Sparse Control and Data plane Telemetry features for BGP anomaly detection

Jose Cordova-Garcia
ESPOL University, Ecuador

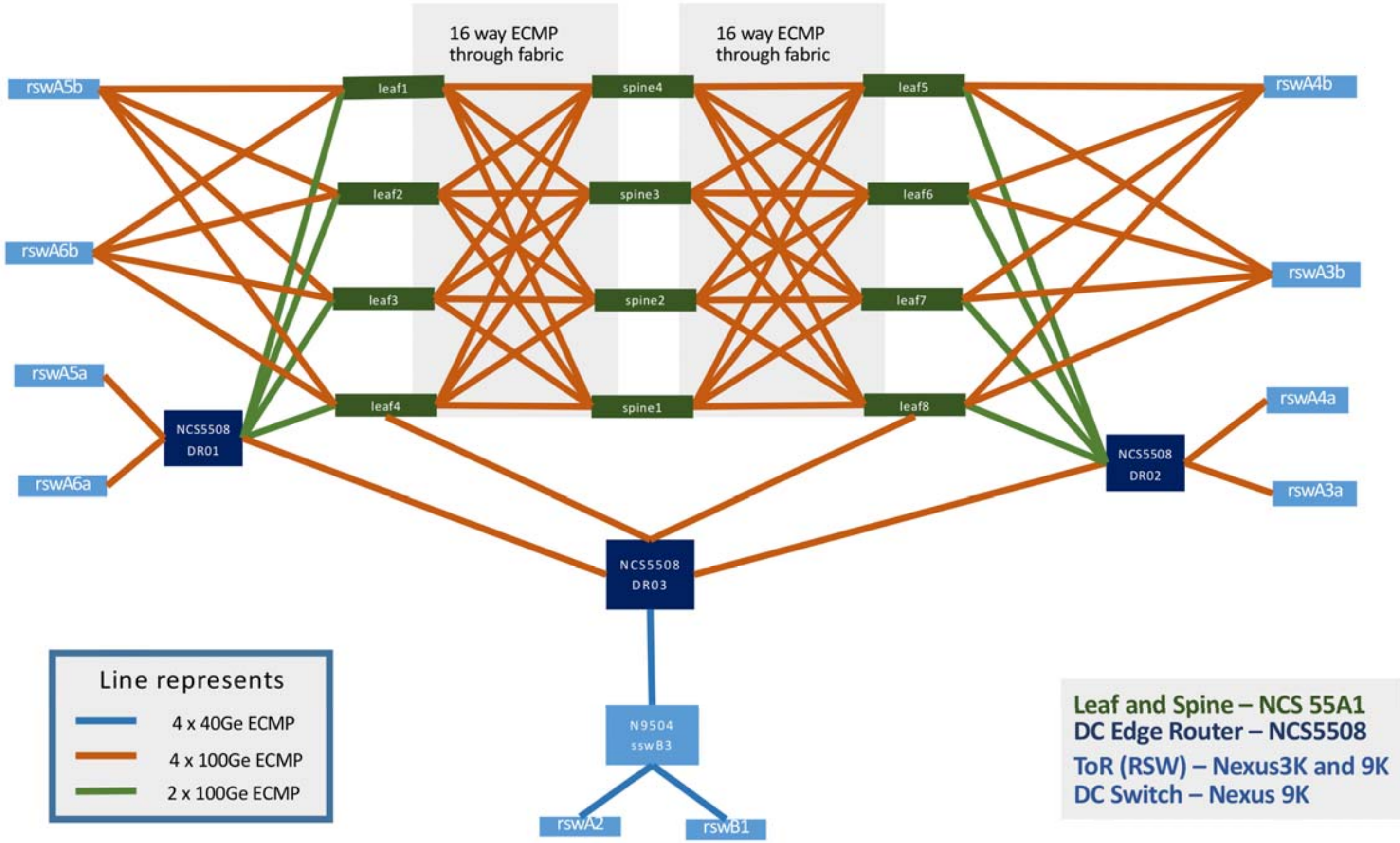
Motivation: Anomaly detection

- Timely detection of network failures is crucial to support daily operations.
- Core (e.g. a data center) networks should rely on automatized detection.
- Problems?
  - Detection based on monitoring features, **which features?**
  - High-resolution/high-dimensionality monitoring data become available, **which nodes should provide data?**
  - Operators favor visual inspection, **can we work with raw features?**

