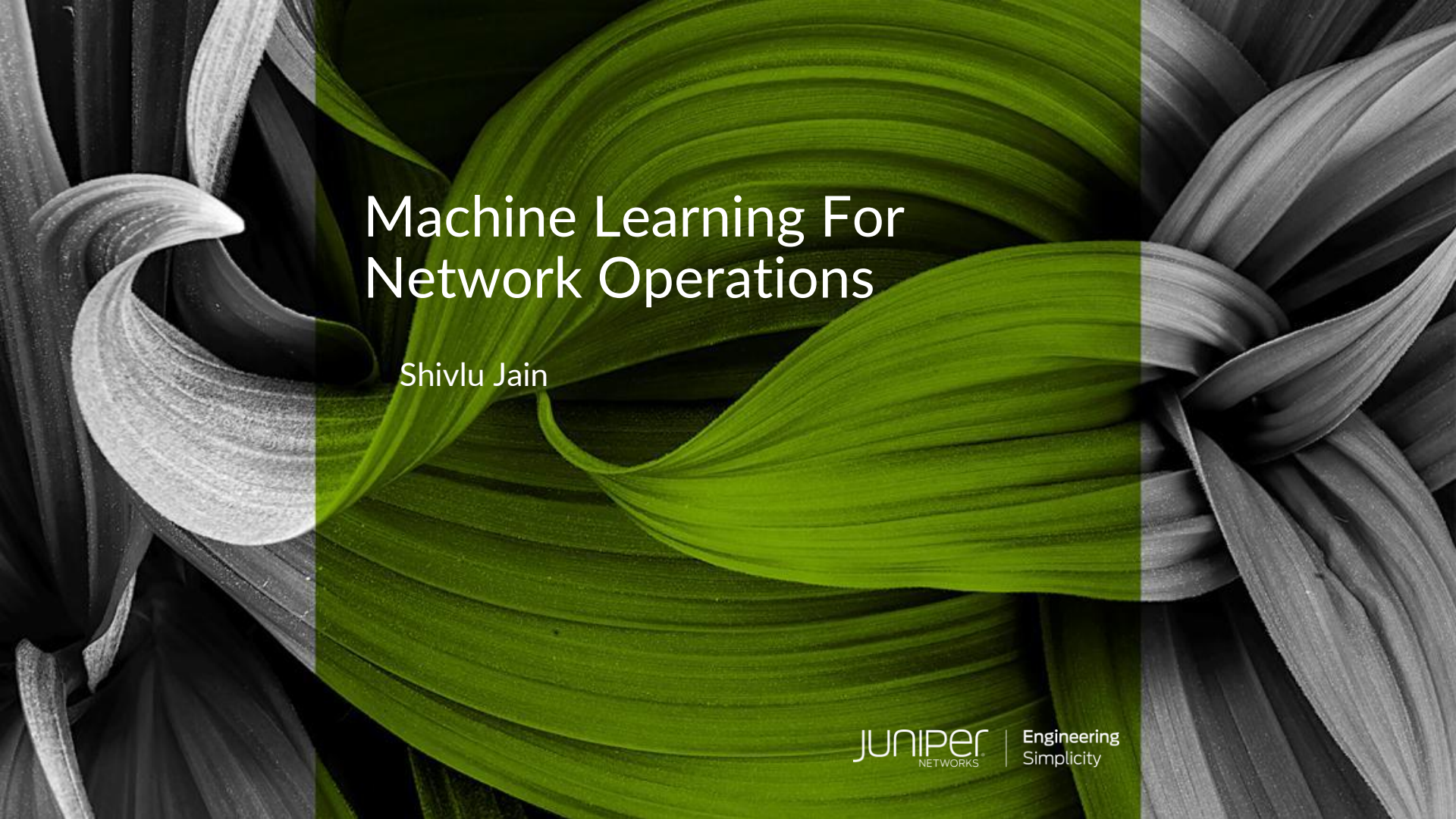


Machine Learning For Network Operations

Shivlu Jain

JUNIPER NETWORKS | Engineering Simplicity

Agenda

Types OF Approaches

What is Learning?

Herbert Simon: “Learning is any process by which a system improves performance from experience task.”

What is the task?

