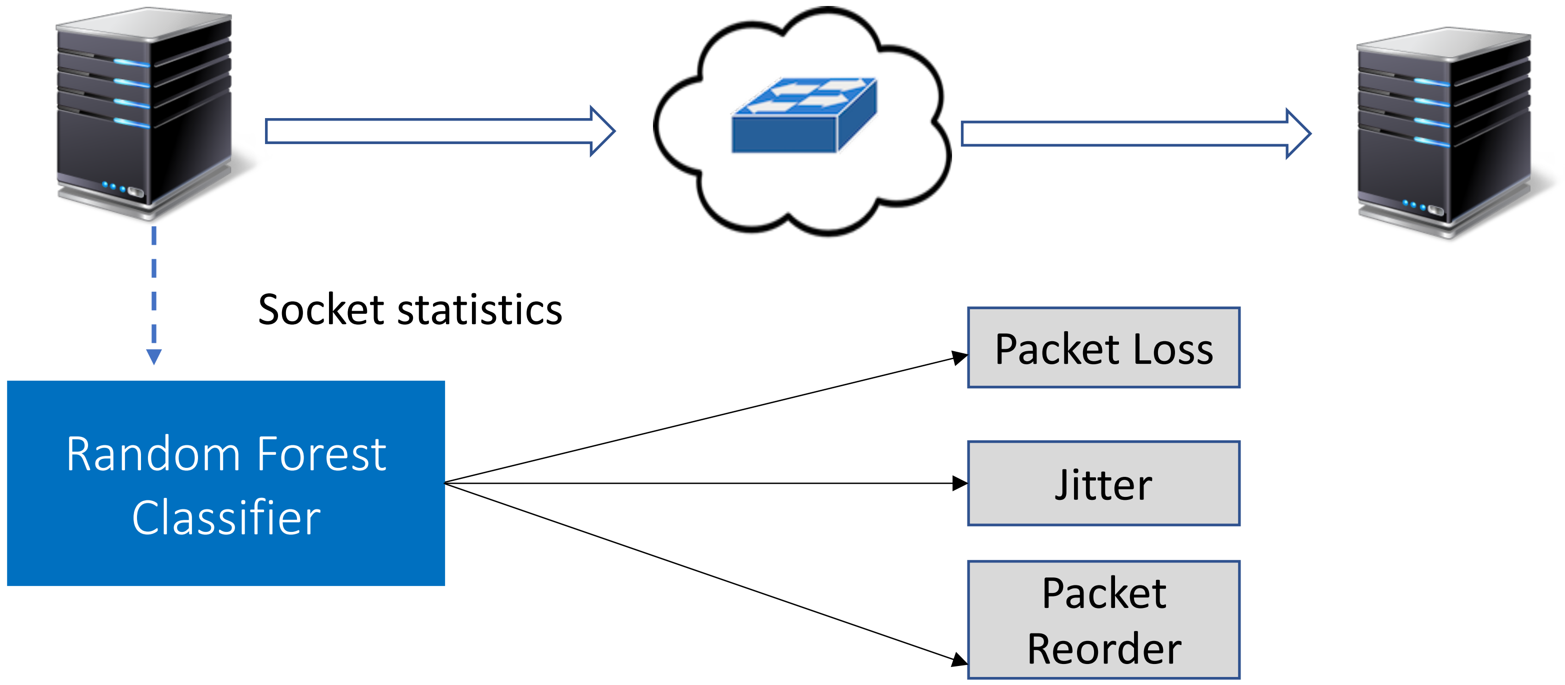
- Supervised Model
  - Multi-label classifications
  - Predict types of anomalies such as packet loss, reordering or duplication
- Heuristic Solution
  - Decision tree-like classification tree but developed heuristically