- Traditional monitoring:
  - Visual inspection at NOCs
  - CLI, scripts, active polling
    - Polling methods: SNMP

A motivating network

- Data center network running BGP (DCN)
- Network anomalies
  - Network topology changes
  - Policy changes
  - Misconfigurations
  - Attacks
  - **Device failure**



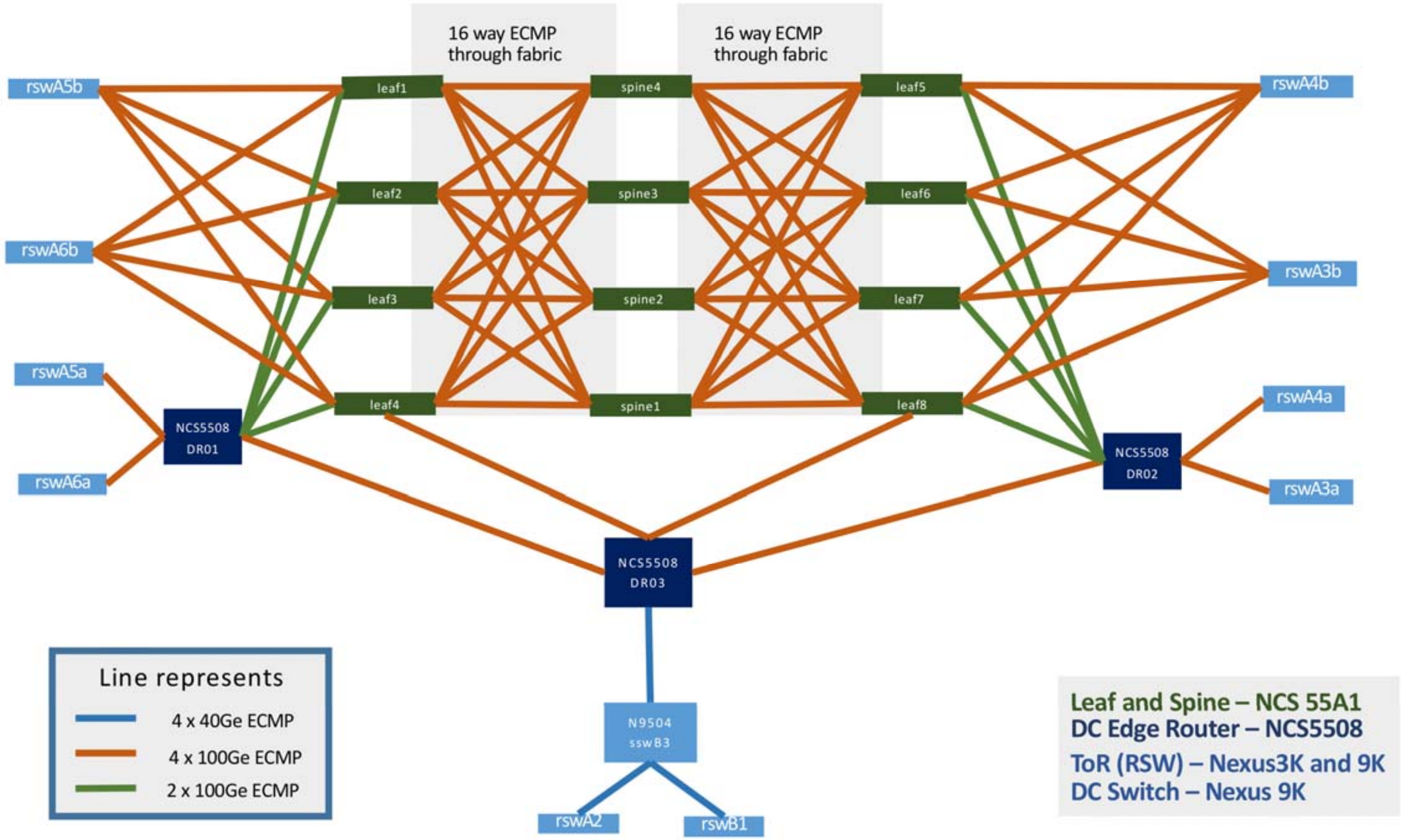
https://github.com/cisco-ie/telemetry/

How to detect anomalies

Work	Data Collection	Features used	Detection Model
Huang'07	10 minutes	1 feature	PCA
Deshpande'09	5 minutes	4 features	Normality test
Al-Rousan'12	1 minute	17 features, select 10	Naive Bayes
Al-Roustan'12	1 minute	37 features, select 10	SVMs, HMM
Ding'16	1 minute,bin	37 features	LSTM
Cheng'16'19	1 minute, bin	33 features	Multistage-LSTM
Nguyen'19	3 minute bins	55 features	VAE

What about a Telemetry network?