- Classification
- Categorization/clustering
- Problem solving / planning / control
- Prediction
- others

Machine Learning is

Learning patterns from data

Supervised, Unsupervised, Semi-supervised and (Reinforcement) Learning

Supervised: Learning from fully labeled data sets

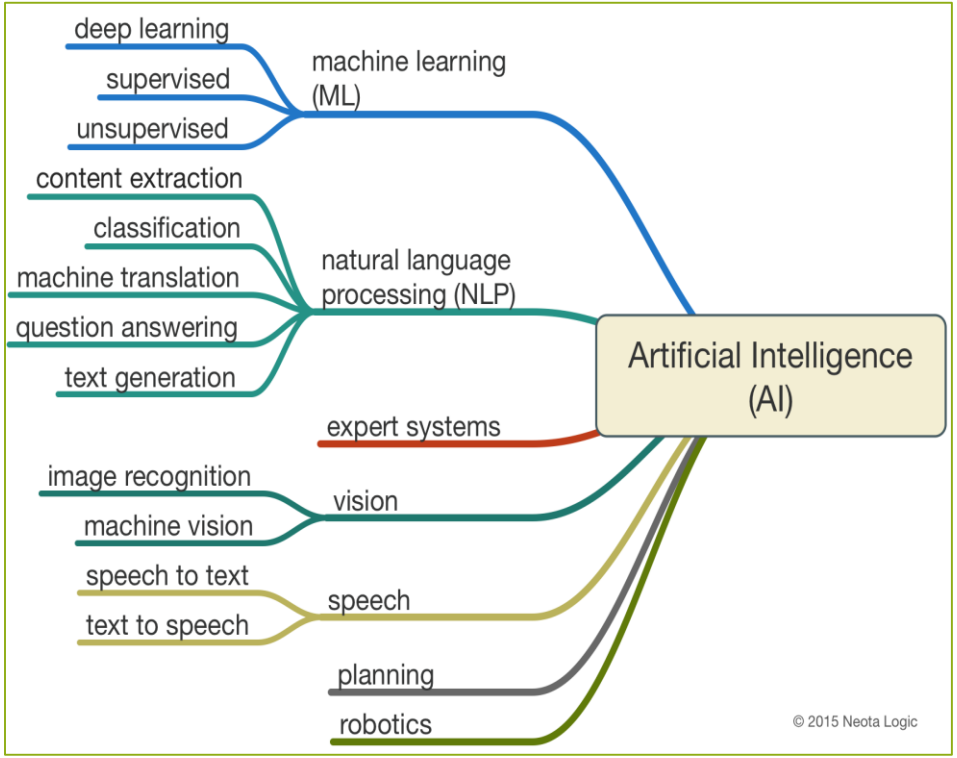
- Any classification or regression types of prediction
- Given a set of malware and clean files, predict if a new file is malware

Unsupervised: Learning from unlabeled data sets

- Explore groups or patterns, completely driven by data
- From the network data, find similar groups of users/nodes, Community detection
- Anomaly detection

