MATLAB EXPO

November 13–14, 2024 | Online

MATLAB & Databricks: Accelerating Data Science and Machine Learning

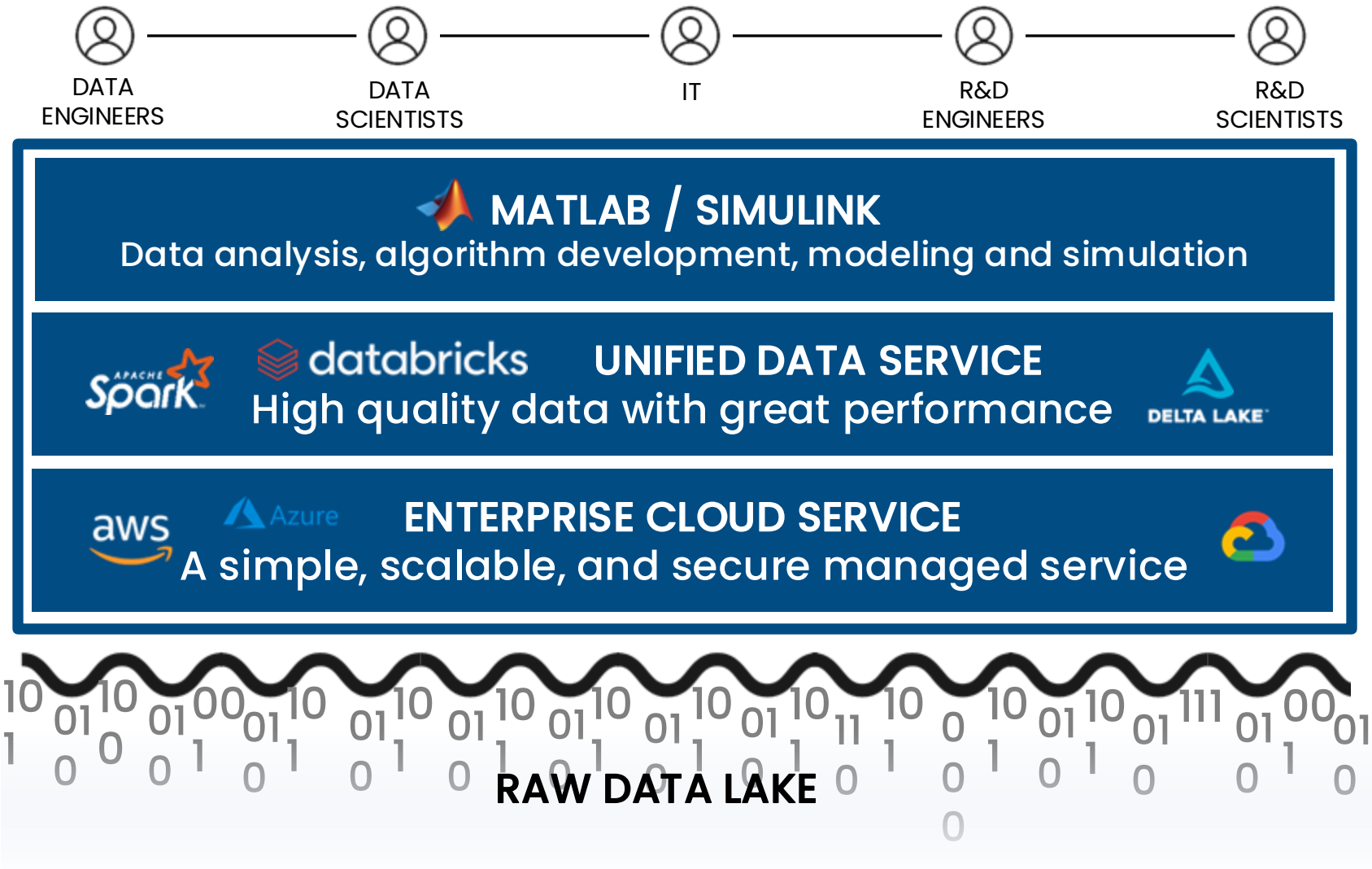
Zachary Jacobson, Databricks

Andy Thé, MathWorks

MathWorks

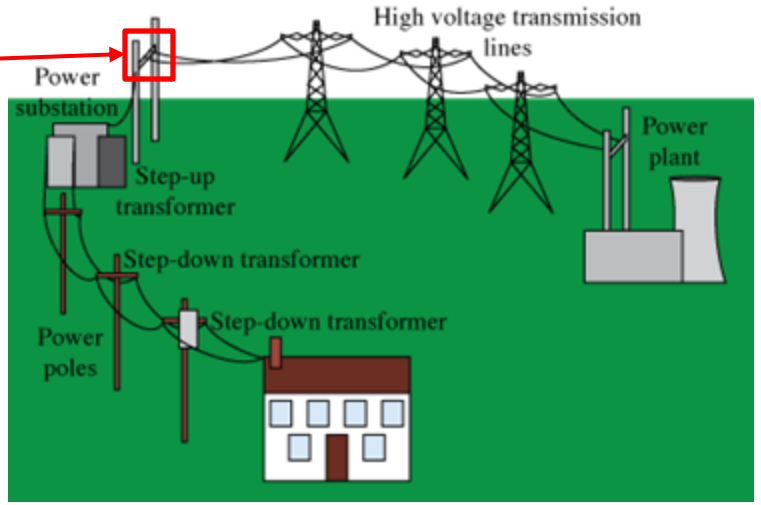
Integration Approach

"The entire company on one data and compute platform in the cloud"



The Problem

Utility companies waste thousands of hours trying to identify and locate damaged power lines caused by electrical faults.

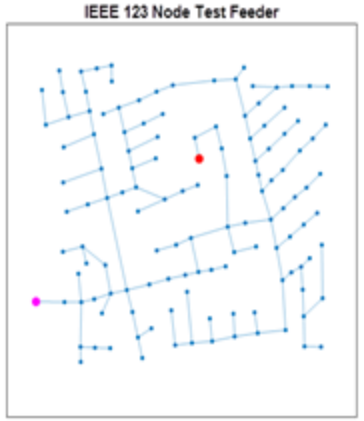


With MathWorks and Databricks you can use data-driven techniques to generate a classification algorithm for fault location detection on electric grids.

This will greatly help utility companies handle power outages.