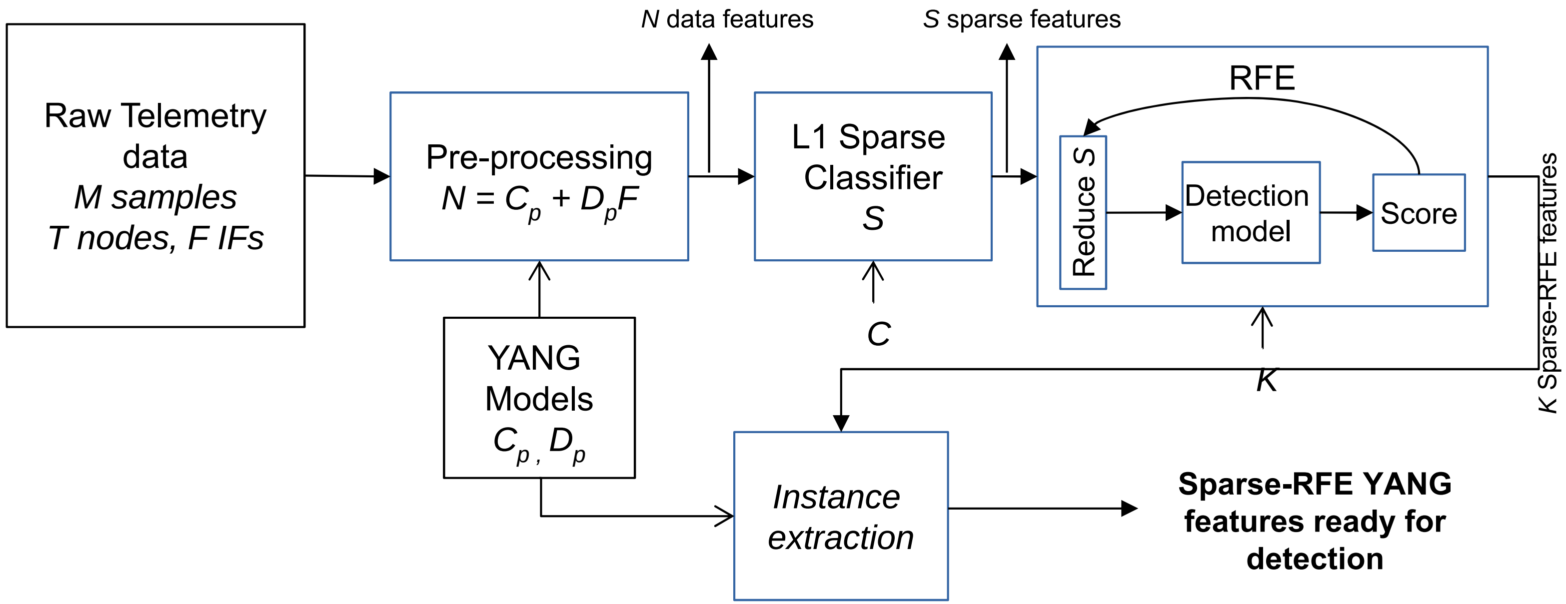
- A Telemetry-enabled network:
  - Nodes automatically stream their status data
  - Seconds' resolution
  - Data available is pre-configured[1]
- When a node fails, BGP will say something!
  - Referred works apply
- What would Telemetry say?
  - **Which features?**
  - **From where?**
  - **Transform?**



https://github.com/cisco-ie/telemetry/

Proposed Scheme

- Most common:
  - Separate FE from Learning Model
  - **UPDATE** features inherited from literature
- Now, Telemetry provides more features
  - Proposal: Integration



Telemetry data

- More than BGP **UPDATE**s
- Vendor provides data based on YANG models
- Model: encoded tree
  - Data/Control plane trees available.
  - Leaf: ***YANG feature***
- Operator configures *Model Driven Telemetry (MDT)* © at Telemetry nodes
  - Each node provides its own instances of YANG features