Semi-supervised: Learning from partially labeled data

Reinforcement: Learning by experience/simulation



Approach

Collect Training Data

Modelling Test Data Based On Training Data

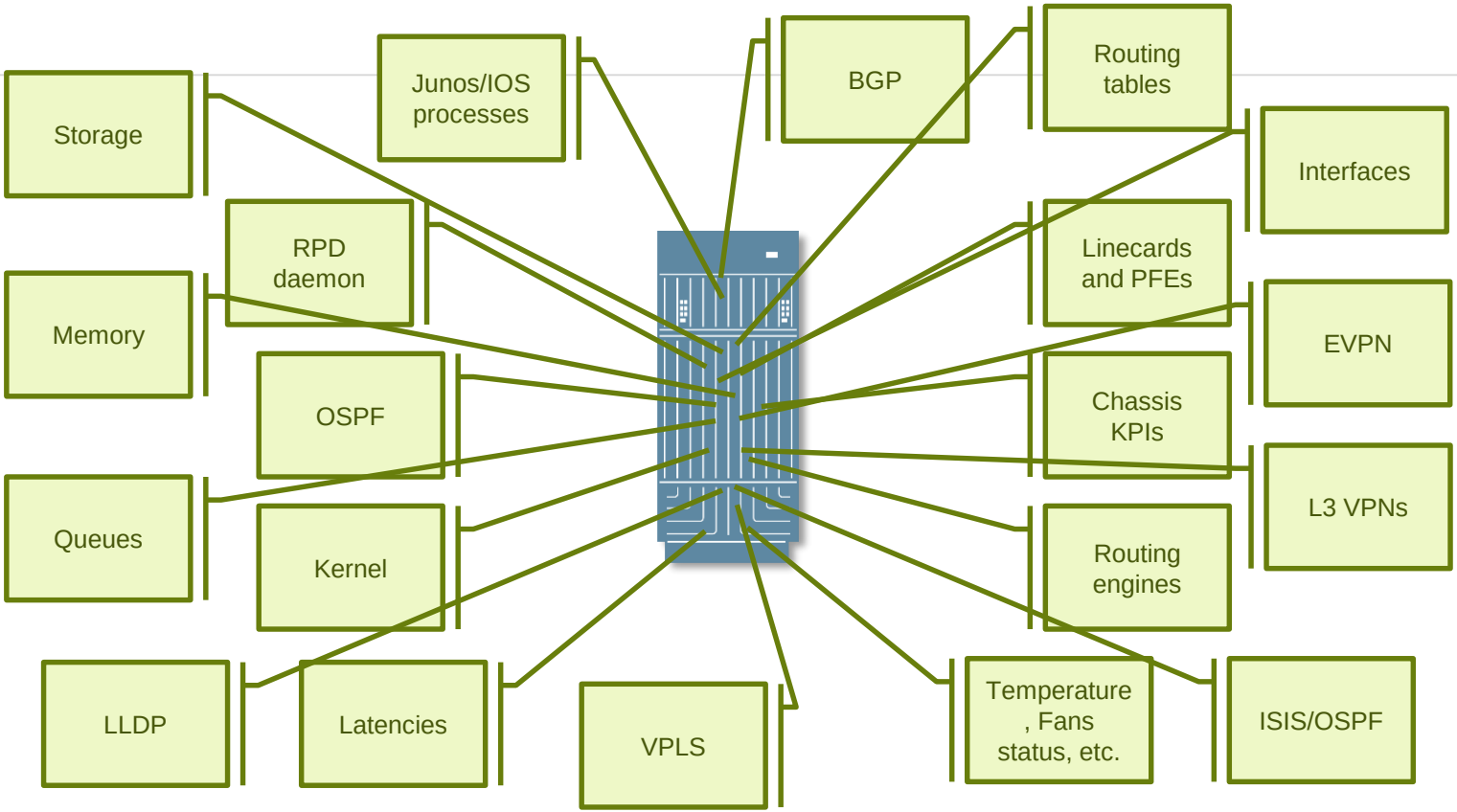
Prediction on Test & New Input Data

Applications of Machine Learning

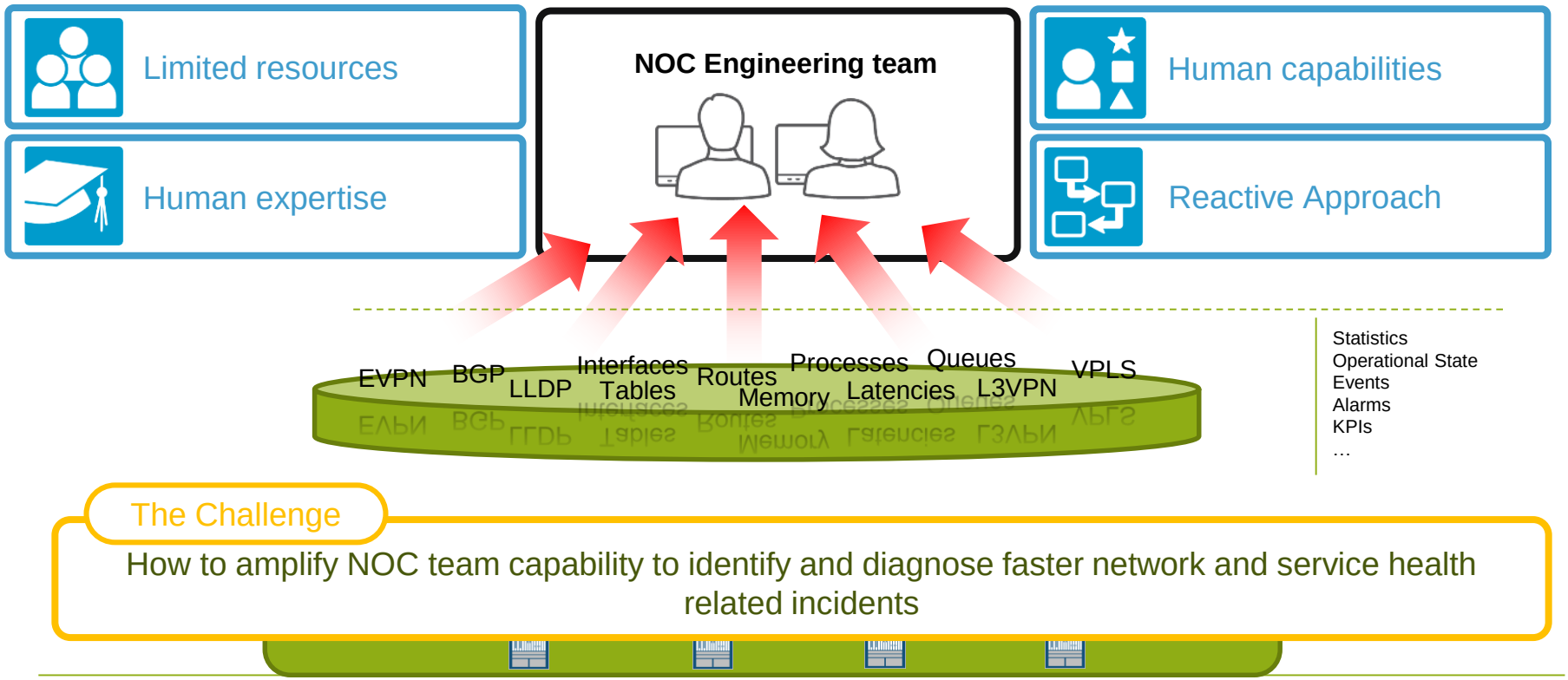
- ✓ Virtual Personal Assistant
- ✓ Predictions While Computing
- ✓ Product Recommendations
- ✓ Email Spam and Malware Filtering
- ✓ Service Provider Networks
  - ✓ Optimization Network Operations
  - ✓ Device Identification By Using DHCP
