Synchro-Waveform Data Analytics Architecture and Big Data Platform for Grid Operations and Situational Awareness

North American Synchrophasor Initiative (NASPI) & IEEE Synchro-Waveforms Task Force, Joint Webinar

Hamed Valizadeh and Michael Balestrieri

Southern California Edison (SCE)

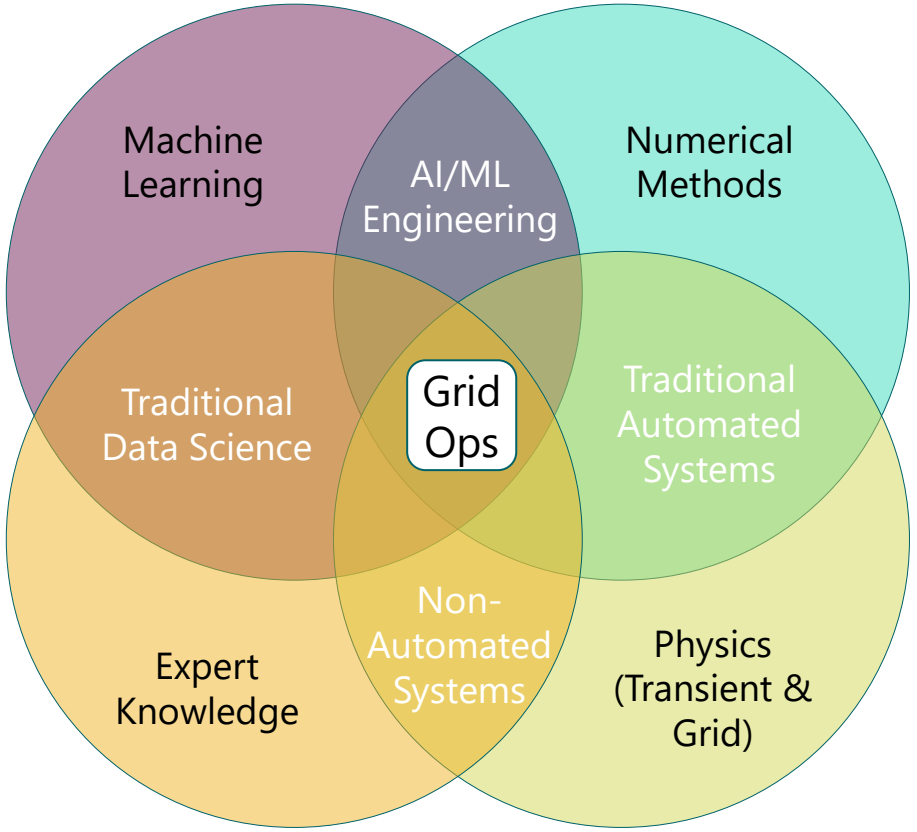
February 26, 2025

Topic 1/2: Modeling & Data Analytics

- Expert and physics augmented machine learning
- Challenges of applied waveform analytics
- Applied AI/ML components
- Anomaly detection
- Anomaly characterization
- Anomaly location identification
- AI/ML guidelines for waveform analytics

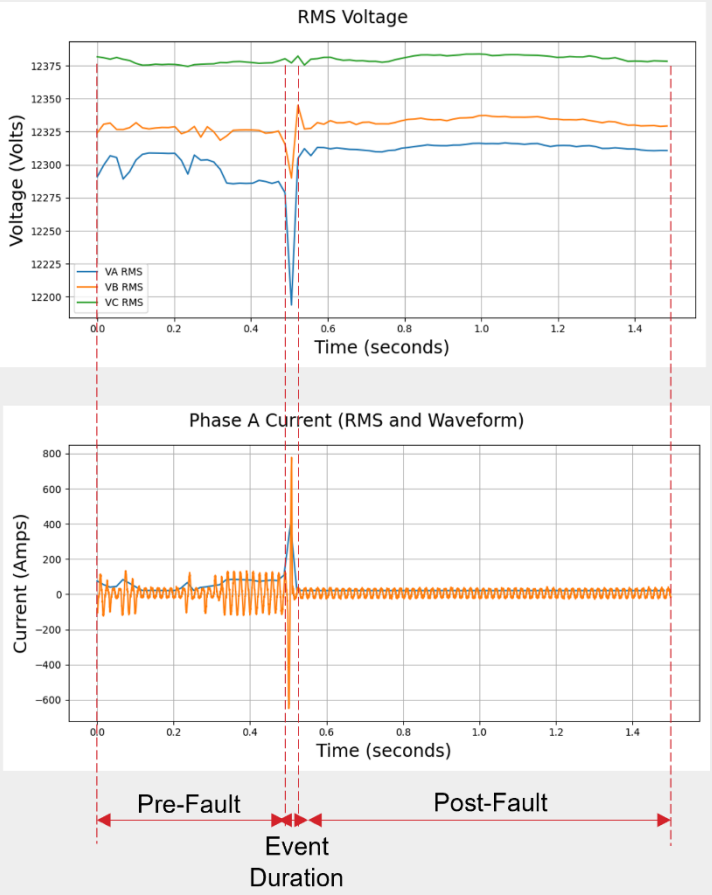
Expert and Physics Augmented Machine Learning

- ML model success in Grid Ops is often limited by the quality and quantity of available data, while its adoption is limited by the level of trust afforded by given models
- In reality, optimal learning strategy may involve combining the complementary strengths of humans, physics modeling, and machines.
- Waveforms with low information density can pose challenges for training AI/ML models. This can make it harder for the model to identify patterns and make accurate predictions.
- To address this, researchers often use techniques such as:
  - Data Augmentation
  - Expert Knowledge and System Under Study Characteristics
  - Feature Extraction
  - Simulated Data

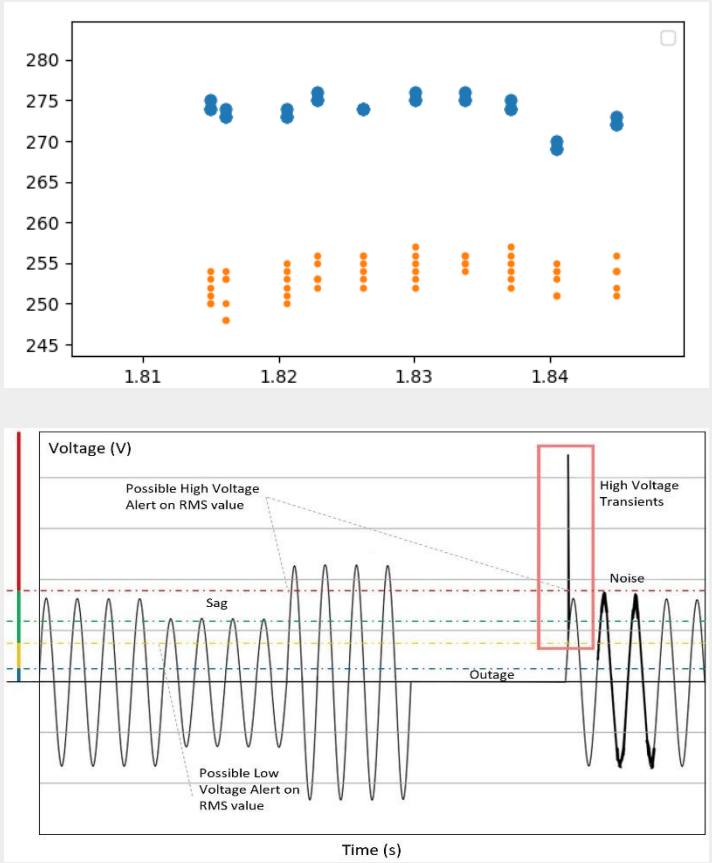


Waveform Measurement Units (WMUs) and Other Data Types

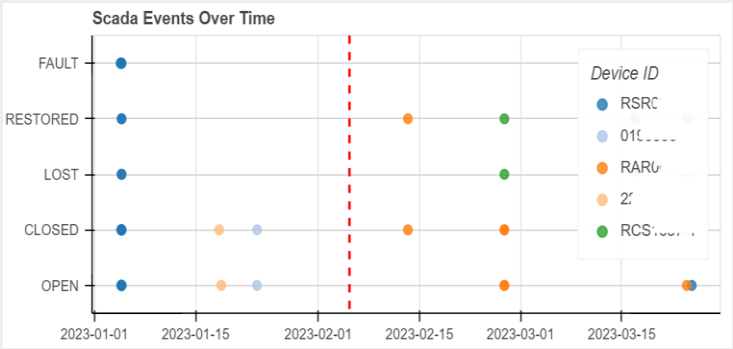
Digital Fault Recorder creates transient and extended continuous oscillography at the substation