The Approach

1. Create a model to generate synthesized data
   - ❑ *IEEE 123 Simscape model*

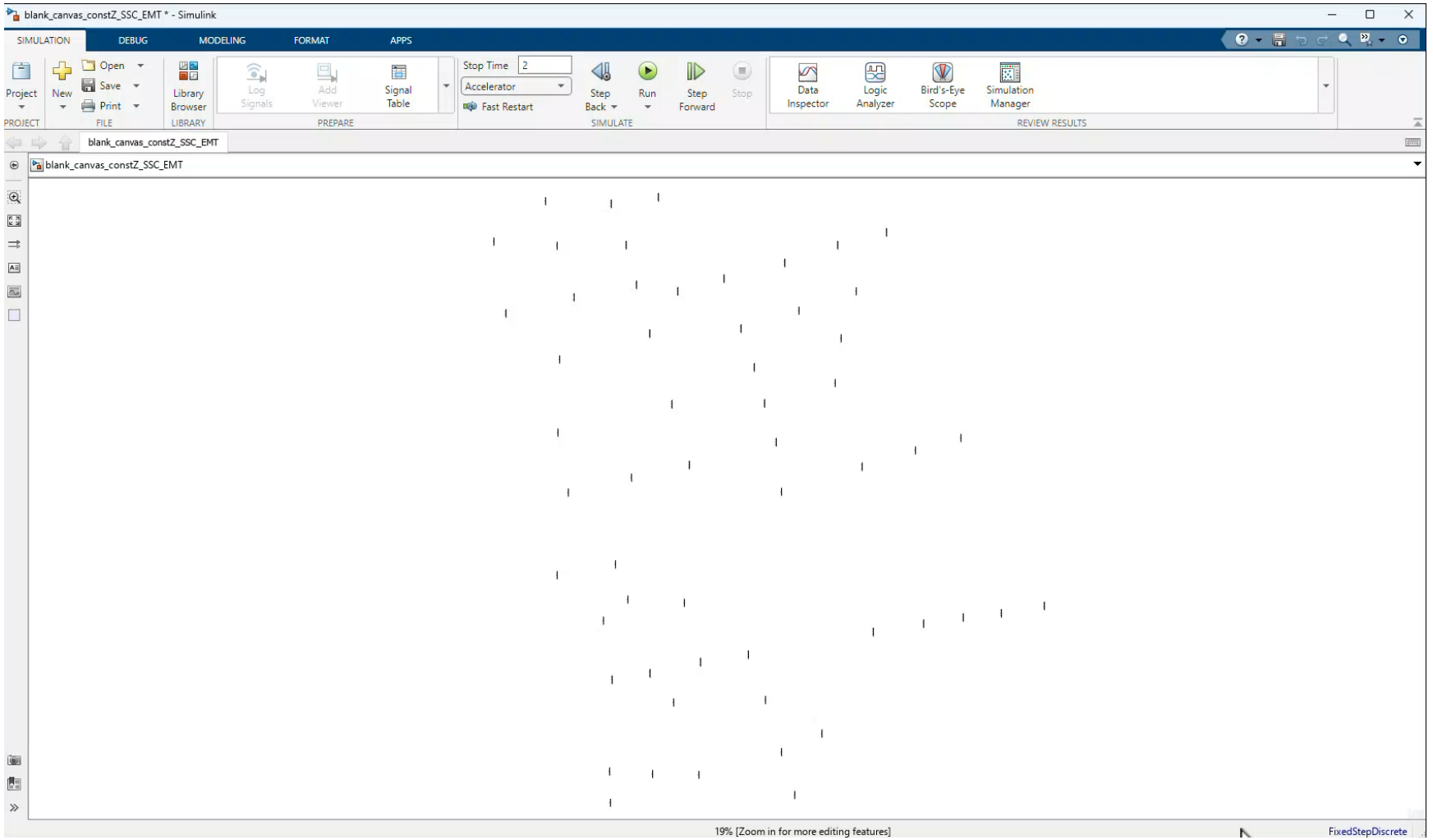


2. Run the model at scale to generate large amounts of data
   - ❑ *Run the Simscape model in Databricks on 100+ instances*

3. Train a classification algorithm using the synthesized data
   - ❑ *Train a neural network in Databricks with the large data set*

4. Use the trained neural network to predict the locations when a fault occurs
   - ❑ *Input a fault into the neural network to get a distance from the power plant*

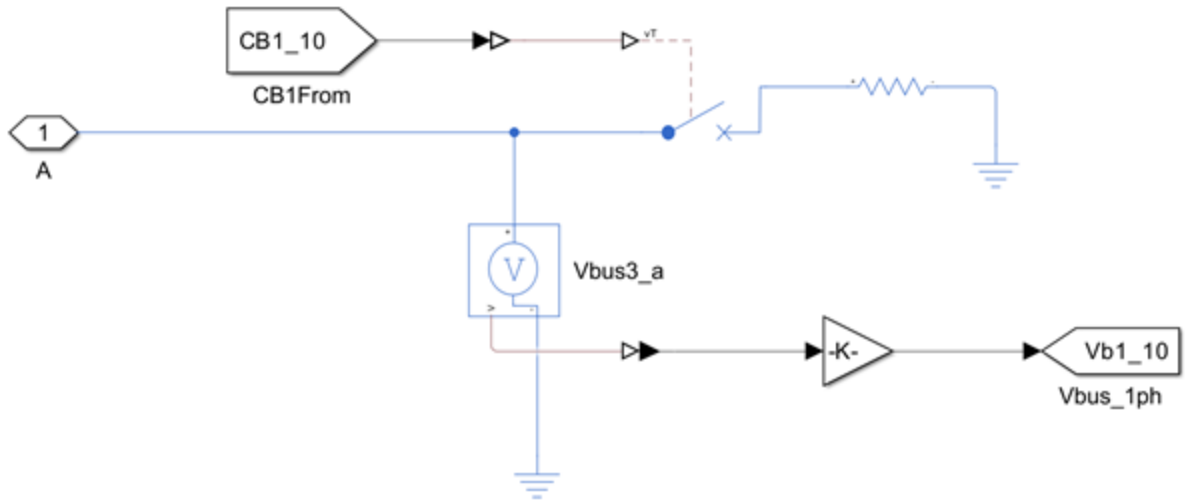
Create a simulated model to generate synthesized data IEEE 123 Simscape model



Use the Simulink API to programmatically build the IEEE 123 Node Test Feeder

Fault Insertion

- Each bus is configured such that a fault can be applied on each phase.



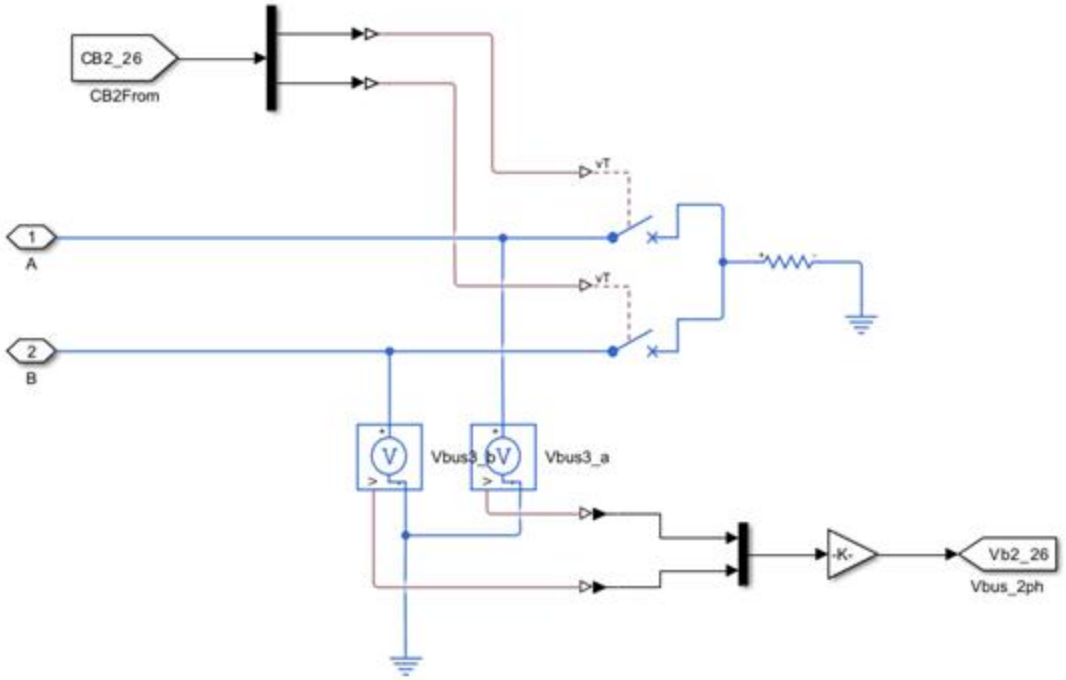
Single-Phase Bus

1 fault combination,

a-g

Fault Insertion

- Each bus is configured such that a fault can be applied on each phase.