  - ✓ How To Deploy & Test Machine Learning
- ✓ **Applications Are Endless…….**

Will Be Covering Use Cases Around It

Network Is All About



The Network Operations Center challenge



The Challenge

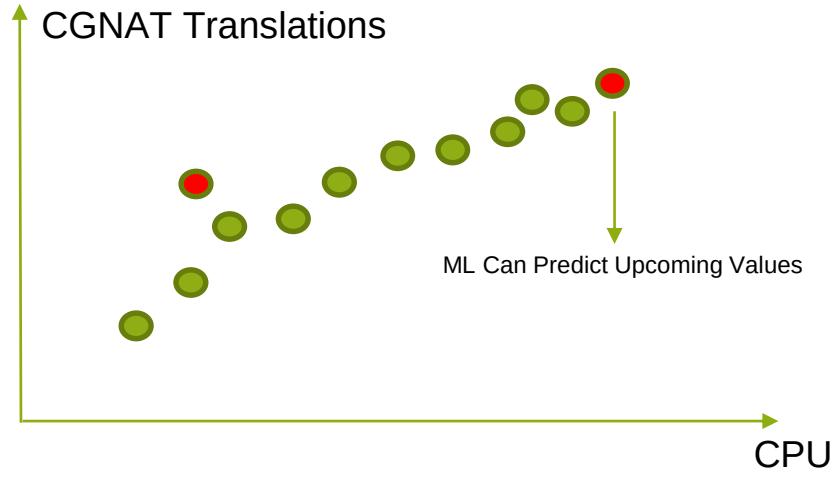
How to amplify NOC team capability to identify and diagnose faster network and service health related incidents

How to Predict CPU Values?

Problem statement

Whenever the CGNAT translations increases, CPU always increases. But How much? This is the unknown number which is not known to anyone?