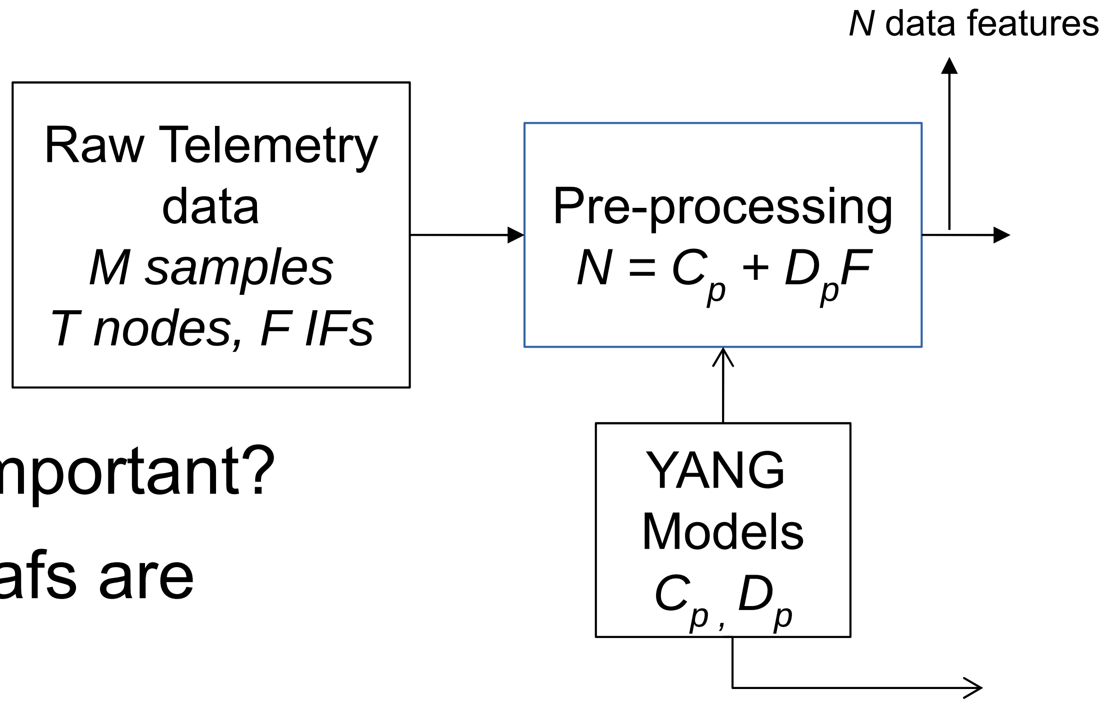
In the network: **T**elemetry nodes=15, 12 fabric, 3 edge

- IFs: leaf/spine 34, edge 74

Raw Telemetry data
M samples
T nodes, F IFs

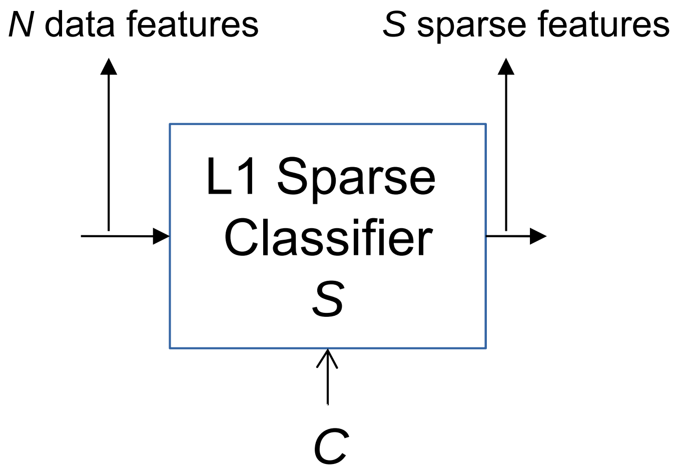
Telemetry data

- Nodes provide *N* **data features** to the detection model
- Network-wide monitoring:
  - Collect N=17000+ features
  - Which data features are important?
  - Moreover, which YANG leafs are important?