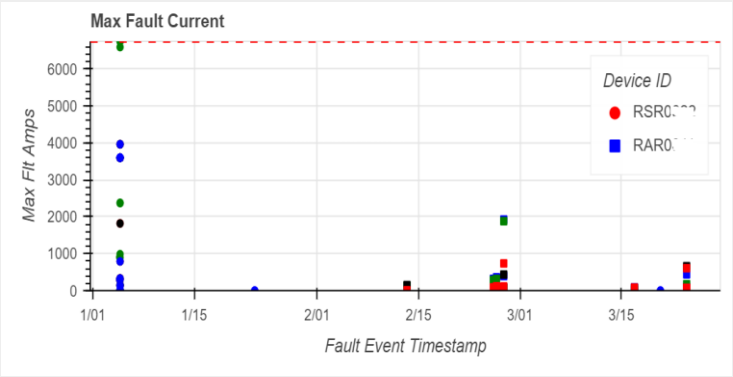


AMI voltage alerts indicate that voltage has exceeded or dipped below or above a preset threshold for at least 1 second



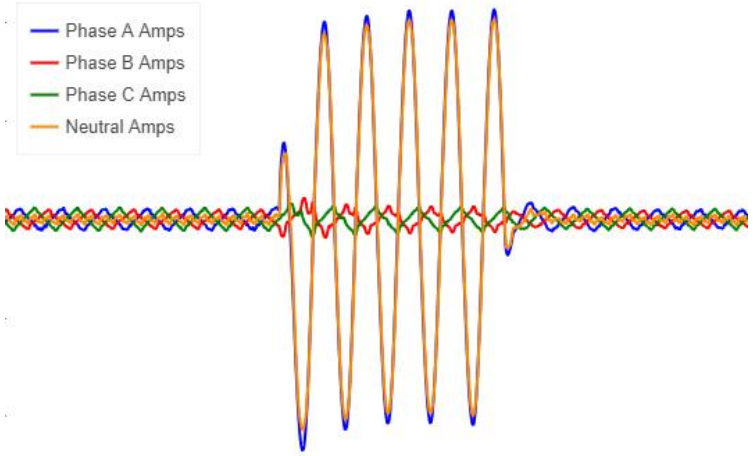
SCADA events, metadata, and fault data varies across different devices. This provides device statuses and lower fidelity fault intelligence from the field



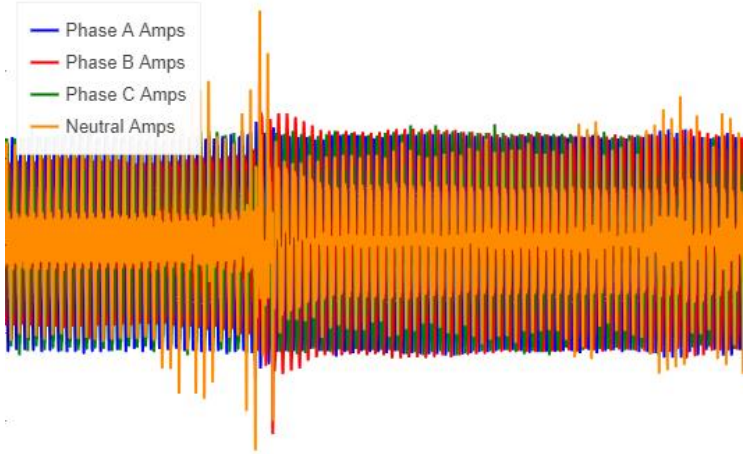


Complexities & Challenges

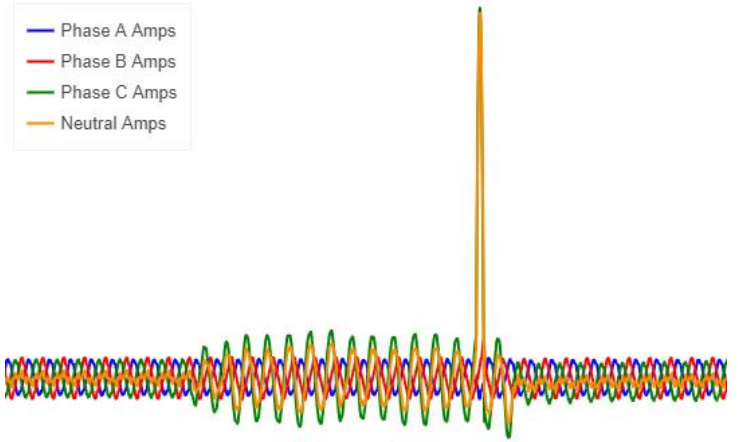
Anomaly detection and characterization of fault data from waveform measurements are challenging tasks, as the profile of a fault voltage and current can vary across different fault types and grid layouts.



Fault due to animal



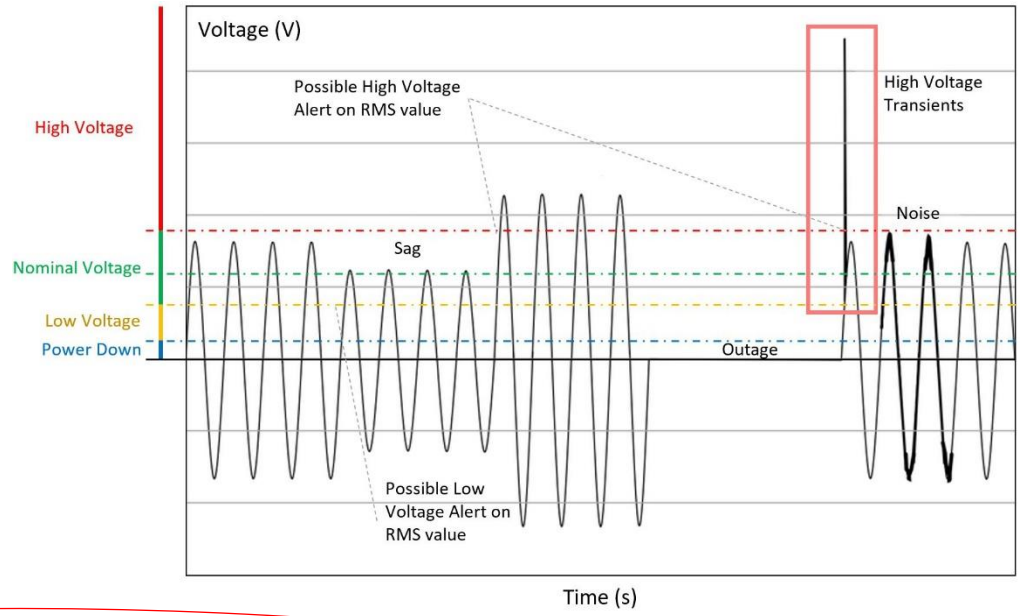
Overhead connection issue

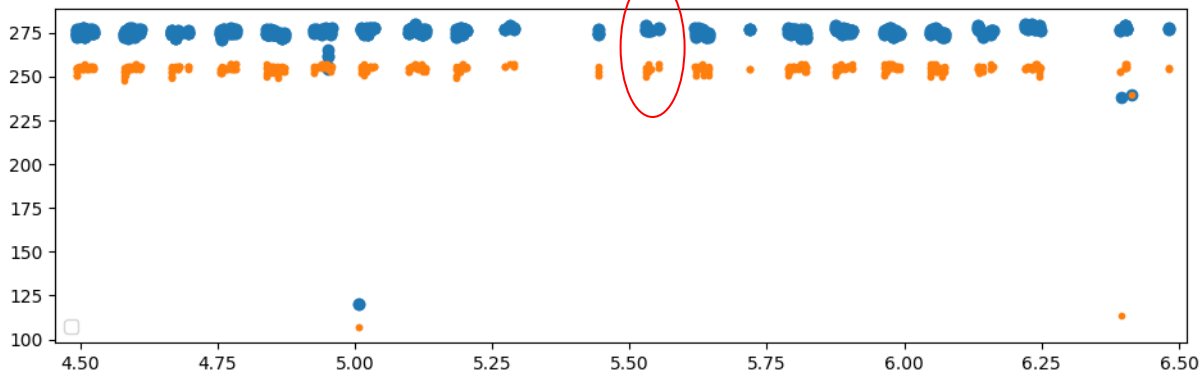


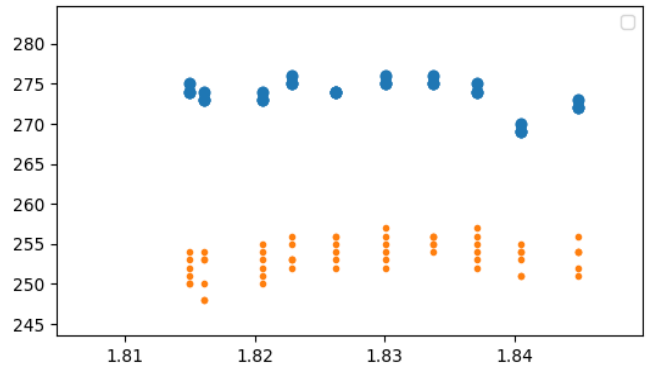
Underground incipient fault

Complexities & Challenges

- Currently available grid edge measurements are imperfect.
- A low voltage alert indicates that voltage has dipped below a preset threshold for at least 1 second
- When voltage recovers on the affected phases for at least 1 second, a phase recovery alert is sent
- These alerts are timestamped by the meter and sent back through the meter network system to the back-office system
- Downstream systems that ingest these real-time alerts make this data available for further processing.