- ✓ Operation Team Can Leverage ML To Find The Next UpComing Value Of CPU With Increase In The NAT Translations.
- ✓ This will help them to plan for any kind of future requirement
- ✓ It can immediately raise alarm if the CPU is high but translations are less

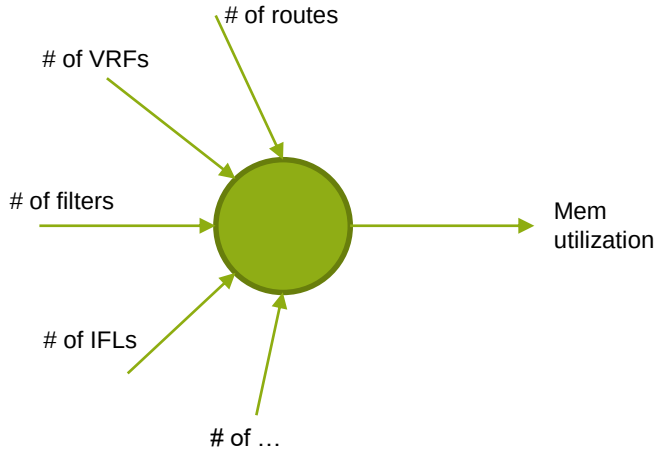


Finding Benchmark To Stop Provisioning

Problem statement

Routers are always loaded with customer links, VRFs, routes etc. But what would be the benchmark to stop the new provisioning?

- ✓ It's hard to stop provision on any router without benchmarks or rule based standards.
- ✓ Rule Based Standards are reactive and has it's finite boundaries.
- ✓ ML can help to use proactive approach to stop or increase the provisioning

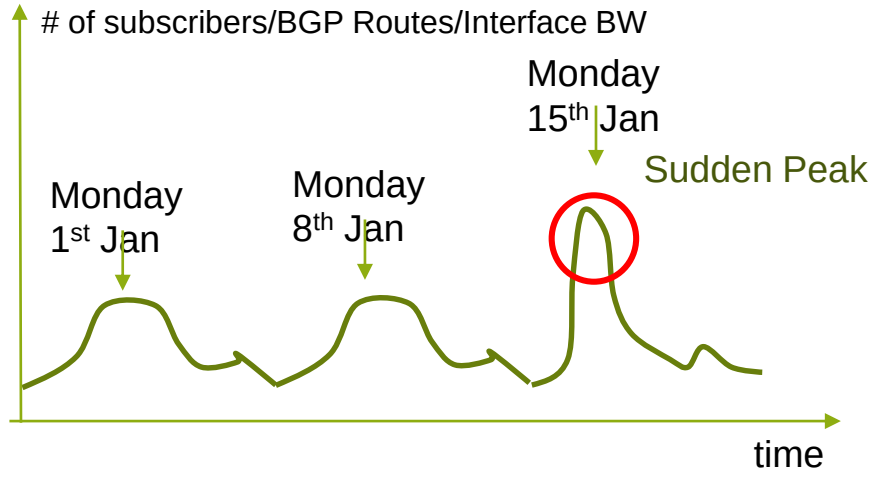


Time Based Anomaly Detection

Problem statement

Identify on a time series non-expected values that can be a symptom of a misbehavior

- ✓ Sudden Peak In Subscriber/BGP Routes/Interface BW Count At Particular Time On Specific Day as Compare To Other Day at same time.
- ✓ Clearly Shows There Is Anomaly
- ✓ Leverage ML to find these anomalies which are not possible in reactive approach

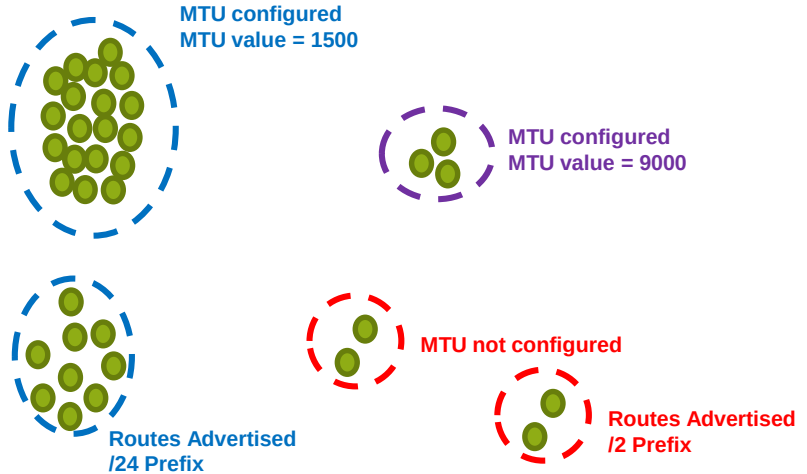


Configuration Anomaly Detection

Problem statement

Configurations sometimes deviate from the expected template especially while using CLI, and that leads eventually to network failures.