Stage 1: Sparse features

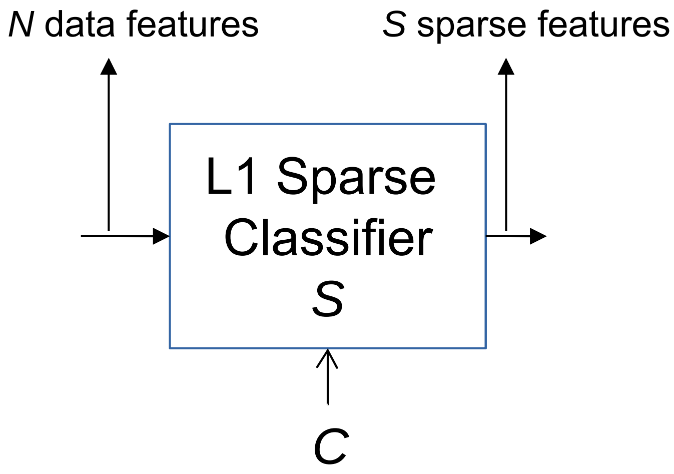
- Let the machine decide what is important for detection
  - Start with data features (node-level instances of YANG leafs)
  - Sparse classifier for feature selection:
    - Lasso, reg. SVM, Coordinate Lasso
  - *S* **sparse features**

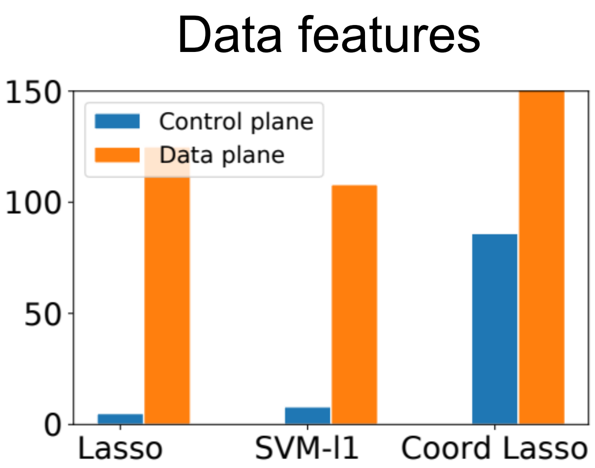


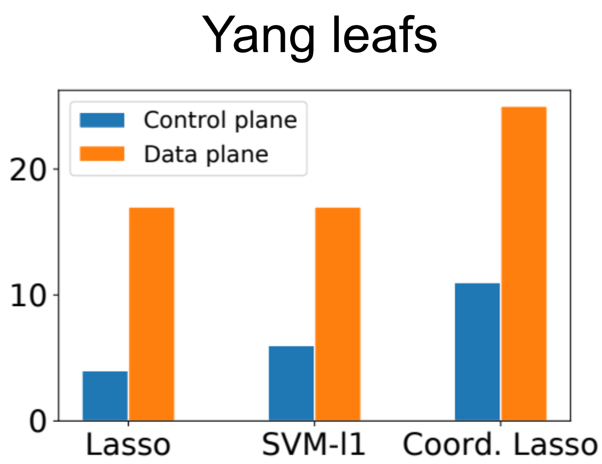
Stage 1: Sparse features

- ***S* sparse features**

S	Model	Accuracy	Precision	Recall
130	Lasso	94.65%	94.11%	93.20%
116	SVM-*l1*	95.06%	94.17%	94.17%
1945	Coord. Lasso	95.06%	96.90%	91.26%