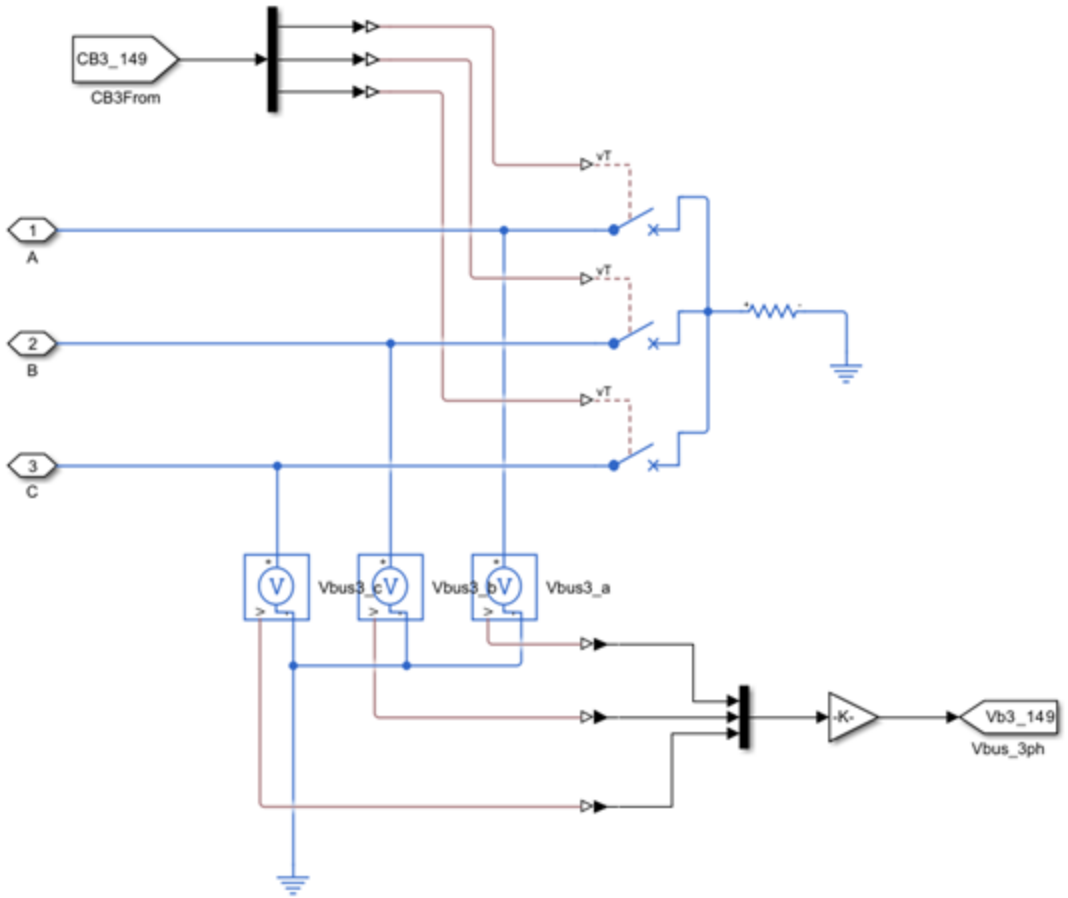
Two-Phase Bus

3 fault combinations,

a-g
b-g
a-b-g

Fault Insertion

- Each bus is configured such that a fault can be applied on each phase.