- ✓ Clustering can help operation team to find the best and suitable configuration for their end customer or internal access, aggregation or backbone links.
- ✓ ML will let operations teams to judge what is required and what is not required in specific conditions.

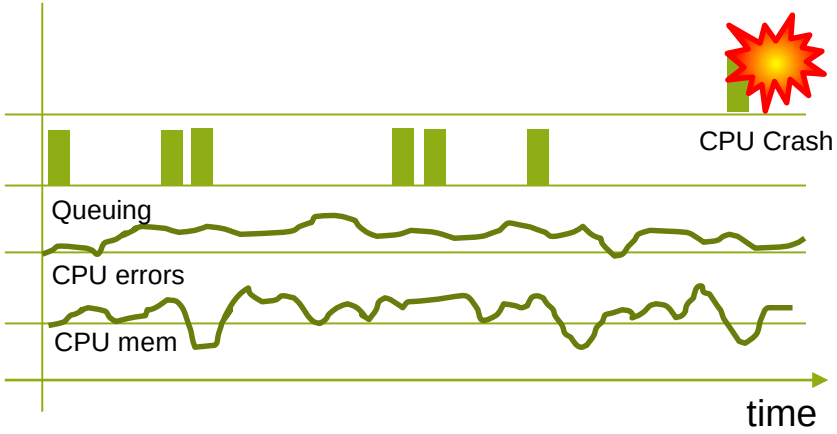


Event Cross-Correlation – Router Will Go Down or Not

Problem statement

Ability to predict the occurrence of certain events like card, power or chassis down

- ✓ Router is generating lot of telemetry data
- ✓ With Correlation of different Events ML can predict whether the router will stay up or can go down.



Device Identification With DHCP

Problem statement

How to classify the device category of new user. Device categories are Laptop/Mobile/Tablet

Label the Device Class

Collect the data from the DHCP logs and based on this we can identify the Device Class

Device Class	MAC Address	DHCP Options	DHCP Vendor
Mobile	0c:72:d9:dd:9e:e3	1,3,6,15,26,28,51,58,59	Android
Laptop	08:05:81:62:a1:cf	1,3,6,15,12	
Tablet	34:ab:37:07:4d:b5	1,121,3,6,15,119,252	

OUI can be used to classify the type of device.

DHCP Options allow the client to request specific responses from the DHCP server. The options requested, and their order are strongly correlated to the device's operating system.

How To Deploy & Test Machine Learning: Its My Own Experience

Tools Used For Implementing Use Case: Jupyter Notebook, Pandas, Python & Sckit learn Library For Machine Learning

Test Use Case Details

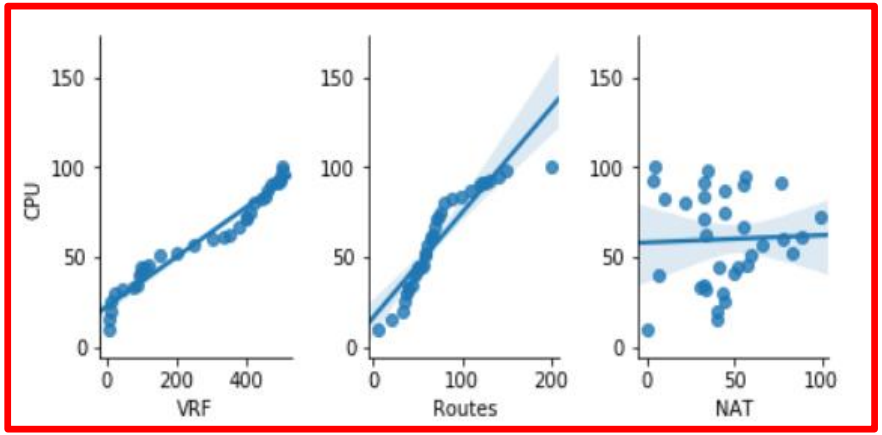
- ✓ Most of the time NOC engineers busy in checking the CPU values
- ✓ There could be n number of reasons for high CPU
- ✓ Identity which process is causing more CPU and restrict the provisioning process
- ✓ Use of Machine Learning will help to predict the closest value of CPU and help NOC engineers to rate limit or restrict the provisioning