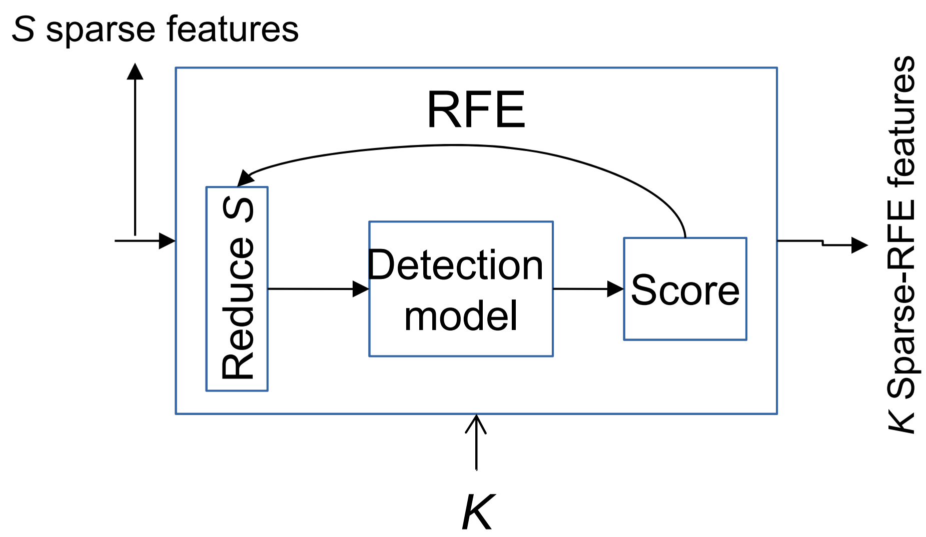






Stage 2: Sparse-RFE

- Push forward → Let the machine decide which important features contribute to better detection
  - Embed model and performance into feature selection
  - Select *K* **Sparse-RFE features**

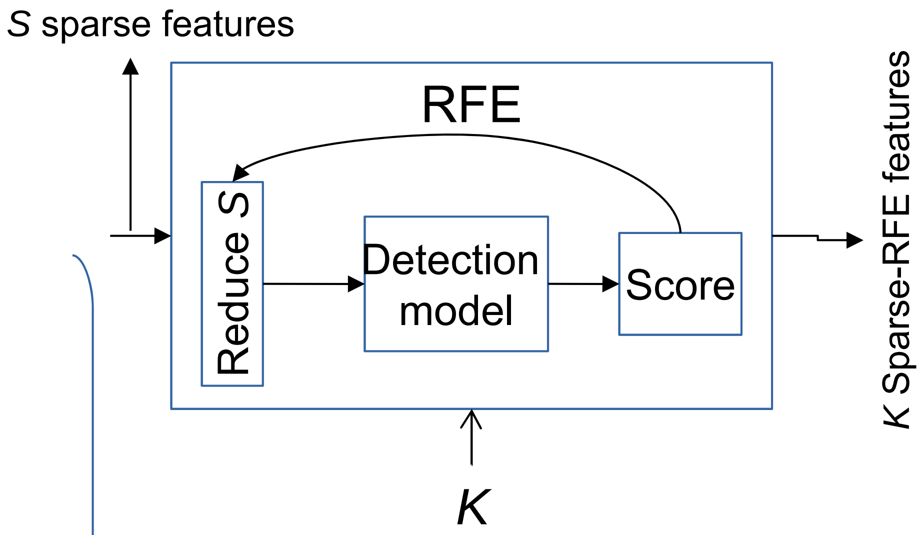


Stage 2: Sparse-RFE

- *K* **Sparse-RFE features**
  - (data features)

K	Model	Accuracy	Precision	Recall
10	Lasso	94.23%	94.95%	91.26%
	SVM-$l1$	91.35%	88.67%	91.26%
	Coord. Lasso	90.12%	96.47%	79.61%
20	Lasso	94.65%	93.26%	94.17%
	SVM-$l1$	95.35%	95.04%	93.20%
	Coord. Lasso	95.47%	96.93%	92.23%
30	Lasso	94.65%	95.00%	92.23%
	SVM-$l1$	94.65%	93.26%	94.17%
	Coord. Lasso	96.29%	95.19%	96.11%
40	Lasso	95.47%	96.00%	93.20%
	SVM-$l1$	95.88%	94.28%	96.11%
	Coord. Lasso	96.29%	95.19%	96.11%
52[a]	Lasso	93.82%	93.13%	92.23%
97[a]	SVM-$l1$	94.65%	94.11%	93.20%
246[a]	Coord. Lasso	96.70%	97.02%	95.14%