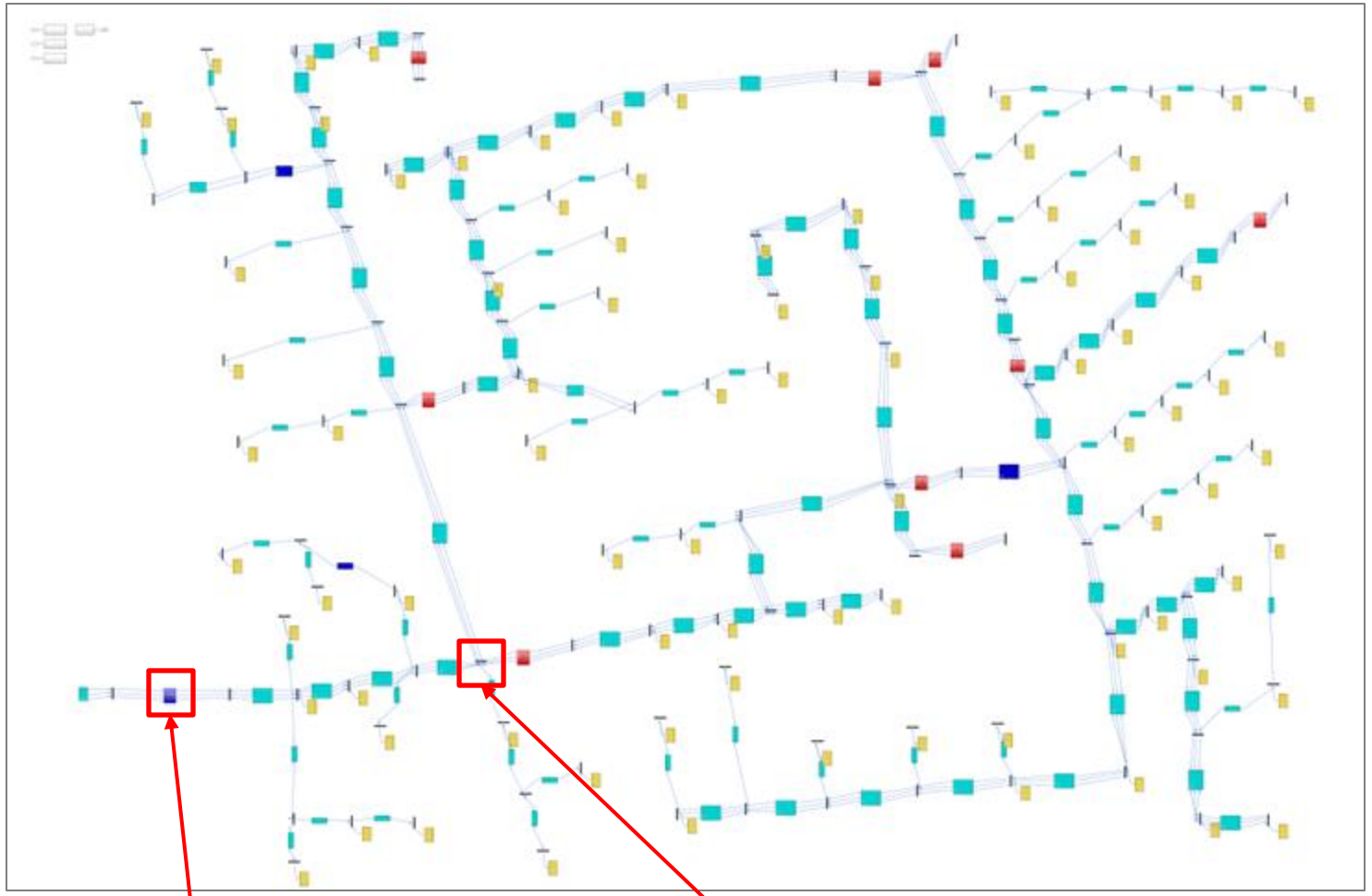


Three-Phase Bus

7 fault combinations,

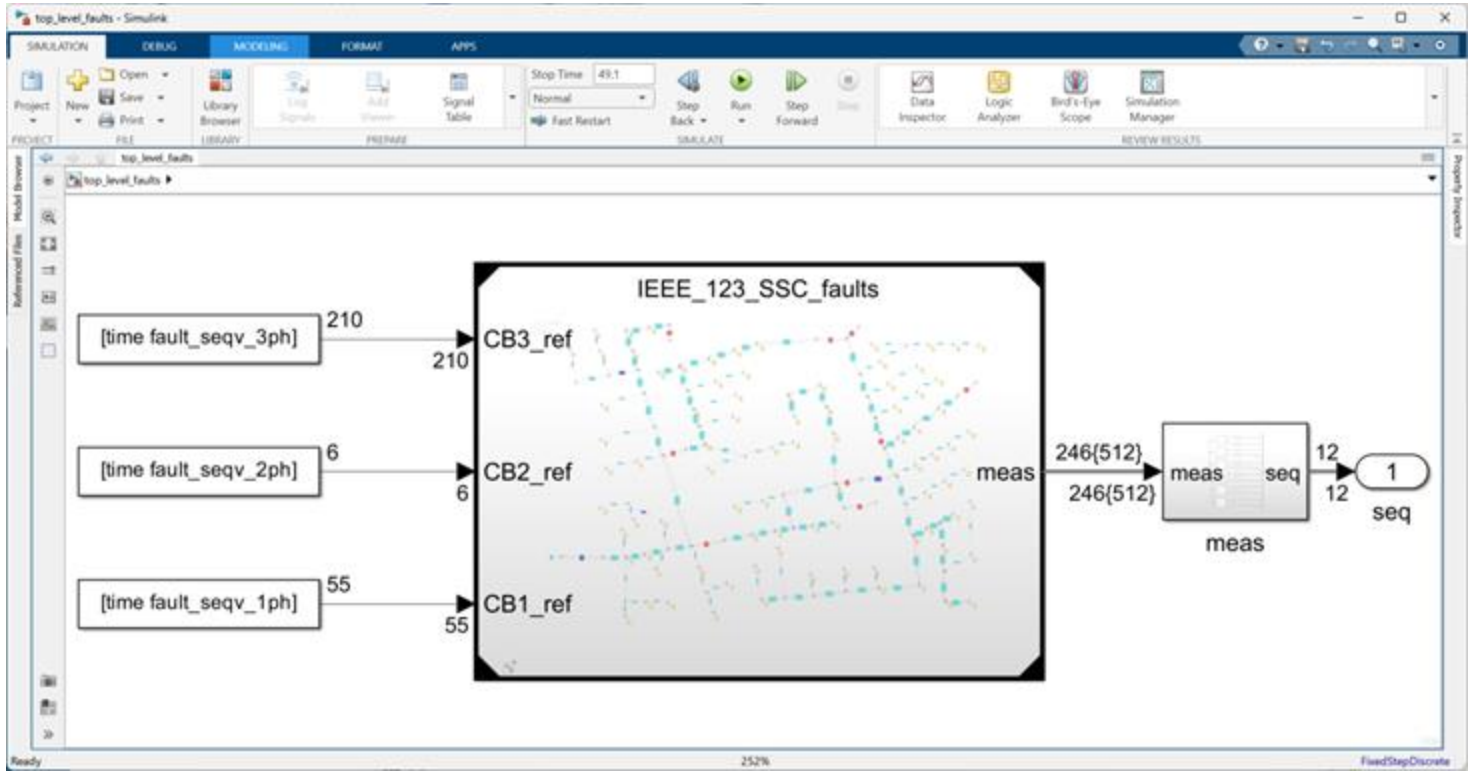
a-g
b-g
c-g
a-b-g
a-c-g
b-c-g
a-b-c-g

Fault Insertion



70 3-phase buses = 490 faults
3 2-phase buses = 9 faults
55 1-phase buses = 55 faults
Total faults = 554 faults

Fault Insertion

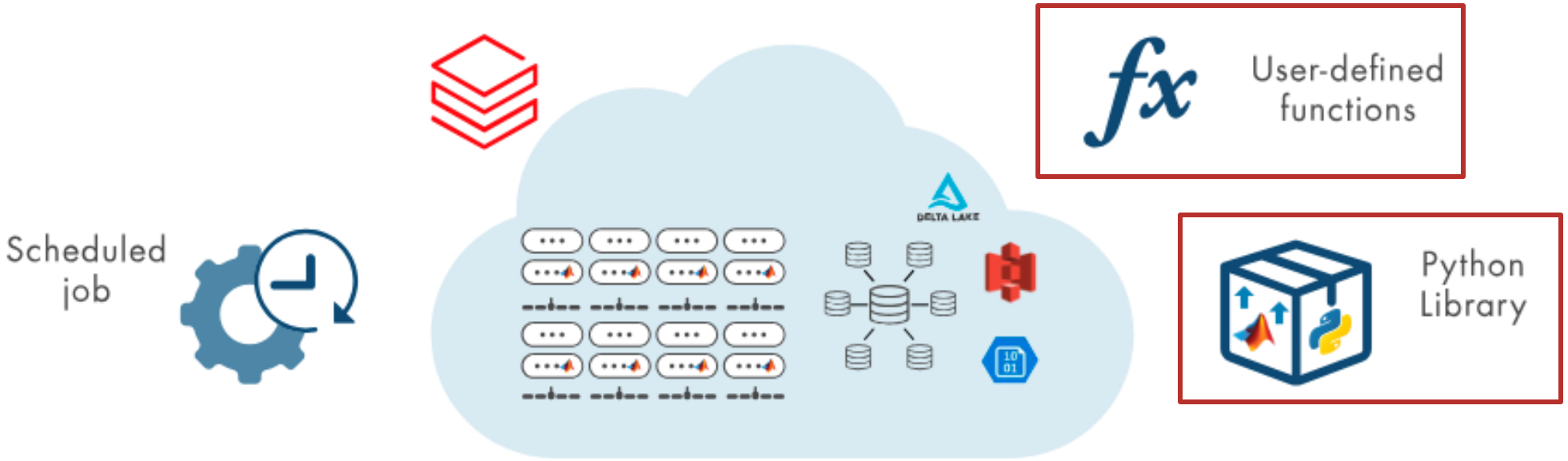


Each fault is applied and cleared sequentially in a single simulation run, through appropriate circuit breaker control signals