$\Rightarrow$

Limitations
- Sensitive to data drift
- Domain dependent

Proposed Solution

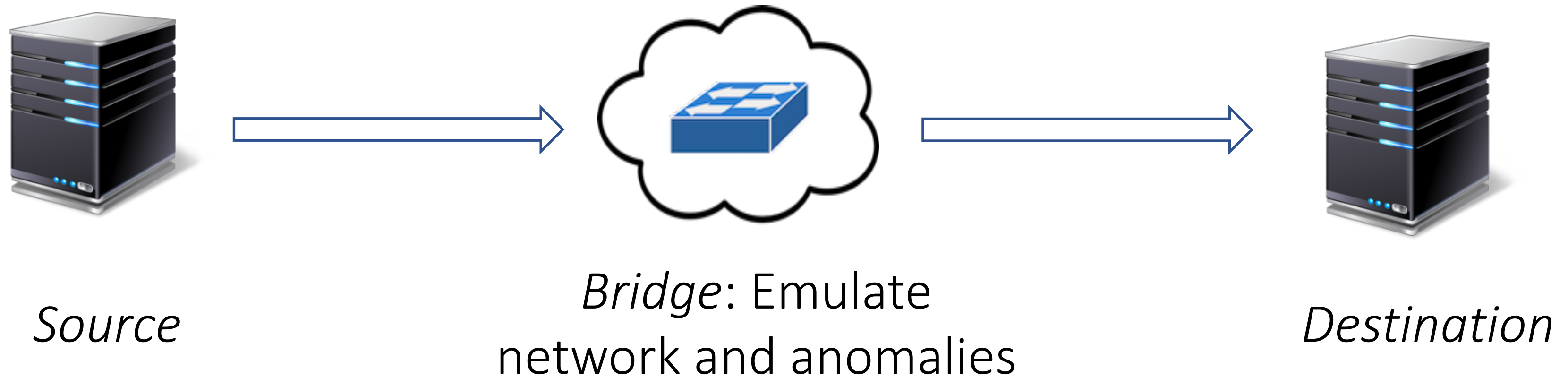
1. Train ML classifiers to predict the root causes of performance issues using socket statistics
2. Mitigate the need for labeled data collection in each network by eliminating/transforming domain dependent features



Data Collection and Modeling

Data Collection

- Created 9 networks with different bandwidth and delay settings in Emulab
  - e.g.,100Mbps and 100ms, 5000 Mbps and 30ms
- Five anomaly types reproduced as *loss, corruption, reorder, duplicate,* and *jitter using Linux traffic control module (tc)*
- Executed 600 sample transfers per anomaly type in each network
- Used **ss** command to capture socket statistics



Socket Statistics (*ss*)

- utility tool to investigate sockets
- Reports *30+* flow-levels metrics
- Metrics are instantly available
- Collected metrics at 10th second to wait for transfer performance to stabilize

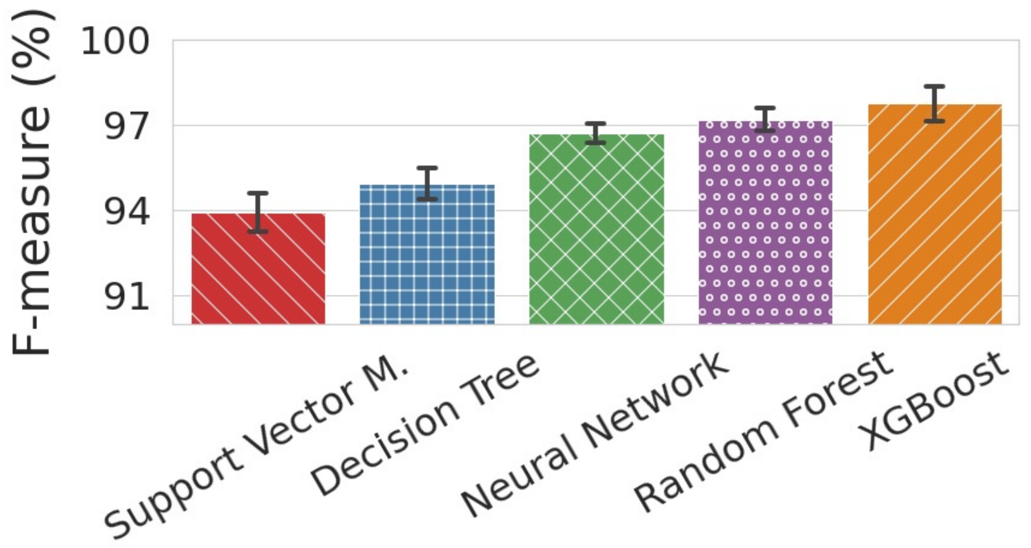
Feature Engineering

- Eliminate redundant features
  - Reduce training cost and improve accuracy

- Selected features with combined 95% importance score
  - Ended up with *nine* features