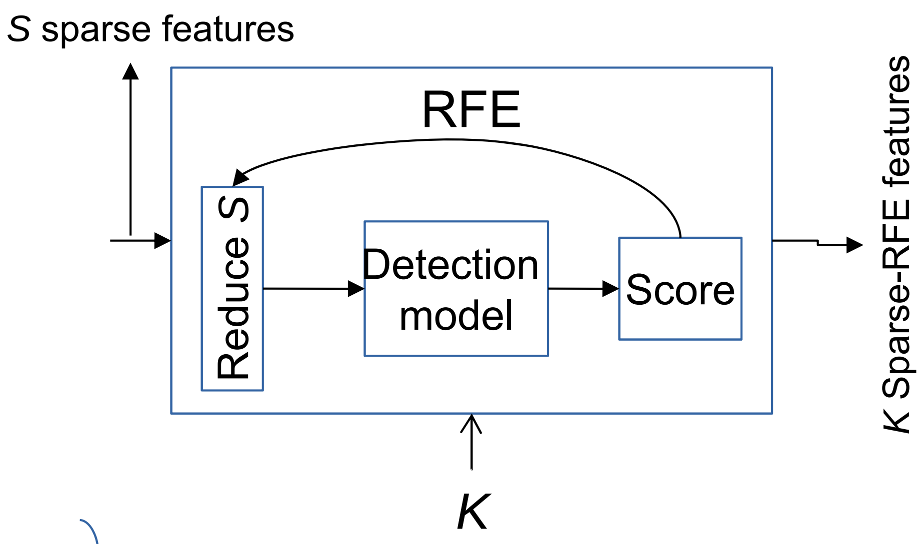
[a]Selected through crossvalidation.

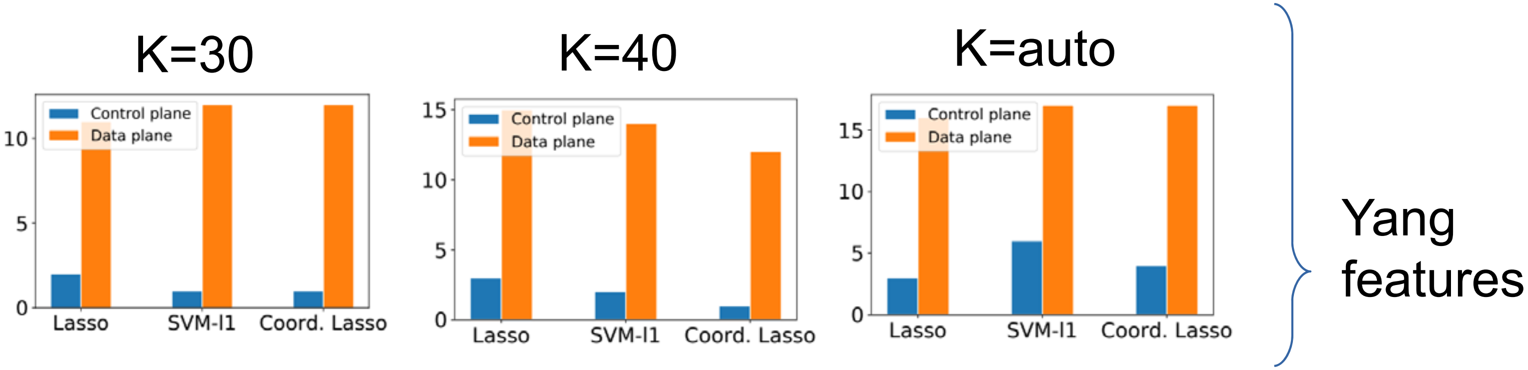


Performance doesn't degrade greatly

Stage 2: Sparse-RFE

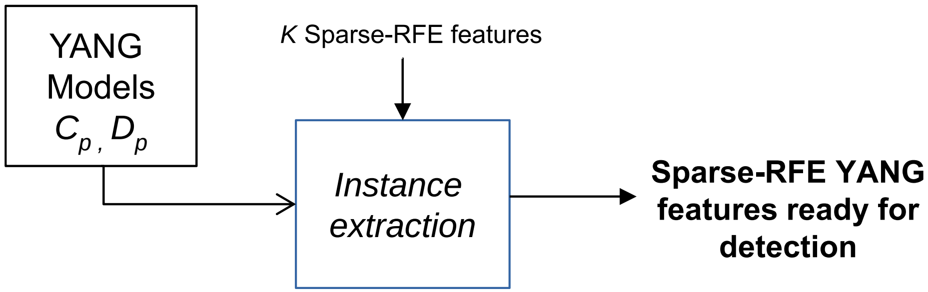
- ***K* Sparse-RFE features**
  - Data features decreased to *K*
  - Comparable performance
  - YANG features??





What Telemetry data are important?