Bridge the Gap:
Sharing Your Models with Databricks Innovators

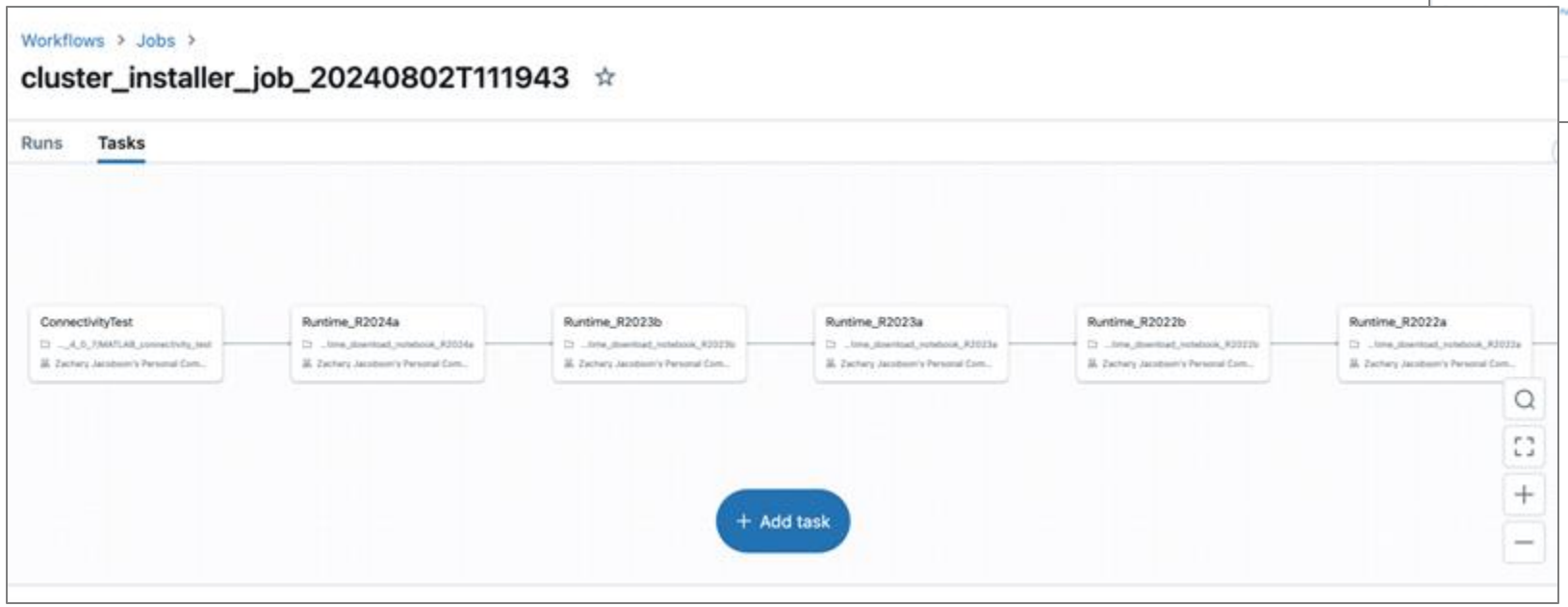
Once models are deployed to Databricks, share them with other users and incorporate them into multi step workflows. Deploy and execute models natively as jobs, at scale, or on MATLAB Runtime-enabled clusters.

Deploy models as a library or a user-defined function for use in any language, including SQL scripts. Java or Python libraries and notebooks can also use this code.



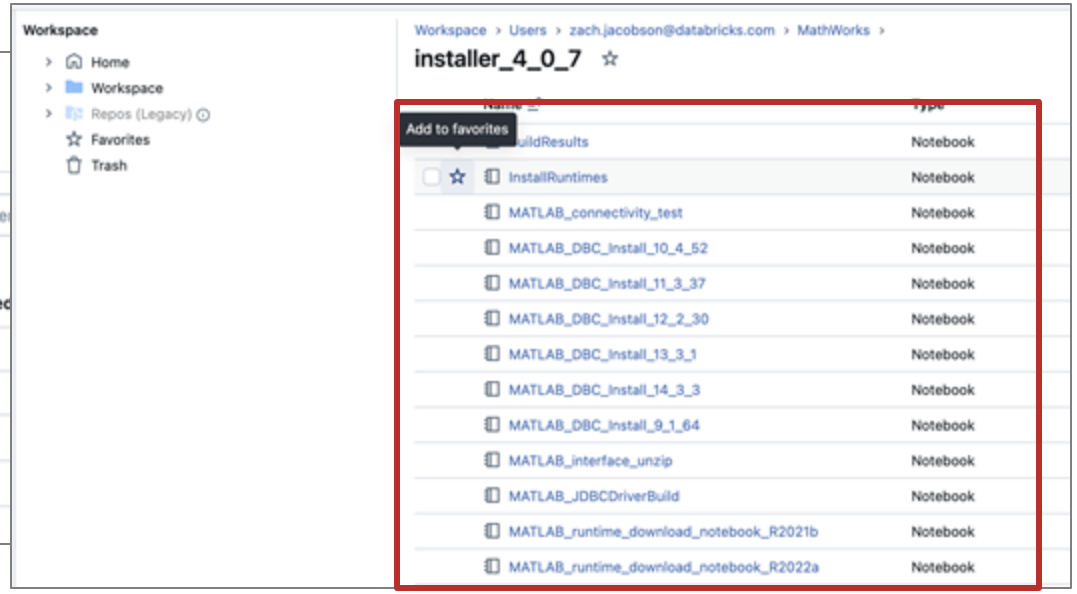
Simulink Runtime Installation on Databricks

- Deploying once on Databricks workflow
- Access to dependencies in Workspace folder
- No need to rebuild again (only build once)



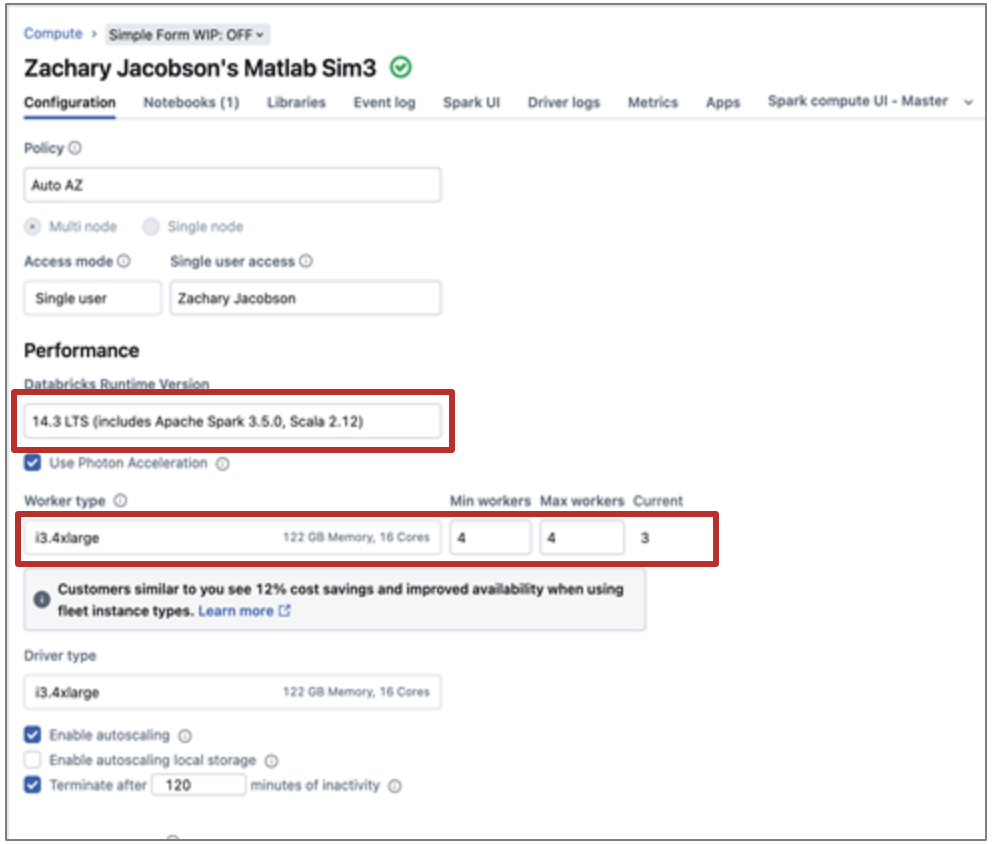
Accessing Runtime from Workspace and UC Volumes

- Runtime is governed under Unity Catalog
- Permissions can be managed under Unity Catalog
- All unstructured data and files can be managed in Volumes