Model Evaluation (Emulab)

- Trained separate ML models for each network
- Hyperparameters are tuned with *Scikit-Optimize* library
- F-measure used as the performance metric

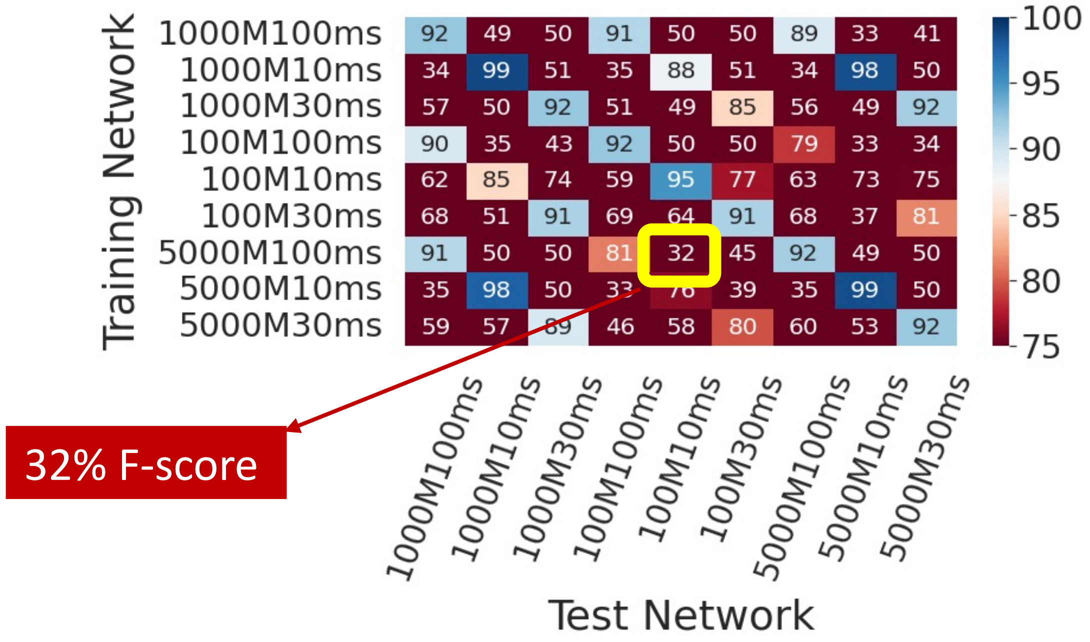


Transfer Learning

Separate Random Forest classifier trained for each network

Each classifier is evaluated in other networks

Train custom models for each network?