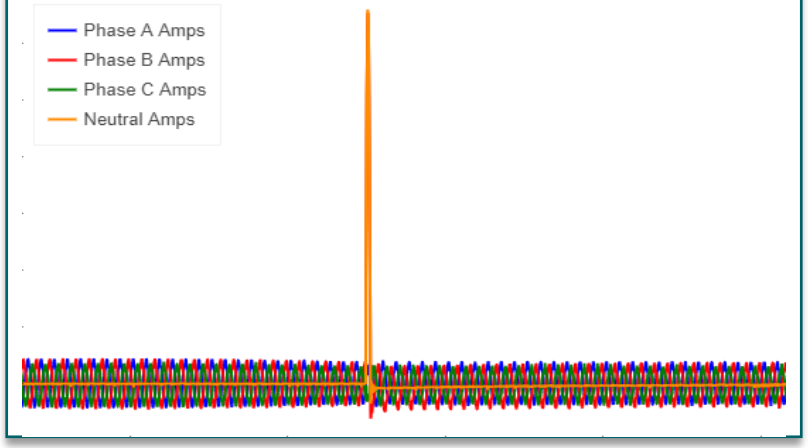




Voltage visibility from grid edge

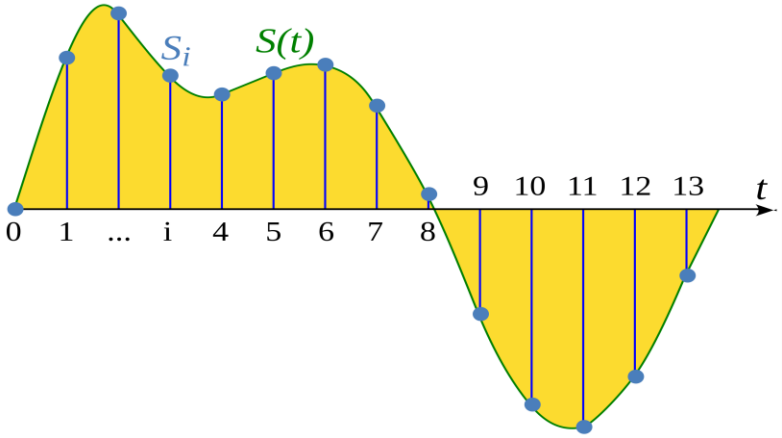
Complexities & Challenges

As incipient faults self-clear within a brief time, it is particularly challenging to identify and locate them using a sole source.
Example: The incipient fault in the underground cable is an intermittent arcing fault with short duration from 1/4 cycle to 1/2 cycles and can be cleared by itself. It always reoccurs for many times before a permanent fault.



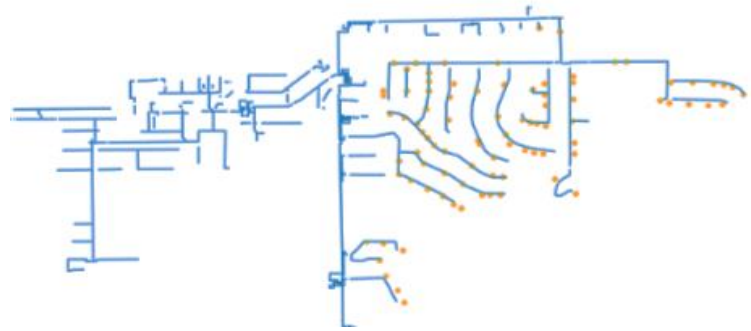
Waveform measurements with high resolutions increase the challenges of working with the data due to technological constraints.

Example: a waveform recorder produces roughly 10 GB of continuous point-on-wave data per day with event data ranging 0 to 1 GB per day.

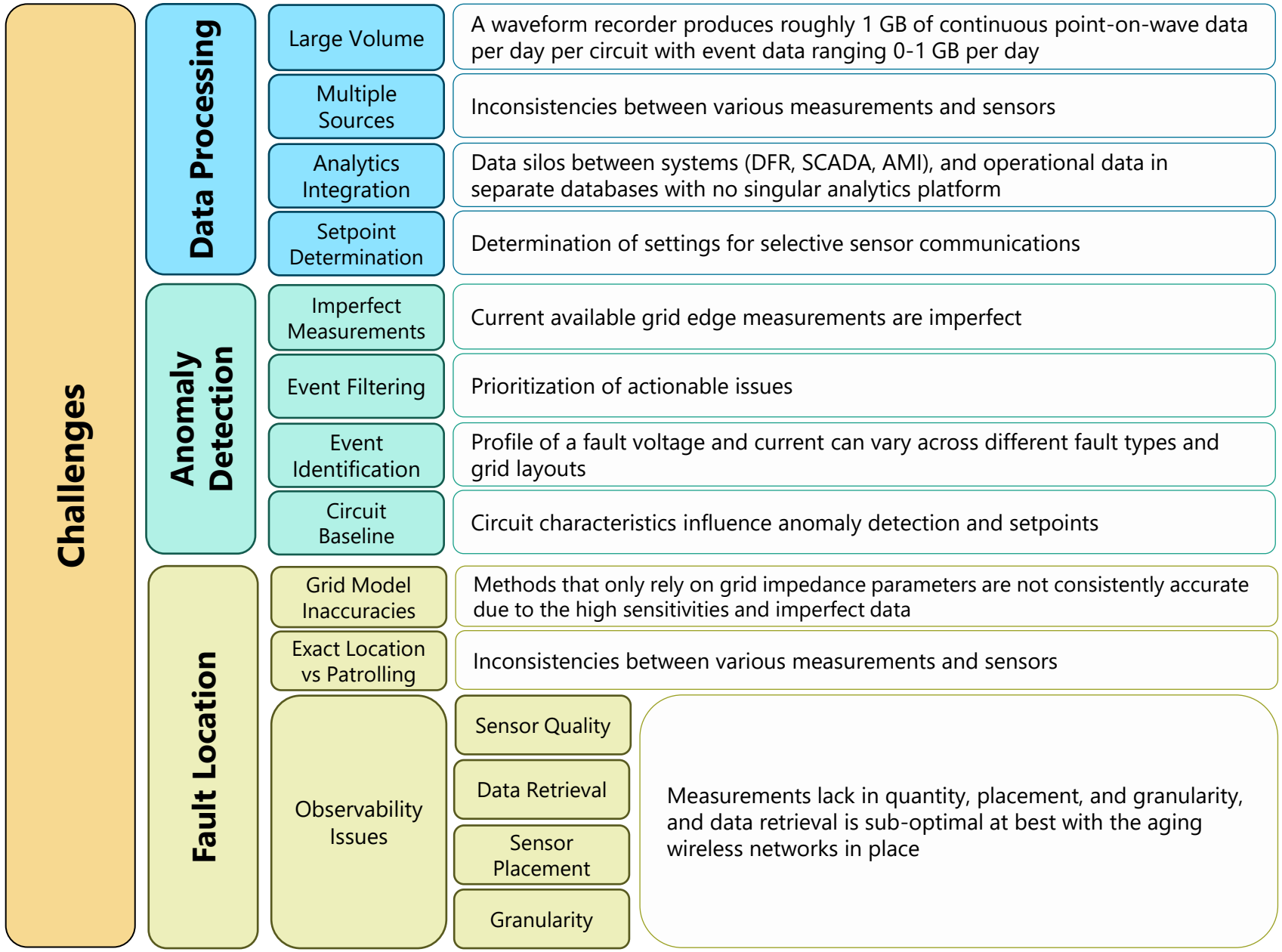


The performance of the methods that rely only on the accuracy of the grid impedance parameters are not consistent due to the high sensitivities and imperfect data management.