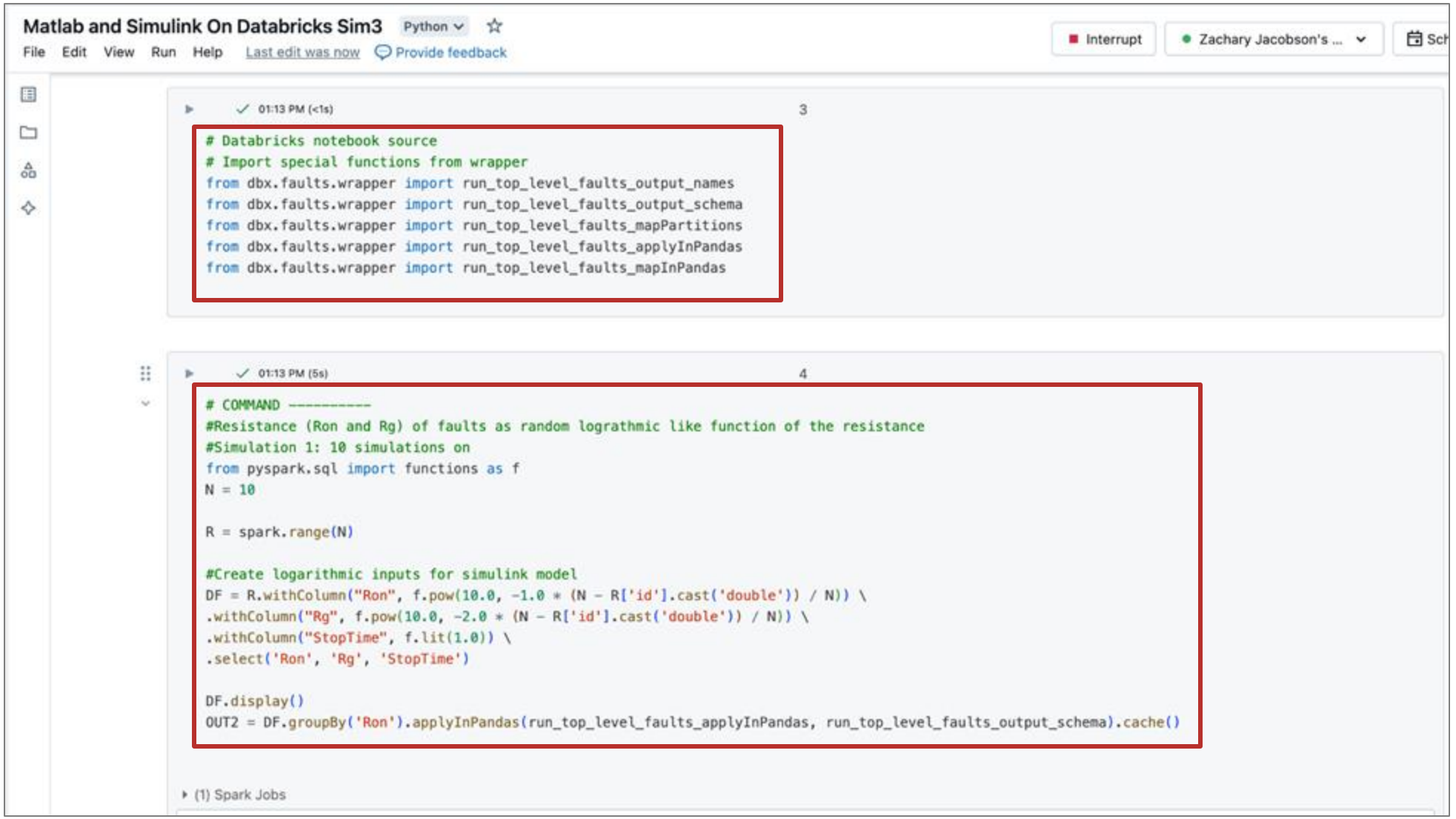


Cluster Deployment

- Cluster configured once
- 14.3 Runtime Version
- Scale cluster to appropriate size
- Install custom libraries/wheel files

