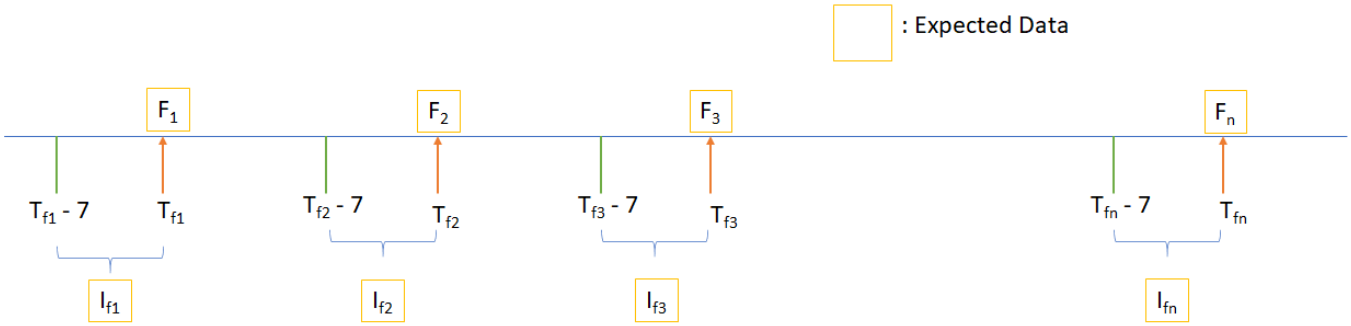
Failure Prediction using AI/ML in NFV Environments

Data



Failure Type	Failure parameter	F_n Failure Event	I_f Infrastructure Metrics	Comments

Links	Link Down.
	
	Link removed
		Virtual Switch link failure
		
		Hardware Failure
		
		Interface Down
		
		`dhcp-agent.log` / `neutron-dhcp-agent`
		`l3-agent.log` / `neutron-l3-agent`
		`linuxbridge-agent.log` / `neutron-linuxbridge-agent`
		`openvswitch-agent.log` / `neutron-openvswitch-agent`
		
		(Ref: https://docs.openstack.org/ocata/config-reference/networking/logs.html)
			Network interface status,
			
			High packet drop,
			
			low throughput,
			
			excessive latency or jitter
			
			crc-statistics, fabric-link-failure, link-flap, transceiver-power-low	

VM	Deployment/Start Failures: 1. Failed to start* 2. Failed to boot* Post-Deployment /Start failures: 1. Shutdown 2. Crash 3. Hang* 4. Panic	nova-compute.log nova-api.log nova-scheduler.log libvirt.log qemu/$vm.log neutron-server.log glance/cinder - flavor Node and Core-mapping	cpu: per-core utilization memory Interfaces statistics - sent, recv, drops Disk Read/Write	If possible, Infrastructure metrics and syslogs from within the VM should be collected. Deployment/Start failures can be the first step.
Container	Deployment/Start Failures: 1. Failed to start* 2. Failed to boot* Post-Deployment /Start failures: 1. Shutdown 2. Crash 3. Hang 4. Panic	• OS layer – syslog, boot.log, kern.log etc. • Kubernetes Layer – container Logs (/var/log/containers) • OpenStack Layer – OpenStack service Logs	cpu: per-core utilization memory Interfaces statistics - sent, recv, drops Disk Read/Write	

<table>
<tr>
<td>Node</td>
<td>A node failure (hardware failure, OS crash, etc)

A) node network connectivity failure

B) nova service failure

C) Failure of other OpenStack services</td>
<td>/var/log/nova/nova-compute.log
(To ensure that it has successfully connected to the AMQP server
Ref: https://docs.openstack.org/operations-guide/ops-maintenance-compute.html)

<table>
<tr><th>Cloud controller</th><th>nova-*</th><th>/var/log/nova</th></tr>
<tr><td>Cloud controller</td><td>glance-*</td><td>/var/log/glance</td></tr>
<tr><td>Cloud controller</td><td>cinder-*</td><td>/var/log/cinder</td></tr>
<tr><td>Cloud controller</td><td>keystone-*</td><td>/var/log/keystone</td></tr>
<tr><td>Cloud controller</td><td>neutron-*</td><td>/var/log/neutron</td></tr>
<tr><td>Cloud controller</td><td>horizon</td><td>/var/log/apache2/</td></tr>
<tr><td>All nodes</td><td>misc (swift, dnsmasq)</td><td>/var/log/syslog</td></tr>
<tr><td>Compute nodes</td><td>libvirt</td><td>/var/log/libvirt/libvirtd.log</td></tr>
<tr><td>Compute nodes</td><td>Console (boot up messages) for VM instances:</td><td>/var/lib/nova/instances/instance-<instance id>/console.log</td></tr>
<tr><td>Block Storage nodes</td><td>cinder-volume</td><td>/var/log/cinder/cinder-volume.log</td></tr>
</table>

(Ref: https://docs.openstack.org/operations-guide/ops-logging.html)

A) node network connectivity failure

1. management network
2. VMs communication network
3. storage network

B) nova service failure (e.g., process crashed) -- detected and restarted by a local watchdog process

1. compute
2. volume
3. network
4. scheduler
5. api.

C) Failure of other OpenStack services -- N/A, assuming redundant /highly available configuration

1. Glance
2. Keystone</td>
<td>Interfaces statistics - sent, recv, drops

Hypervisor Metrics, Nova Server Metrics, Tenant Metrics, Message Queue Metrics

Keystone and Glance Metrics</td>
<td></td>
</tr>
<tr>
<td>Application</td>
<td>Crash/Connectivity /Non-Functional</td>
<td>Application Log i.e. If it is Apache then logs of Apache

(/var/log/apache2)</td>
<td>Packet Drops, Latency, Throughput, Saturation, Resource Usage</td>
<td>Deploy Collectd within the application and collect both application logs and infrastructure metrics</td>
</tr>
<tr>
<td>Middleware Services</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</table>

Models

We have taken three types of models and in those models we have considered **Failure Prediction** problem and the remaining types are given as:

1. Event Correlation
2. Anomaly Detection
3. Failure Prediction

We are focusing on Failure Prediction of Node, Application, VM, Service, Container and Links. Our aim is to predict the failures before they happen so that user can take necessary actions regarding those failures. So, to implement Failure Prediction models we are developing our models using Classical Neural Networks techniques i.e. **RNN & LSTM.**

Gaps

From the perspective of Telco after doing a literature survey we found most of the work has done on VM and Applications. There are less work has done for Node Failures, Link Failures, Middleware Services and also there is a lack of Publicly available datasets for these failures. Majority of researchers have used **ARIMA & RNN** so to improve the performance of the prediction model we can do some experiments with **Generative Adversarial Networks (GAN), Graphical Neural Network (GNN)**. Also, in our literature survey we found that majority of the publicly available data does not contain time stamp. So to make the future predictions we will need of Time Series data.

Enhancements