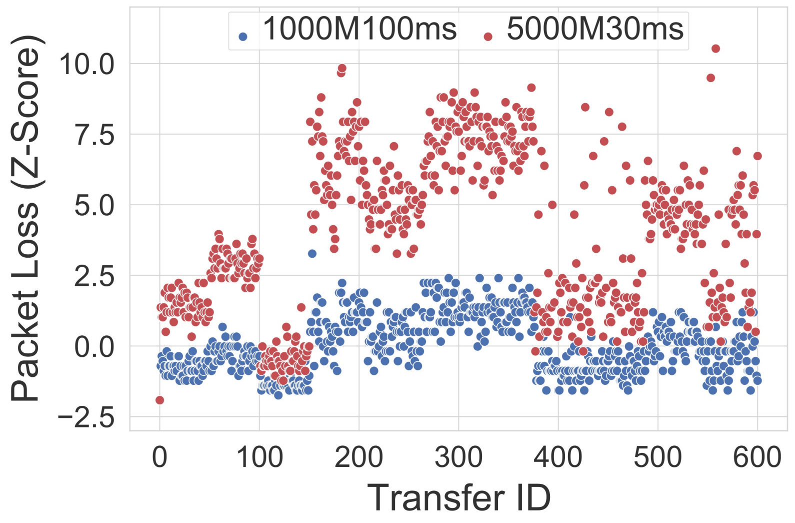
Training Network \ Test Network	1000M100ms	1000M10ms	1000M30ms	100M100ms	100M10ms	100M30ms	5000M100ms	5000M10ms	5000M30ms
1000M100ms	92	49	50	91	50	50	89	33	41
1000M10ms	34	99	51	35	88	51	34	98	50
1000M30ms	57	50	92	51	49	85	56	49	92
100M100ms	90	35	43	92	50	50	79	33	34
100M10ms	62	85	74	59	95	77	63	73	75
100M30ms	68	51	91	69	64	91	68	37	81
5000M100ms	91	50	50	81	32	45	92	49	50
5000M10ms	35	98	50	33	76	39	35	99	50
5000M30ms	59	57	89	46	58	80	60	53	92

ML models fail to achieve good performance when transferred to different networks.

Transfer Learning – Proposed Solution

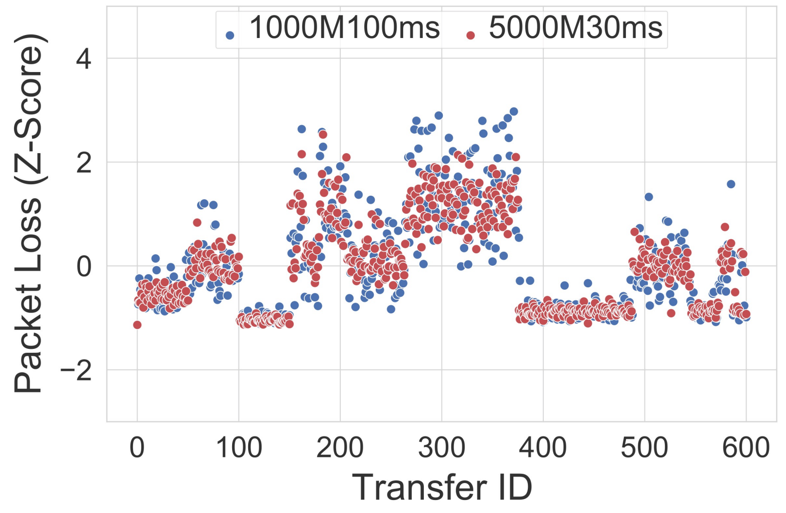
- Standard normalization results in different distributions in transfer learning