Example: Impedance based fault location will at best calculate an area of the circuit not a pinpointed location



Complexities & Challenges Summary

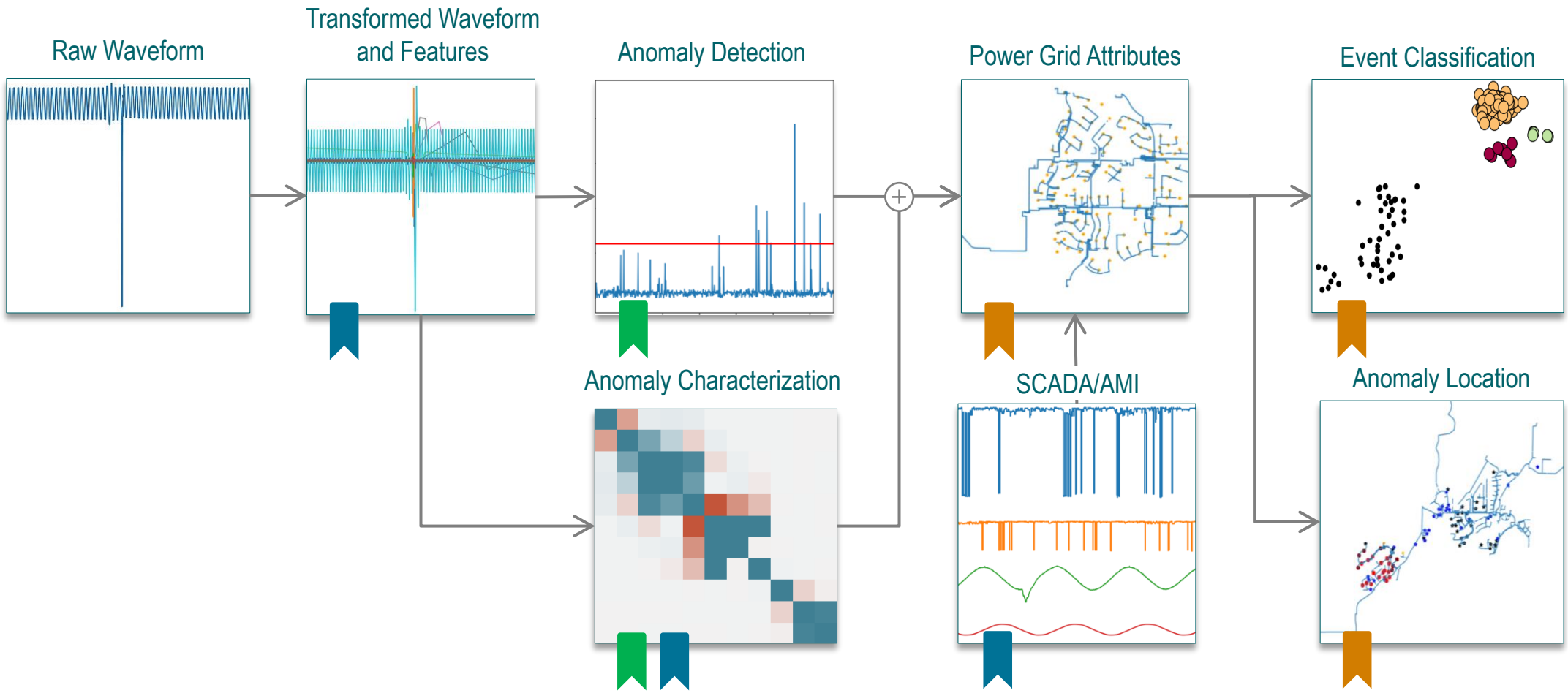


Applied AI/ML Components

Expert Knowledge

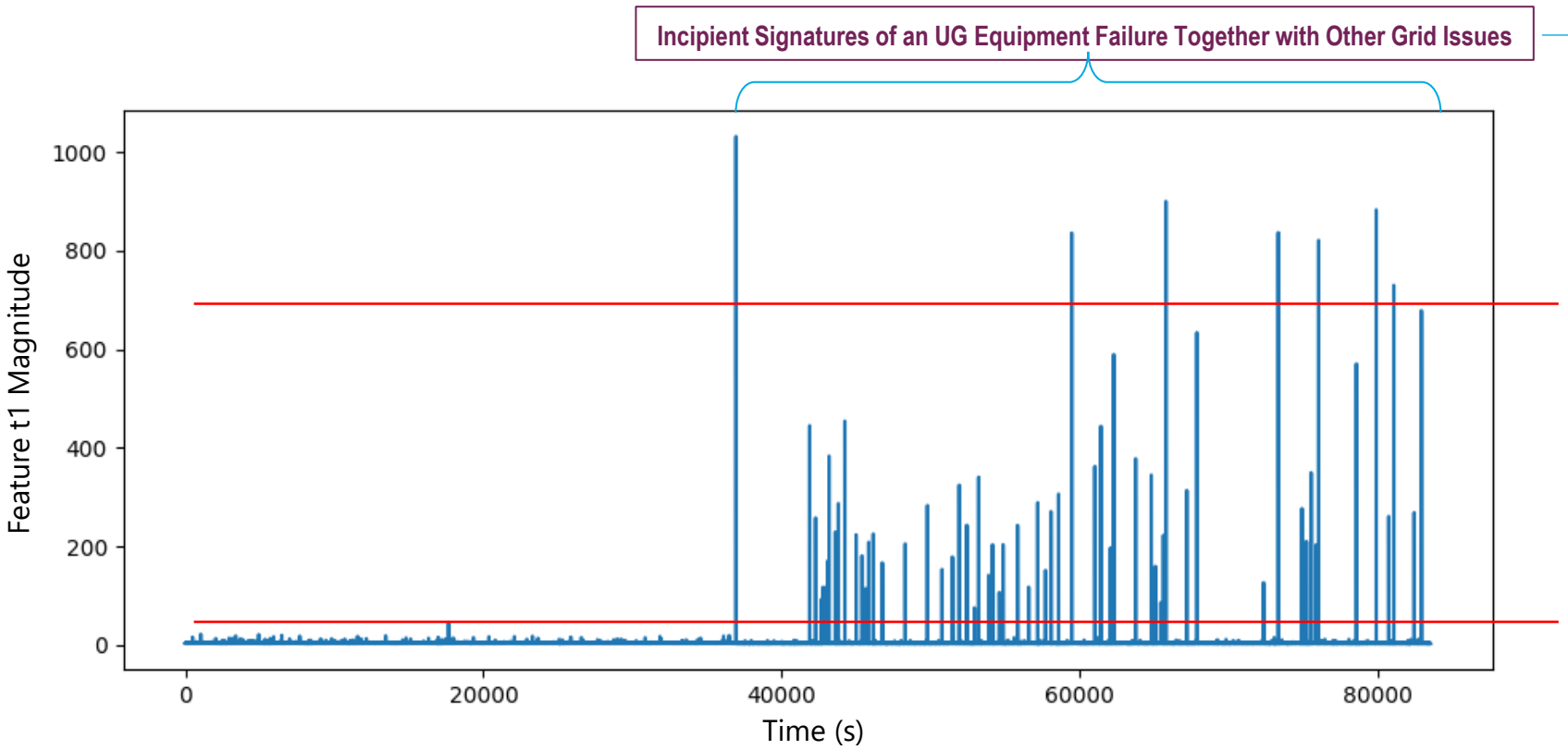
Feature Extraction

Data Augmentation



Anomaly Detection

- A threshold is intended to detect when something is abnormal. Abnormalities aren't always problems.
- Any alert you set on a metric exceeding what you think is normal is going to fire a lot.
- A monitoring system needs to know the difference between an unusual state and a real problem, and this isn't possible with only a threshold.

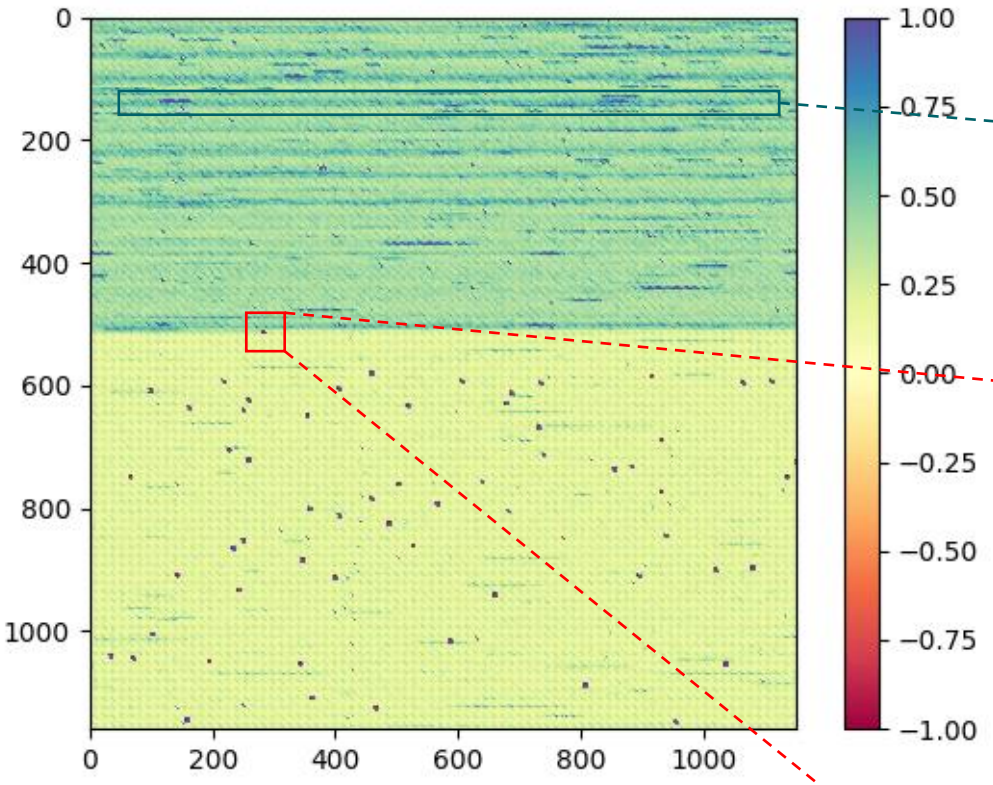


Features have dynamic range to allow for setting/using varying levels of thresholds for varying levels of actionability (basic filtering of anomaly and event-types)