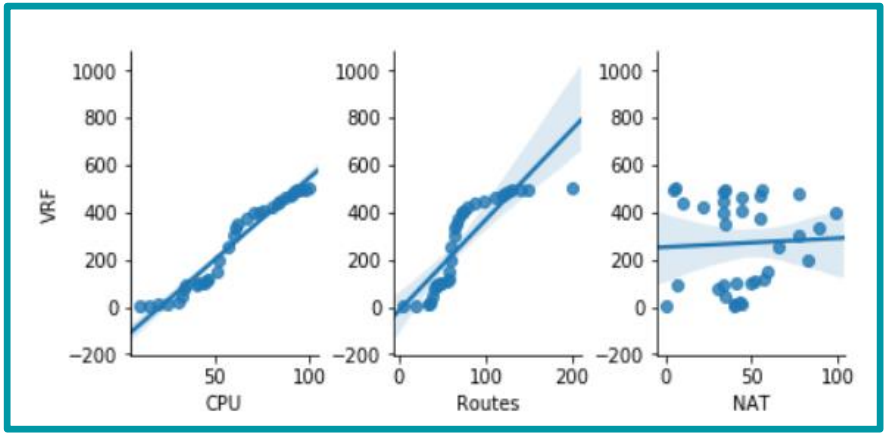
Note: The values used in this example is not taken from any network. These are imaginary values which are used for showcase purpose only

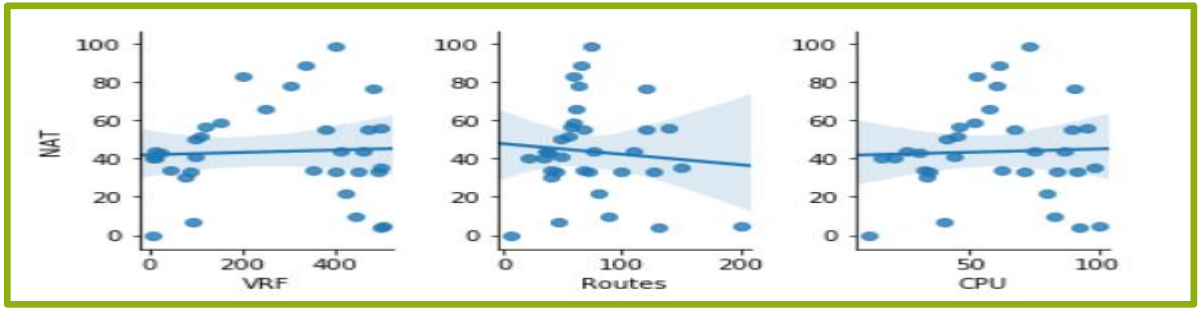
Evaluation Approach

1. Created dummy data set of 33 rows and 4 columns. Data set has CPU, VRF, Routes and NAT numerical values.
2. All the values are stored in CSV file and Pandas library is used for import.
3. Split the dataset in train and test split (70 – 30 ratio)
4. Passed the train data set to machine learning model
5. Train the model with training data set
6. Test the model with test values
7. Compare the test output values vs original values
8. Provide the new values and model will predict the CPU outcome of the router

Analysis of the dataset and finding the correlation







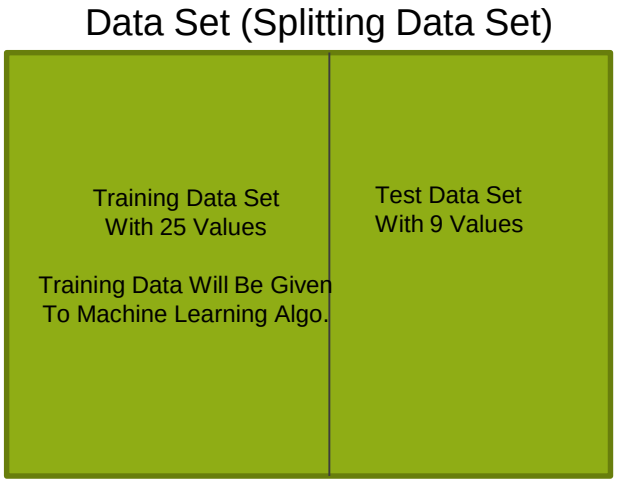
The Machine Learning Training Model Will Work Best If We Exclude NAT CPU From The Data Set