$$Z = \frac{x - \mu}{\sigma}$$

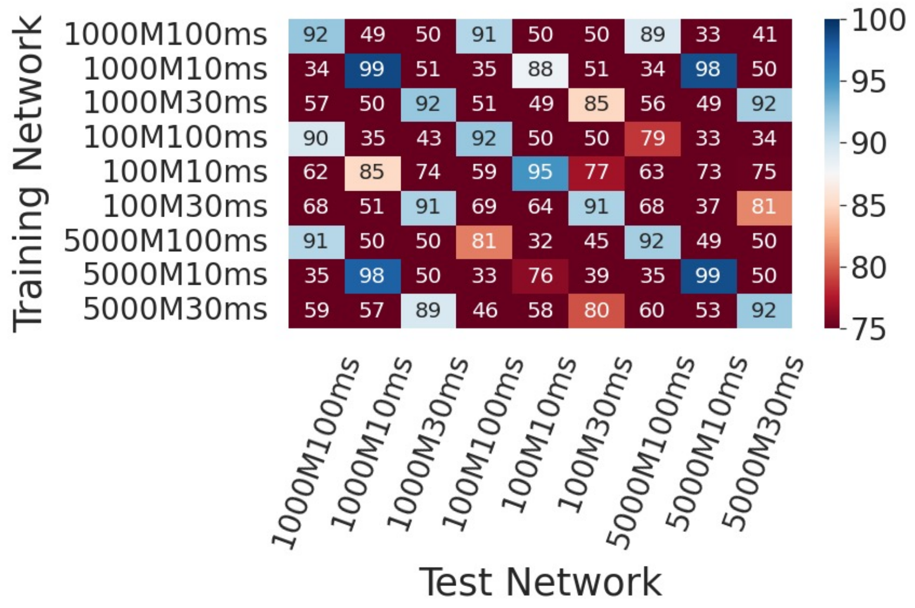


Feature Transformation: Remove network dependency from collected metrics.

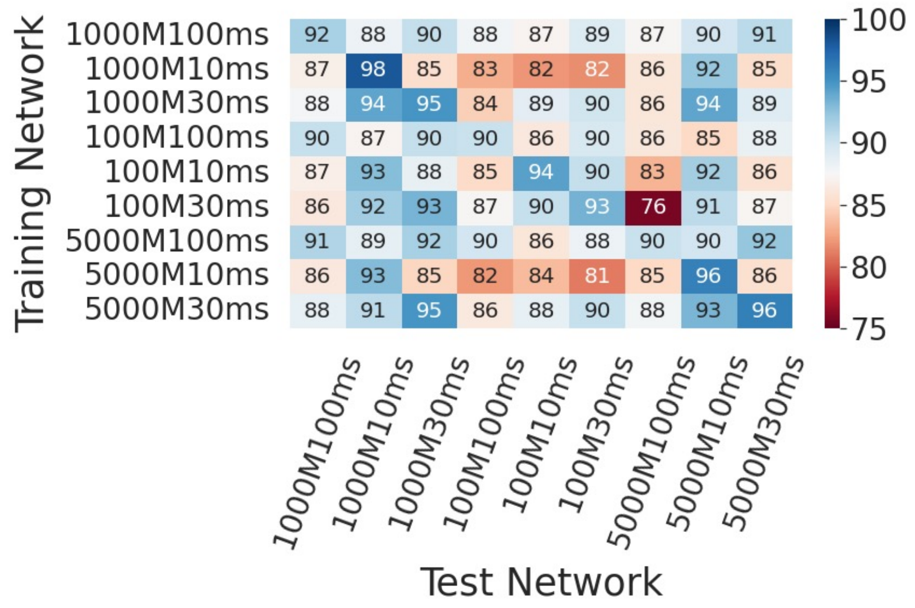
- E.g., divide retransmitted packet count to total packets to convert it to the ratio.



Transfer Learning – Proposed Solution



Training Network \ Test Network	1000M100ms	1000M10ms	1000M30ms	100M100ms	100M10ms	100M30ms	5000M100ms	5000M10ms	5000M30ms
1000M100ms	92	49	50	91	50	50	89	33	41
1000M10ms	34	99	51	35	88	51	34	98	50
1000M30ms	57	50	92	51	49	85	56	49	92
100M100ms	90	35	43	92	50	50	79	33	34
100M10ms	62	85	74	59	95	77	63	73	75
100M30ms	68	51	91	69	64	91	68	37	81
5000M100ms	91	50	50	81	32	45	92	49	50
5000M10ms	35	98	50	33	76	39	35	99	50
5000M30ms	59	57	89	46	58	80	60	53	92



Training Network \ Test Network	1000M100ms	1000M10ms	1000M30ms	100M100ms	100M10ms	100M30ms	5000M100ms	5000M10ms	5000M30ms
1000M100ms	92	88	90	88	87	89	87	90	91
1000M10ms	87	98	85	83	82	82	86	92	85
1000M30ms	88	94	95	84	89	90	86	94	89
100M100ms	90	87	90	90	86	90	86	85	88
100M10ms	87	93	88	85	94	90	83	92	86
100M30ms	86	92	93	87	90	93	76	91	87
5000M100ms	91	89	92	90	86	88	90	90	92
5000M10ms	86	93	85	82	84	81	85	96	86
5000M30ms	88	91	95	86	88	90	88	93	96