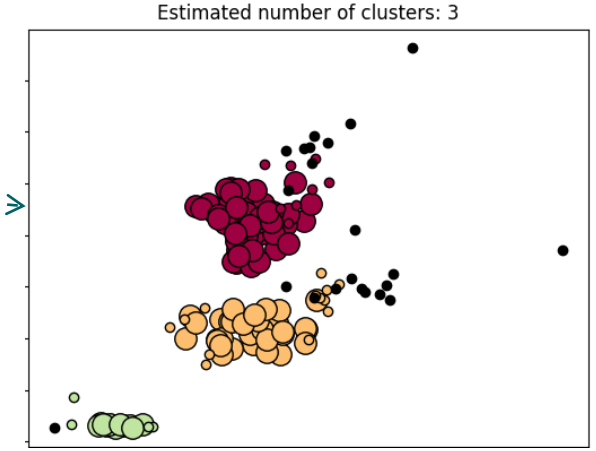
Features are very sensitive to anomalies (anomaly characterization prevents repeated triggering)

Anomaly Characterization

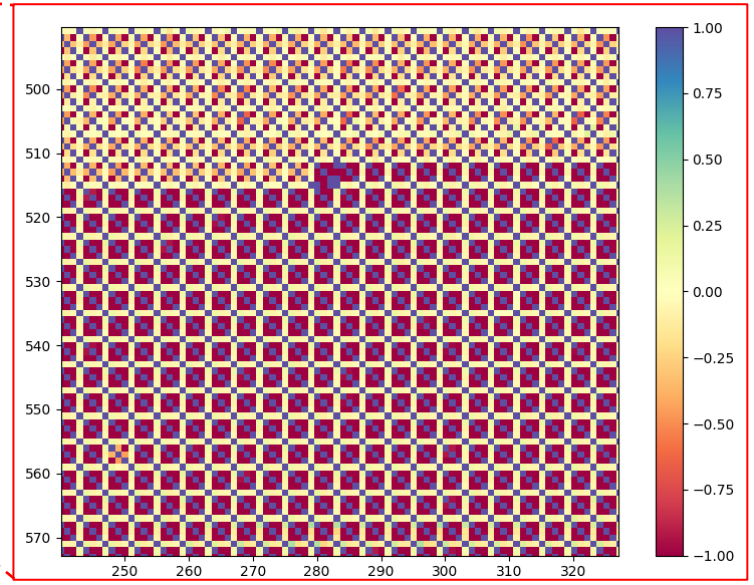
- Anomaly characterization (by temporal clustering) is developed together with the anomaly detection.



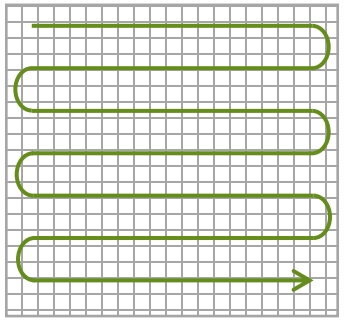
Temporal Clusters of the Selected Area (Time Period)