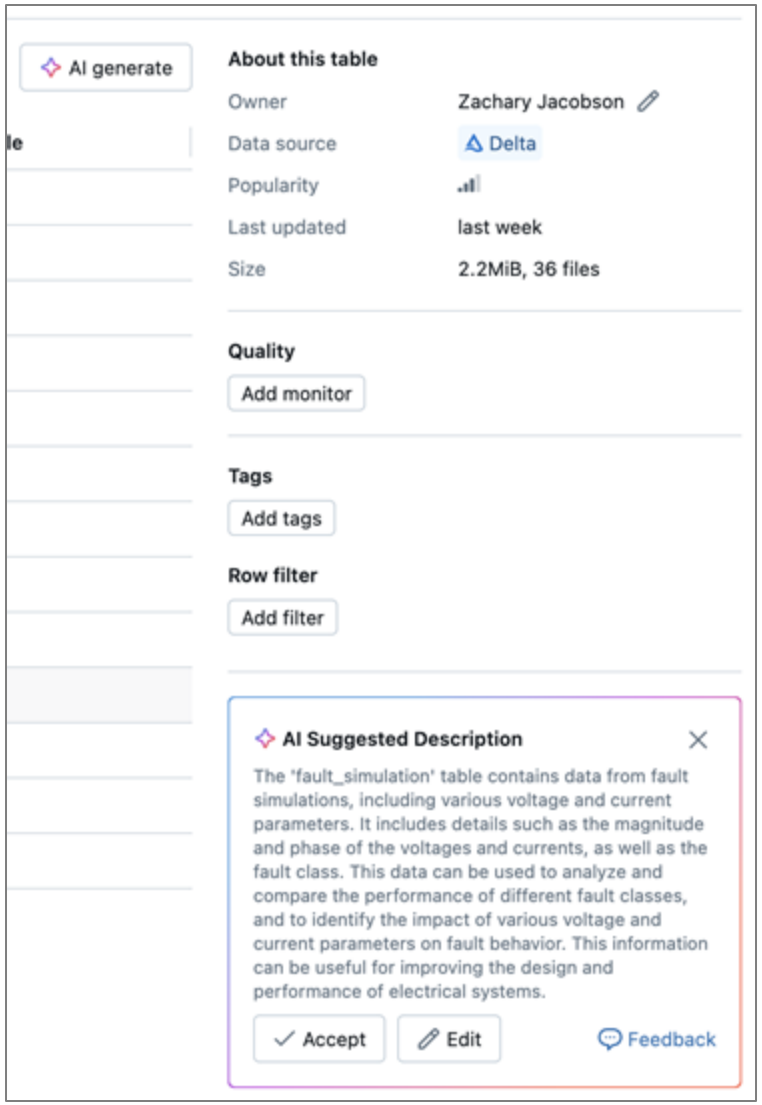


Import MATLAB Packages/Model and Use Pandas UDFs



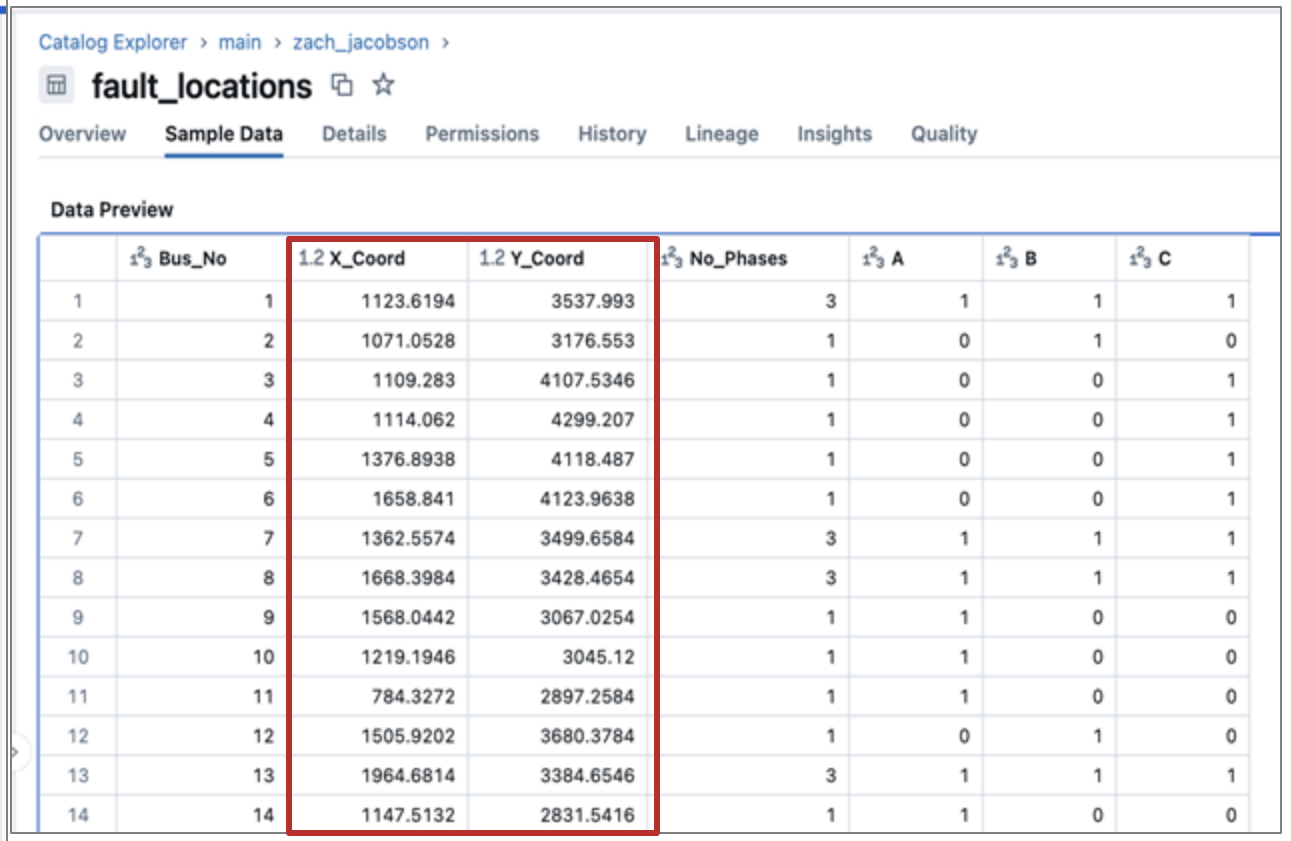
Govern Output of MATLAB Model in Unity Catalog

Column	Type	Comment
V1mag_13	double	
V1ph_13	double	
V2mag_13	double	
V2ph_13	double	
V0mag_13	double	
V0ph_13	double	
I1mag_1	double	
I1ph_1	double	
I2mag_1	double	
I2ph_1	double	
I0mag_1	double	
I0ph_1	double	
fault_class	string	



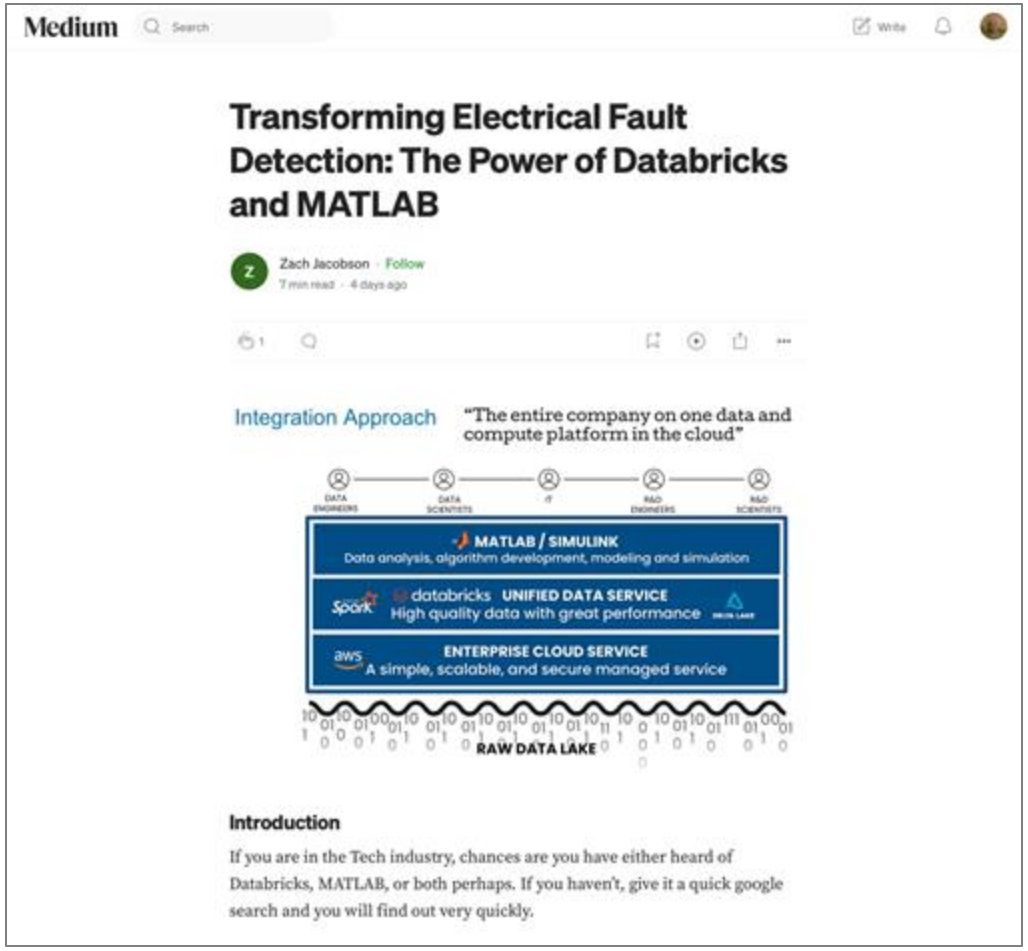
Output continued…

1.2 I0ph_1	fault_class
-43.76099147279311	unfaulted
-58.96895313025194	CB3_1_a_g
-28.701253657066538	CB3_1_b_g
-46.339389547752276	CB3_1_c_g
-44.62491344478943	CB3_1_a_b_g
-55.222329953611066	CB3_1_a_c_g
-37.112247990877904	CB3_1_b_c_g