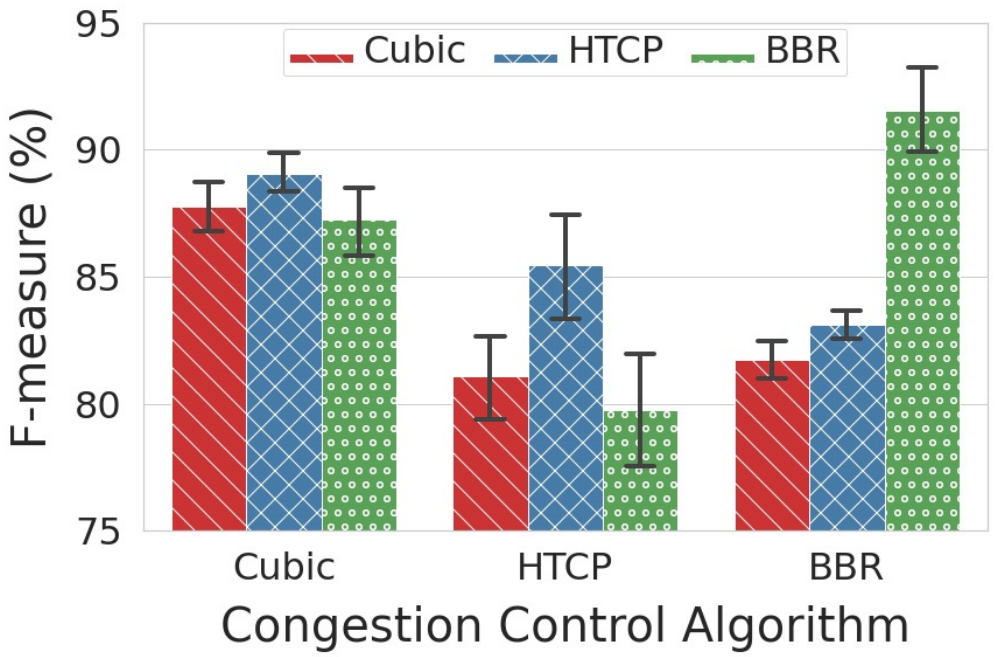
Significant improvement in transfer learning performance. From less than 60% to over 85% in Emulab testbeds