Advanced Features for ML-based Anomaly Characterization

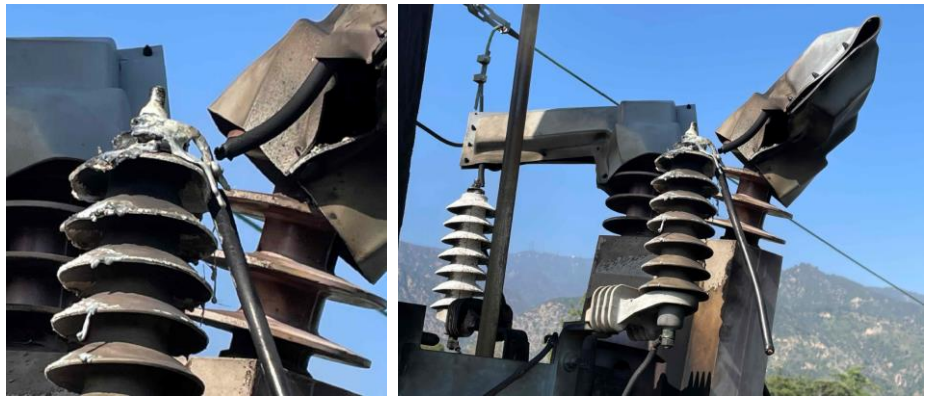


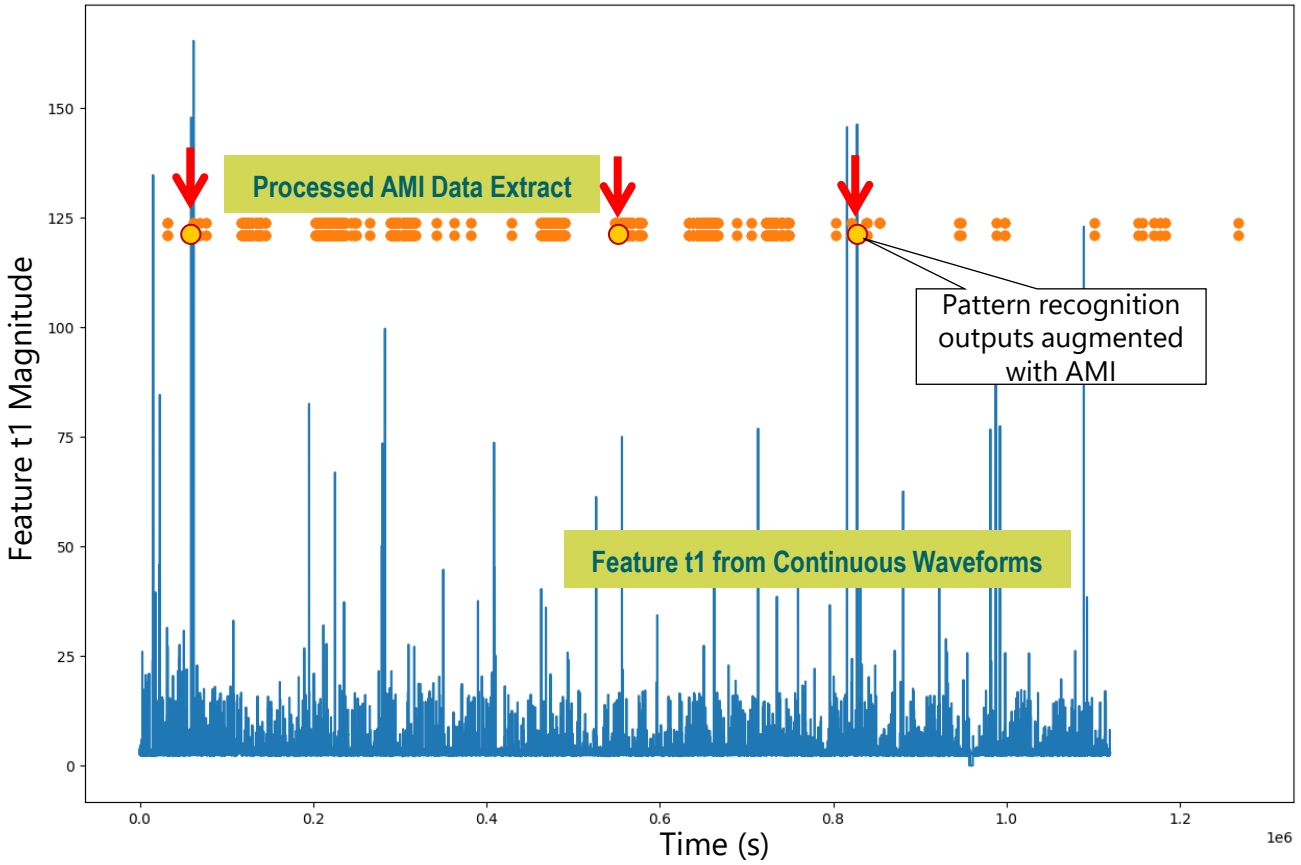
Visualization Guide



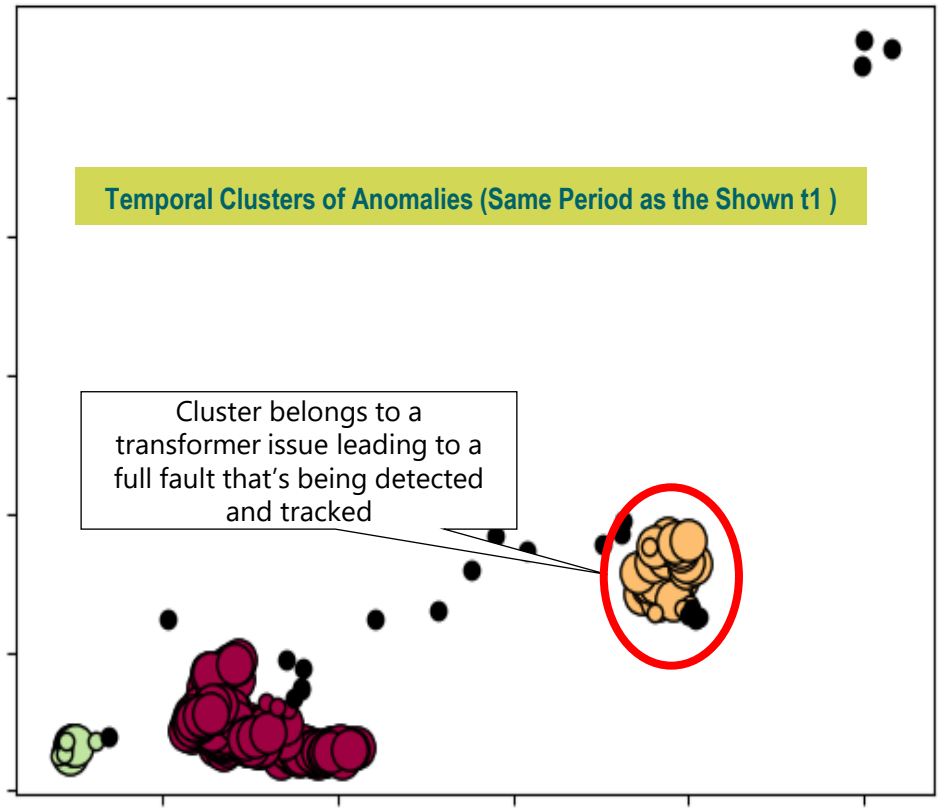
Anomaly Location

- Waveform snippets can replace continuous point on wave (CPOW) recording if analytics and sensor settings are coordinated
- In practice, clustering without physics modeling and system characterization may fail.





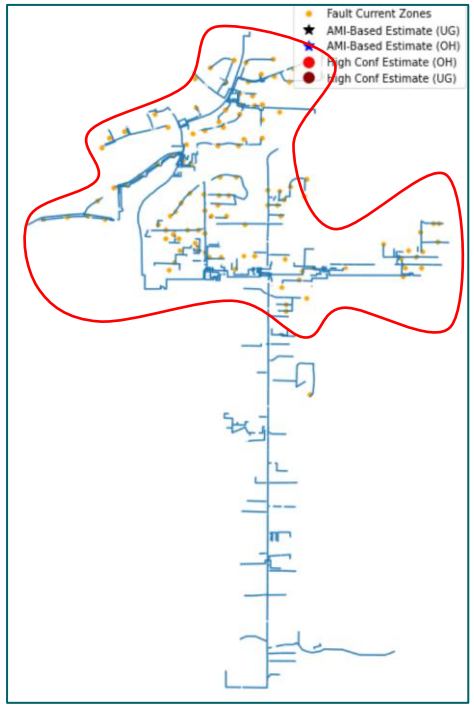
Estimated number of clusters: 3



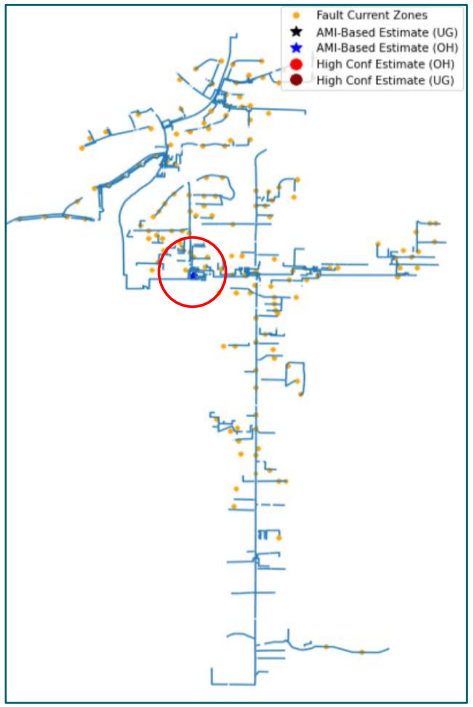
Actionable Anomaly Location

- AMI-based abnormal voltage recordings help pinpoint fault location via spatiotemporal modeling of imperfect and sporadic measurements
- Frequency of incipient signature occurrence increases over the course of the failure as time passes.

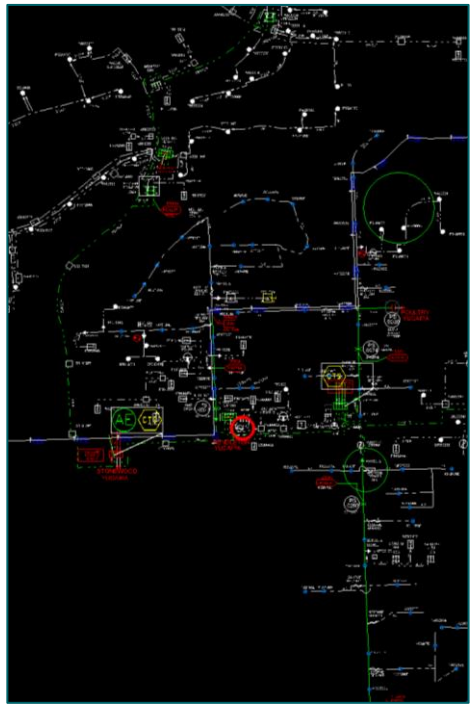
Estimated area ~90 days ahead



Estimated structure(s) few days ahead



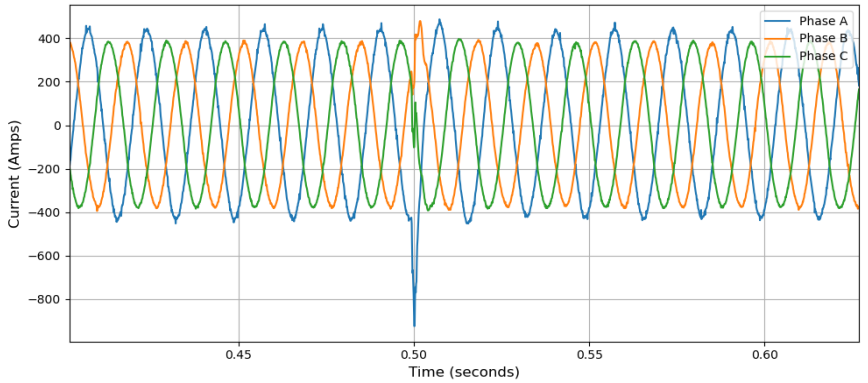
Estimated structure to dispatch



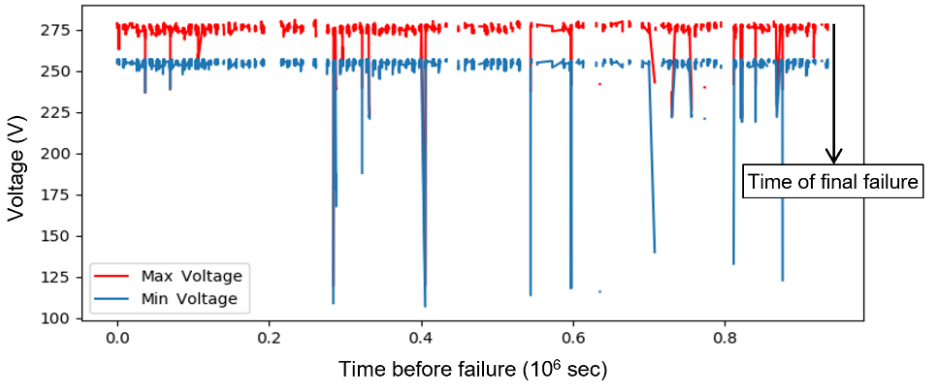
Failing component

Actionable Anomaly Location – Continued

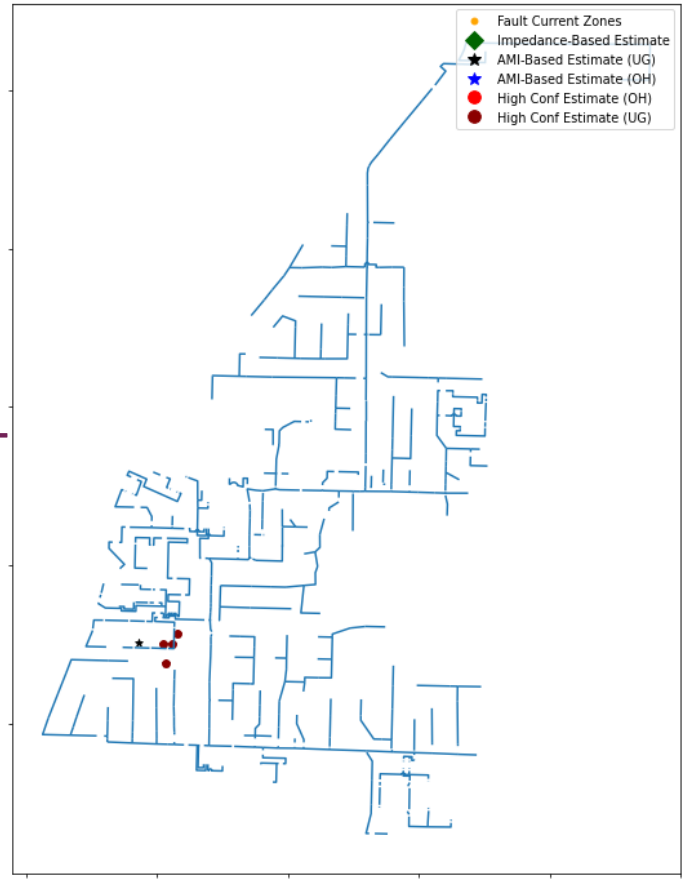
A waveform signature prior to the failure