Training Process

From The Correlation, It is clearly understood that the CPU is directly proportion to the routes and vrf. Being direct relation, we can train the model with Linear Regression Machine Learning Program
We will be splitting the dataset into 2 parts:- Training and Test. Training is used to train the model and Test is used to check how good the model is.

The training values are:

	VRF	Routes	CPU
30	495	130	93
17	333	65	61
22	410	75	75
4	23	37	30
2	10	34	20
21	401	73	73
23	420	80	80
10	100	49	44
29	490	126	92
28	480	121	91
18	350	67	62
6	75	40	33
13	150	58	51
7	89	44	34
33	500	200	100
1	6	20	15
16	300	64	60
0	5	5	10
15	250	60	57
5	45	39	32
11	110	55	45
9	99	48	41
31	496	140	95
8	92	46	40
12	120	57	46



The testing values The test values are:

	VRF	Routes	CPU
14	200	59	52
19	375	68	67
3	11	35	25
27	470	120	90
32	498	150	98
26	460	110	87
20	400	72	71
25	450	99	84
24	440	88	83

CPU Prediction on Test Data & New Data

Actual Values

Predicted Values

The testing values are:

	VRF	Routes
14	200	59
19	375	68
3	11	35
27	470	120
32	498	150
26	460	110
20	400	72
25	450	99
24	440	88

The test values are:

CPU
52
67
25
90
98
87
71
84
83

VRF and Routes Are Passed In The Train Model

The predicted values are

```
[[ 49.83956496]
 [ 69.40579519]
 [ 26.10392312]
 [ 88.61926509]
 [ 96.95808833]
 [ 85.77137169]
 [ 72.69595043]
 [ 82.74111743]
 [ 79.71086317]]
```

Model has predicted the above values corresponding VRF and Routes

Compare Actual Value

Vs

Predicted Value

Predict CPU By Give Random Values Of VRF and Routes

Input Values In The ML Model

VRF	Routes
300	80
310	100
250	110
300	120
370	140
400	150
410	160

The predicted CPU values:

```
[[ 63.91199029]
 [ 68.58349238]
 [ 64.26139281]
 [ 71.20642504]
 [ 82.02363537]
 [ 86.92009819]
 [ 89.76799158]]
```

Machine Learning Code

```
from sklearn.cross_validation import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(x_axis, y_axis, random_state=1)


from sklearn.linear_model import LinearRegression
linreg = LinearRegression()
linreg.fit(X_train,Y_train)
y_pred = linreg.predict(X_test)
print(y_pred)
print(Y_test)
```

The network operations center challenge

Expert community
MX, SRX, QFX, BGP, EVPN, L3VPN, Memory, MPLS, PTX, Cisco, …

Machine Learning

ML/AI Generated insights
Anomaly detection, cross-correlation, outlier detection, prediction, automated clustering,

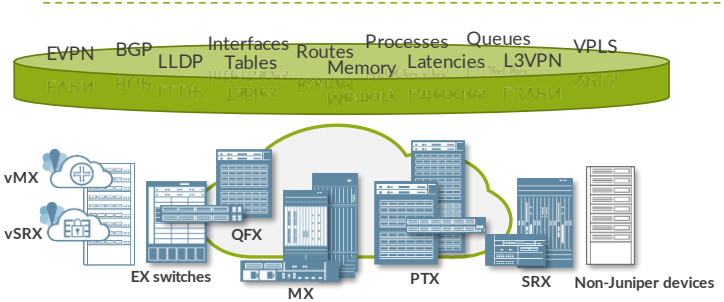
NOC Engineering team

Expert community knowledge amplify NOC team expertise

- Limited resources
- Human expertise

Machine generated knowledge amplifies NOC team capabilities

- Human capabilities
- Tools




Thank you