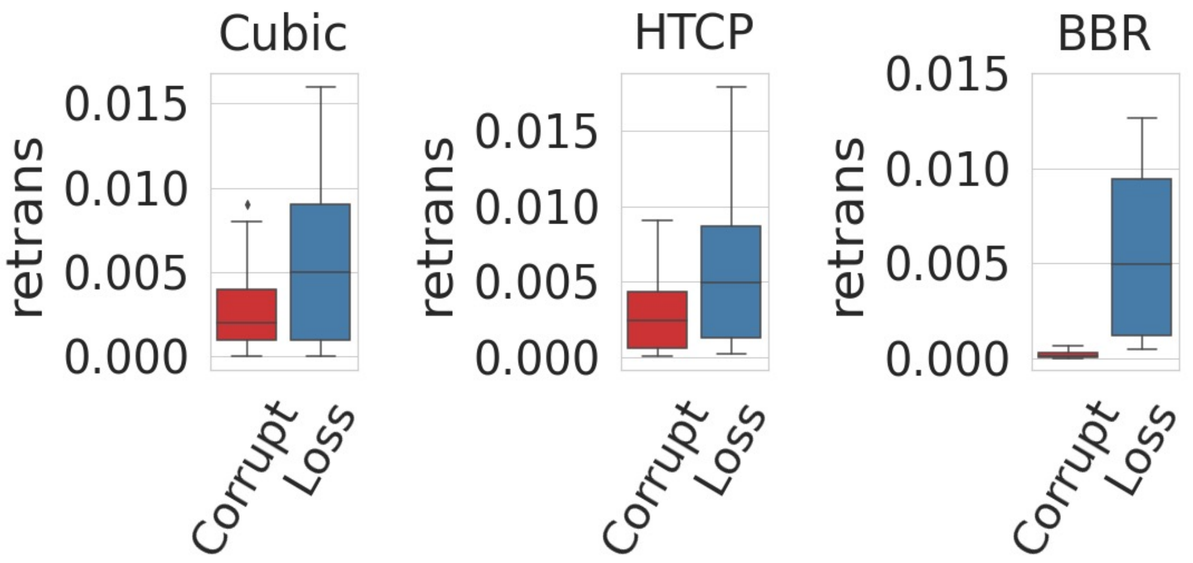
Evaluation in Production Networks

Train Dataset	Test Dataset	F-measure (%)
Emulab (Salt Lake City)	HPCLab (Reno)	81.9 ± 2.8
HPCLab (Reno)	Emulab (Salt Lake City)	81.1 ± 3.7
Austin-Chicago	Austin-San Diego	81.2 ± 1.6
Austin-San Diego	Austin-Chicago	79.8 ± 1.8

* Please see networks specifications in the paper

Impact of Congestion Control Algorithms



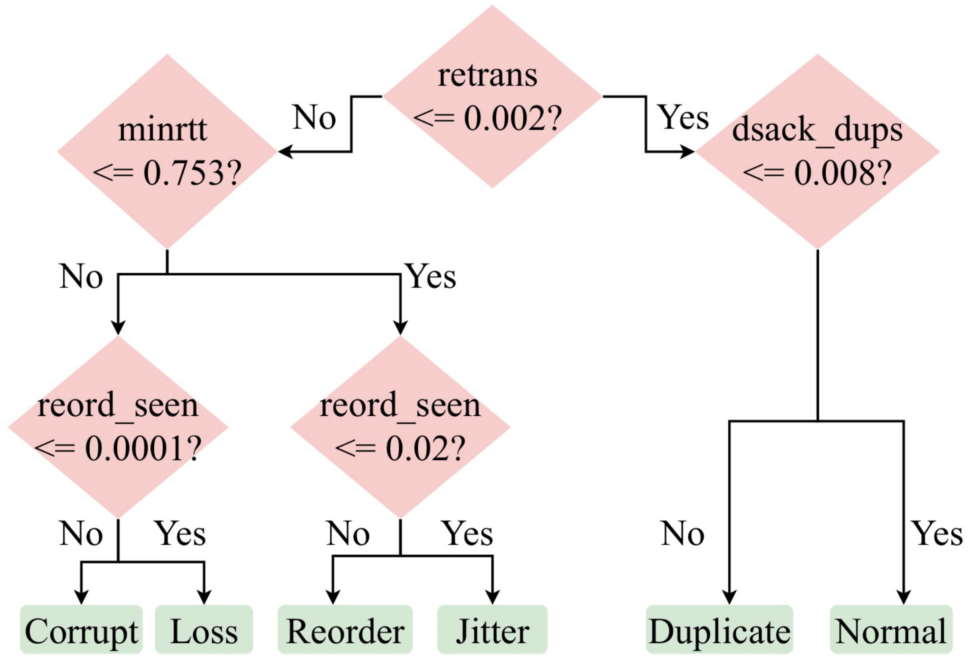


BBR significantly outperforms Cubic and HTCP

BBR has more distinguishable feature values

Explainable Models

- Random Forest or XGBoost models consist of hundreds of decision trees
- Debugging is severely challenging
- The Decision tree method with the original seven features still build very complicated trees (as many as 19 depths)
- Eliminated least important 3 features
- Performance drop 2-5%, but makes model interpretable



Decision Tree using only four features

Summary

ML models perform well diagnosing anomalies but require labeled data collection for each network due to poor transfer learning performance

Feature transformation eliminates domain dependency and significantly improves transfer learning performance

BBR based model perform better compared to other congestion control algorithms

Thanks!

This project is funded by National Science Foundation under award number 2007789