- i.e. which YANG leafs have been instantiated?
  - **12 CP+DP common YANG features**



Plane	Instances	Level
Control	deleted-routes-count global-established-neighbors-count* protocol-route-memory	Node
Data	free-application-memory free-physical-memory incomplete-adjacency-packets no-route-packets ram-memory* system-ram-memory* total-cpu-fifteen-minute total-cpu-five-minute* total-cpu-one-minute	Node
	input-data-rate* input-load* input-packet-rate* output-data-rate* output-drops* output-load* output-packet-rate* reliability*	Interface

*Common to any choice of K.

Monitoring using Sparse-RFE

- **Which features?** 12 CP+DP YANG leafs are important for detection
- **From where?**
  - Data features could be down to small *K* →
  - Keep *T* small:
    - For visualization
    - To avoid transformation
  - Any node?
    - Avoid failure nodes

- Pick a ML model, test with:
  - A: 1 spine, 1 leaf, 1 edge node
  - B: 1 edge, 2 leafs nodes
  - C: 3 fabric nodes
  - D: 3 edge nodes

Set	Accuracy	Precision	Recall	YANG CP+DP	Total features
A	95.46	95.98%	93.20%	12	1028
B	95.88	96.03%	94.17%		1028
C	94.23	94.95%	91.26%		756
D	93.41	93.93%	90.20%		1644

What did we learn?

- Telemetry data can be big, careful preprocessing is needed for visualization and interpretability.
- Automatically selected YANG features can reduce the number of monitoring nodes and data processed.
  - Operators can select the predictors to inspect using Sparse-RFE
- Not all CP/DP leafs are needed but they enable other applications, e.g. localization.

Thank you

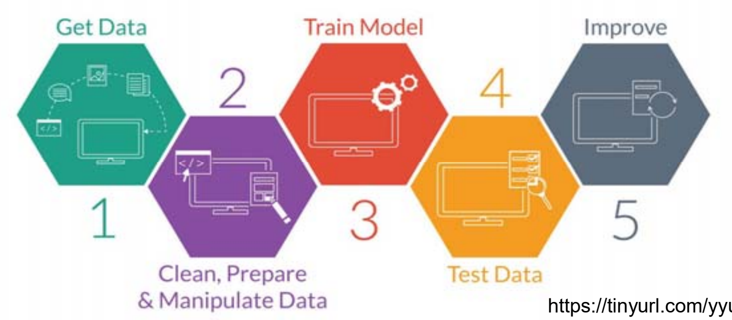
- Currently working on:
  - Localization through multi-class/multi-label techniques.
  - Exploring larger YANG trees.
  - Efficient Telemetry placement.

- Contact:

Jose Cordova-Garcia
jecordov@espol.edu.ec

Summary

- Vast literature on anomaly detection,
  - Steady interest on BGP anomalies
- Based on ML techniques:
  - PCA [Huang'07]
  - SVMs [Al-rousan'12]
  - **DBScan [Putina'18]**
  - LSTM [Chen'19]



https://tinyurl.com/yyu5j73g

- Revise the data collection and features used.
  - Data collection → Telemetry → Large monitoring data
  - Telemetry features for detection of node failures:
    - **Which features?**
    - **From where?**
    - **Transform?**

Extra info on other methods

Features used in automated methods

- *Al-Rousan'12: 17 features of volume and AS-path: numbers, means, and maximum values. Samples are 1 minute resolution. 10 features are selected and use Naive Bayes.*
- *Al-Rousan'12: 20 new features based on common values of AS-path lenghts and edit distances are added and use SVMs and HMMs*.

Features used in automated methods

- *Based on BGP update messages, commonly accepted [Ding'16].*
  - *Volume: # of BGP announcements/withdrawals*
  - *AS-path: length and edit distance*
  - *Ding'16: 37 features from BGP updates, 1 minute resolution*

Features used in automated methods

- *Deshpande'09: 4 features from BGP updates, but may be less, 5 minute resolution, inspect per-feature and analyze correlations*.

Features used in automated methods

- *Cheng’16: 33 BGP traffic updates, aim to find the proper scale for processing features by averaging p samples, i.e. in a way finding the right bin size. Once found, use LSTM for detection. Considers minute resolution data points.*
-

Features used in automated methods

- *53 aggregated features: mean and std. Deviation of average packet size, entropy of destiantion ports, etc. [Nguyen’19]*
  - *Stats from netflow data binned in 3 minutes flows VAE*
  - *Bins with less than 10 netwflow records are removed.*
- *Flow-based methods rely on proper stats and granularity (bin/window size) gathered in the flow and source nodes in that flow, not really polling status.*
-