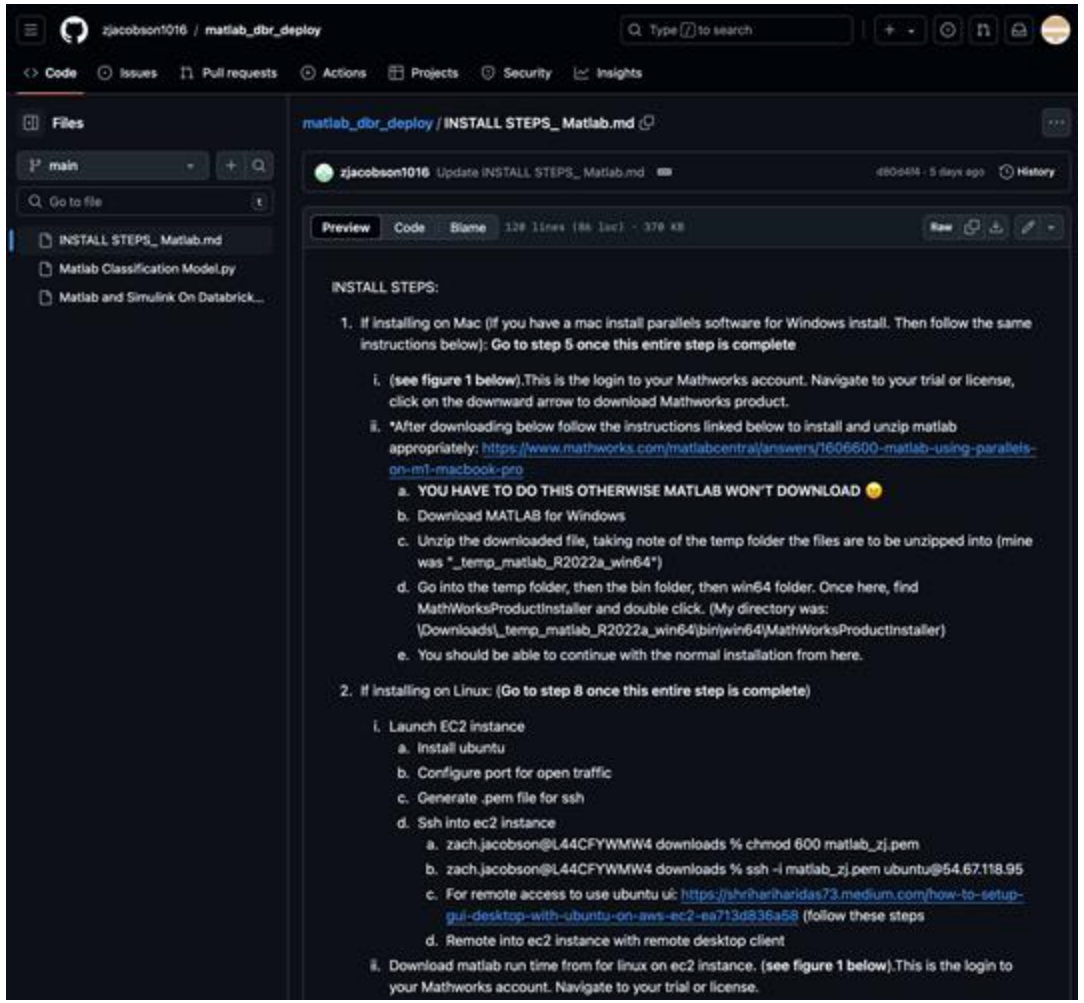
	Bus_No	X_Coord	Y_Coord	No_Phases	A	B	C
1	1	1123.6194	3537.993	3	1	1	1
2	2	1071.0528	3176.553	1	0	1	0
3	3	1109.283	4107.5346	1	0	0	1
4	4	1114.062	4299.207	1	0	0	1
5	5	1376.8938	4118.487	1	0	0	1
6	6	1658.841	4123.9638	1	0	0	1
7	7	1362.5574	3499.6584	3	1	1	1
8	8	1668.3984	3428.4654	3	1	1	1
9	9	1568.0442	3067.0254	1	1	0	0
10	10	1219.1946	3045.12	1	1	0	0
11	11	784.3272	2897.2584	1	1	0	0
12	12	1505.9202	3680.3784	1	0	1	0
13	13	1964.6814	3384.6546	3	1	1	1
14	14	1147.5132	2831.5416	1	1	0	0

Blog: Transforming Electrical Fault Detection: The Power of Databricks & MATLAB



https://medium.com/@jacobsonzach23/transforming-electrical-fault-detection-the-power-of-databricks-and-matlab-0aff9f2ab7bf

GitHub: Instructions & MATLAB Code



https://github.com/zjacobson1016/matlab_dbr_deploy/tree/main

The IEEE 123 Node Test Feeder

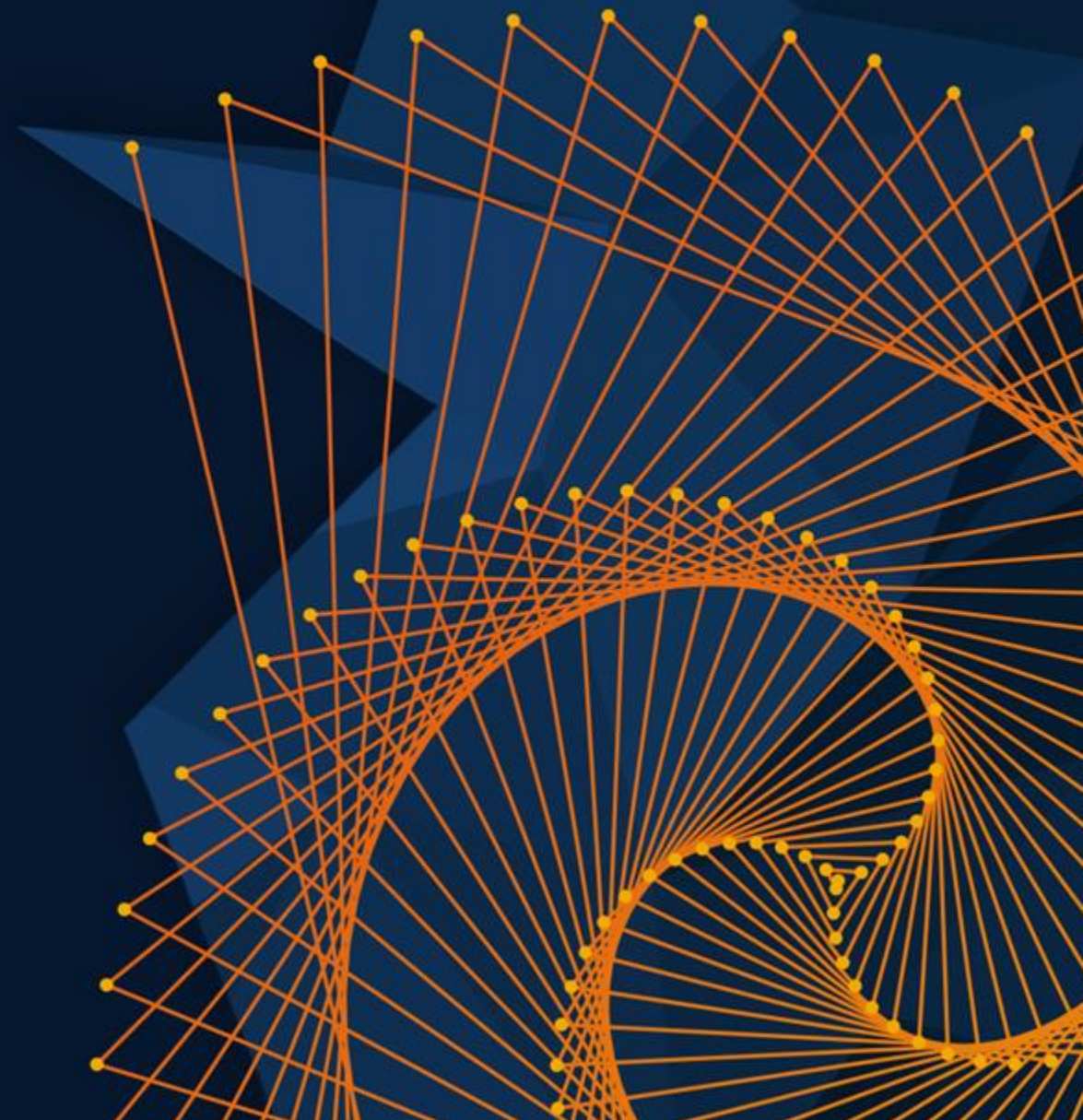
Kersting, W.H., "Radial distribution test feeders," *Power Engineering Society Winter Meeting, 2001. IEEE* , vol.2, no., pp.908,912 vol.2, 2001
doi: 10.1109/PESW.2001.916993

URL:
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=916993&isnumber=19809

Files describing the network are available at the following URL:
https://cmte.ieee.org/pes-testfeeders/

MATLAB EXPO

Thank you!



MathWorks