Smart meter exceptions data aligned with the modeled waveforms to help locate the issue

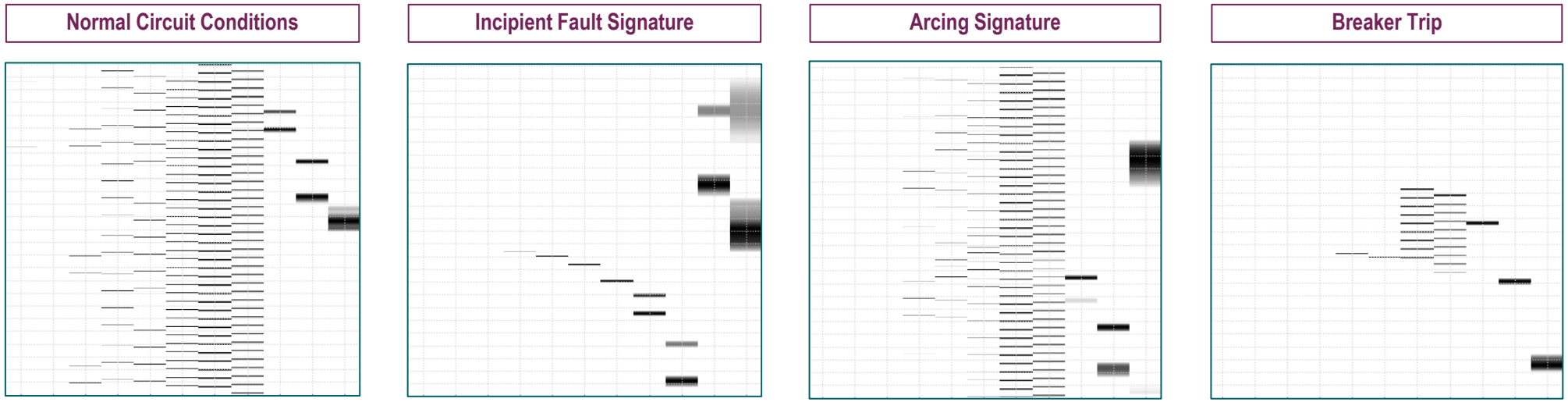


Estimated location



AI/ML Guidelines for Waveform-Based Detections – Utilities Summary

- **Enhancing Model Performance**: By crafting features that capture the essence of the event, we enable our models to extract meaningful patterns and relationships from the data, leading to more accurate insights.
- **Improving Model Interpretability**: Engineering features that are interpretable, enables transparency and is essential for building trust in machine learning systems and making informed decisions based on model outputs.
- **Handling Complex Data**: Distribution grid data is rarely clean. It often contains missing values and outliers that can hinder model performance. Expert-augmented analytics are key.
- **Minimal Need for Labeled Data**: Well-engineered features minimize the requirements for labeled signature libraries.
- **Implementation:** Detection of event location and protection system awareness are key. Moving from "traditional protection" to "anomaly recognition"



Topic 2/2: Data Platform & Architecture

- Problem Statement
- Platform Components and Architecture
- DFR Data Ingestion
- DFR Database Design
- Utilizing Apache Spark
- Visual Application Architecture and User Interface
- Next Steps

Problem Statement

Waveform data is not managed in a fashion that facilitates developing and deploying data analytics.

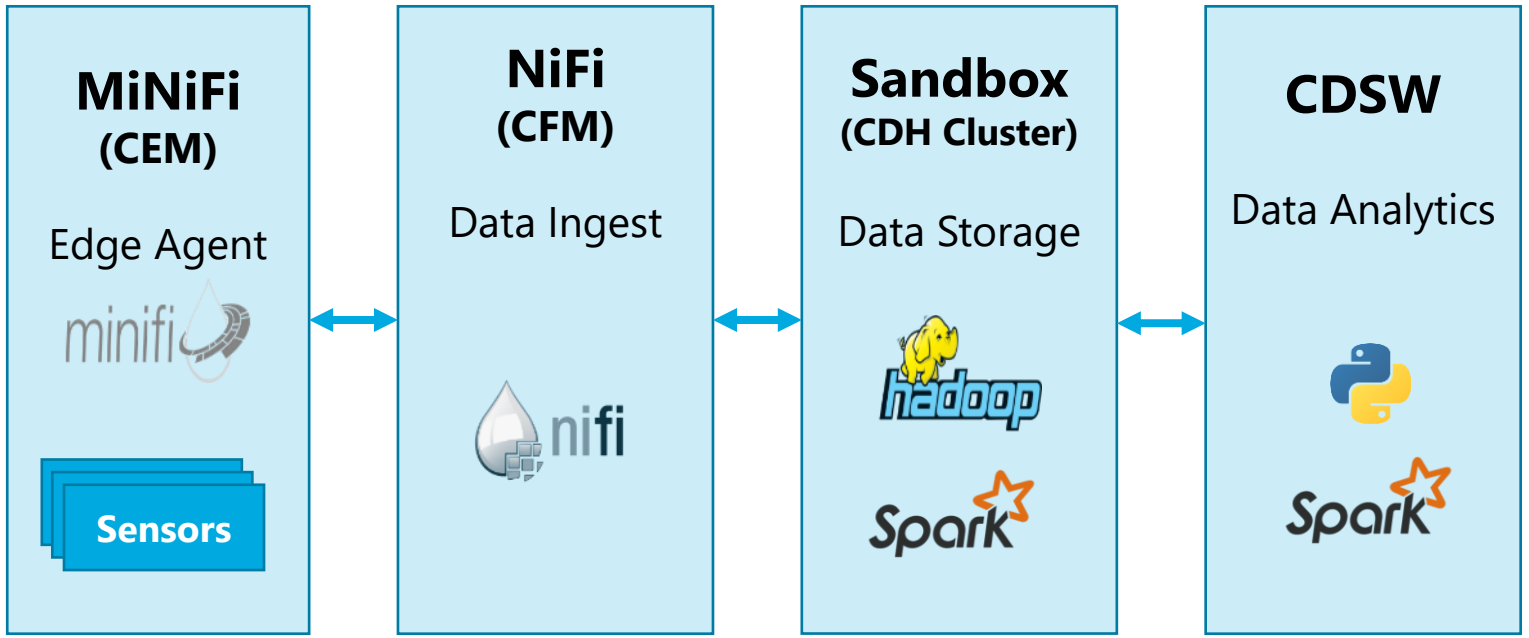
Business Requirements:

- Data platform must support automated data ingest pipeline
- Handle TB scales of data minimum for pilot
- Support development *and* deployment of data analytic pipelines
- Flexibility in solution design allotted to development team

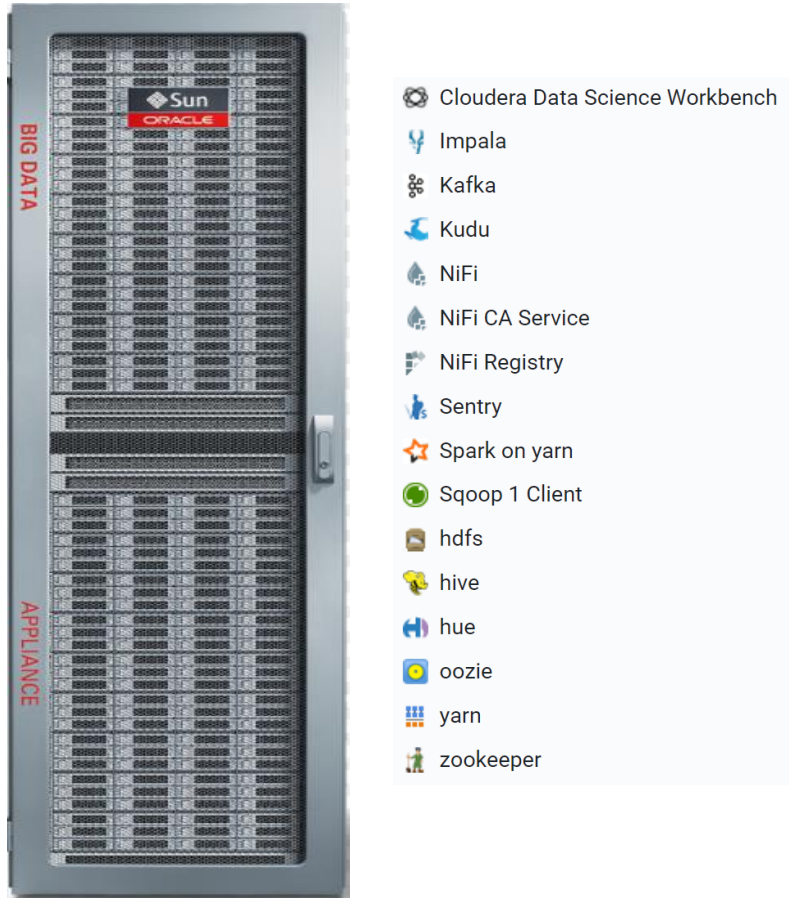
Options considered:

- Purchase vendor-built platform for waveform data
- Use existing infrastructure and design our own platform
- Cloud vs On-Prem deployment

Platform System Components (Cloudera)



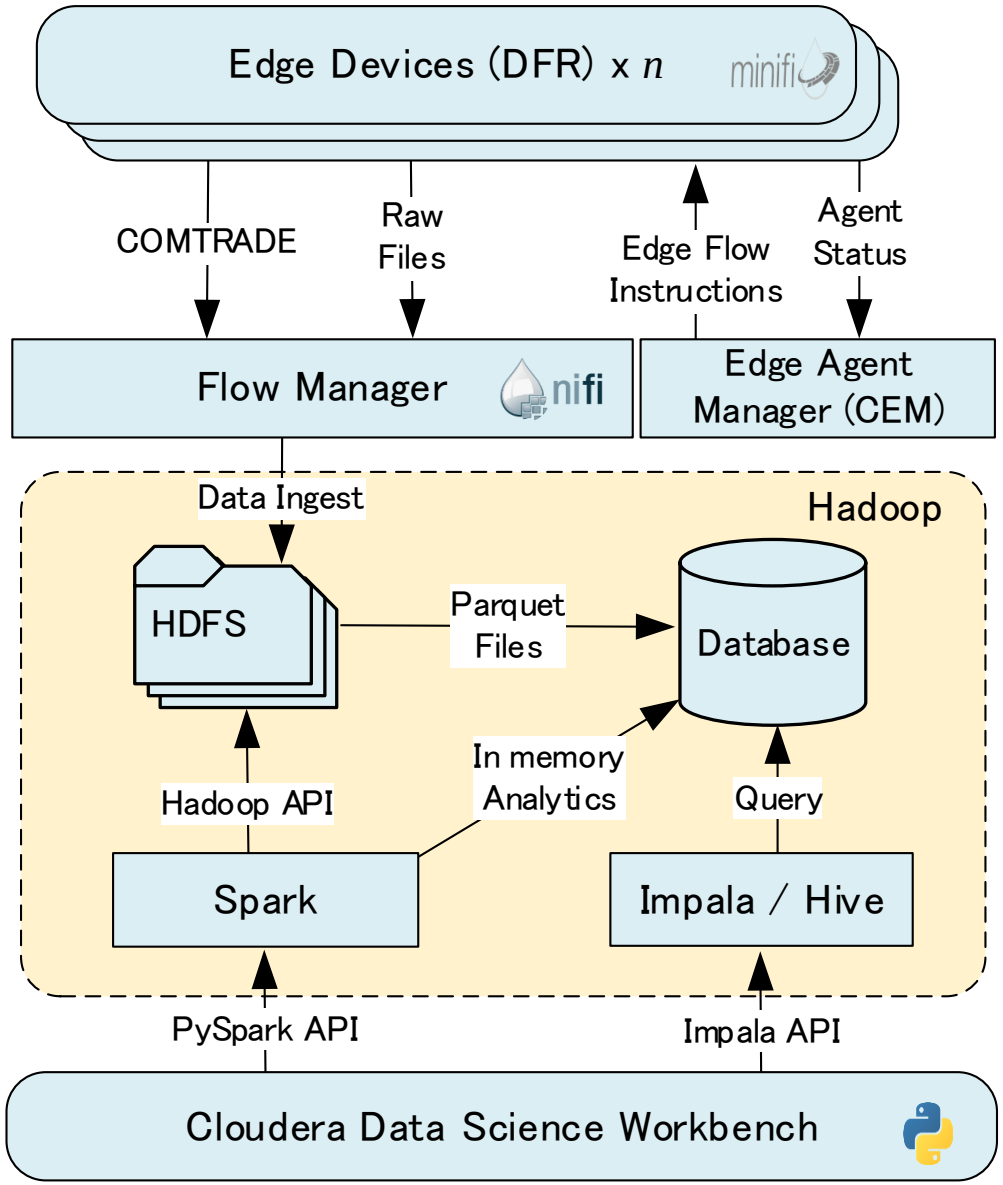
On-Premise Big Data Appliance running CDH (Old Version)



Platform Architecture (DFR ONLY)

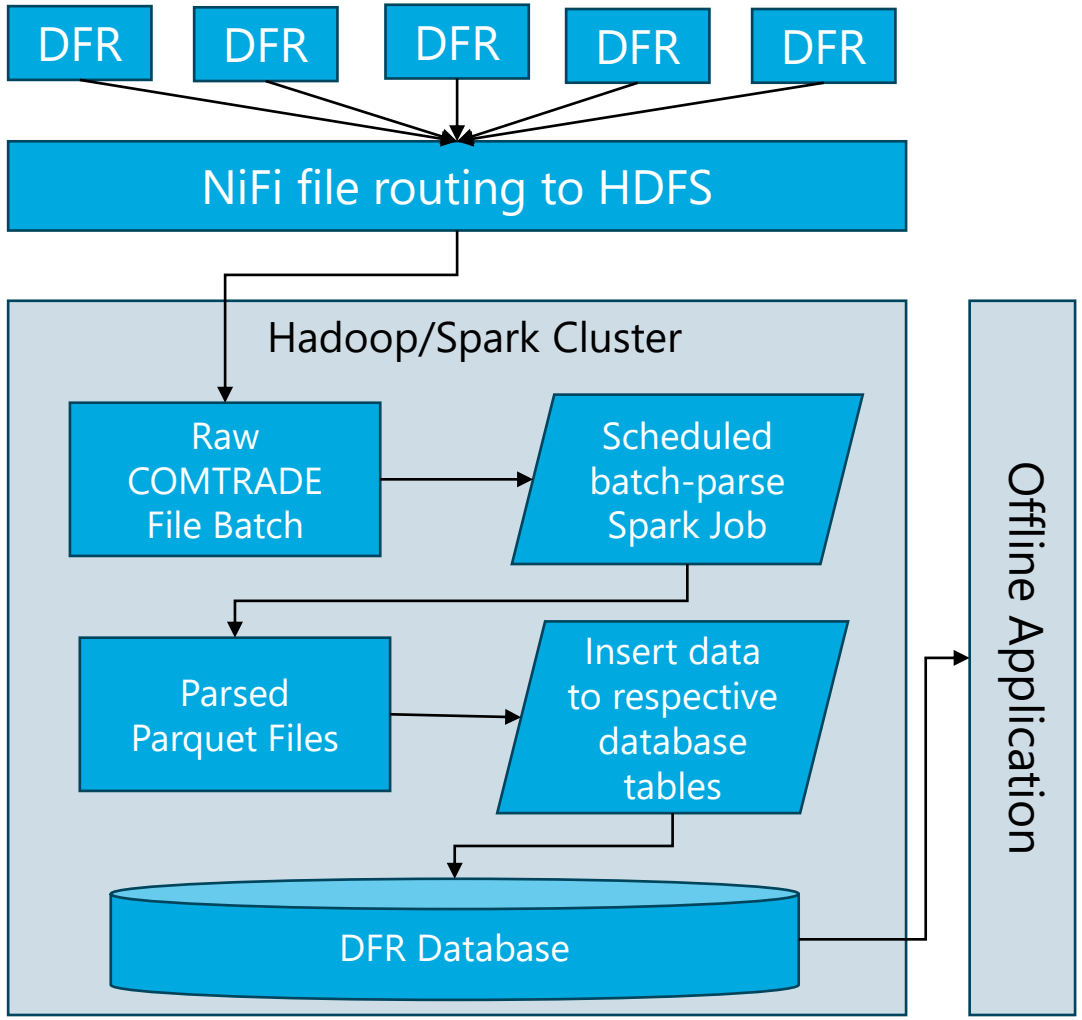
Components

- Edge Agent (MiNiFi)
- Edge Agent Manager (CEM)
- Data Flow (NiFi / CFM)
- Distributed File System (HDFS)
- Query Engines (Impala/Hive)
- Distributed Compute (Spark)
- Analytics Workbench (CDSW)
- Web Application (Streamlit)



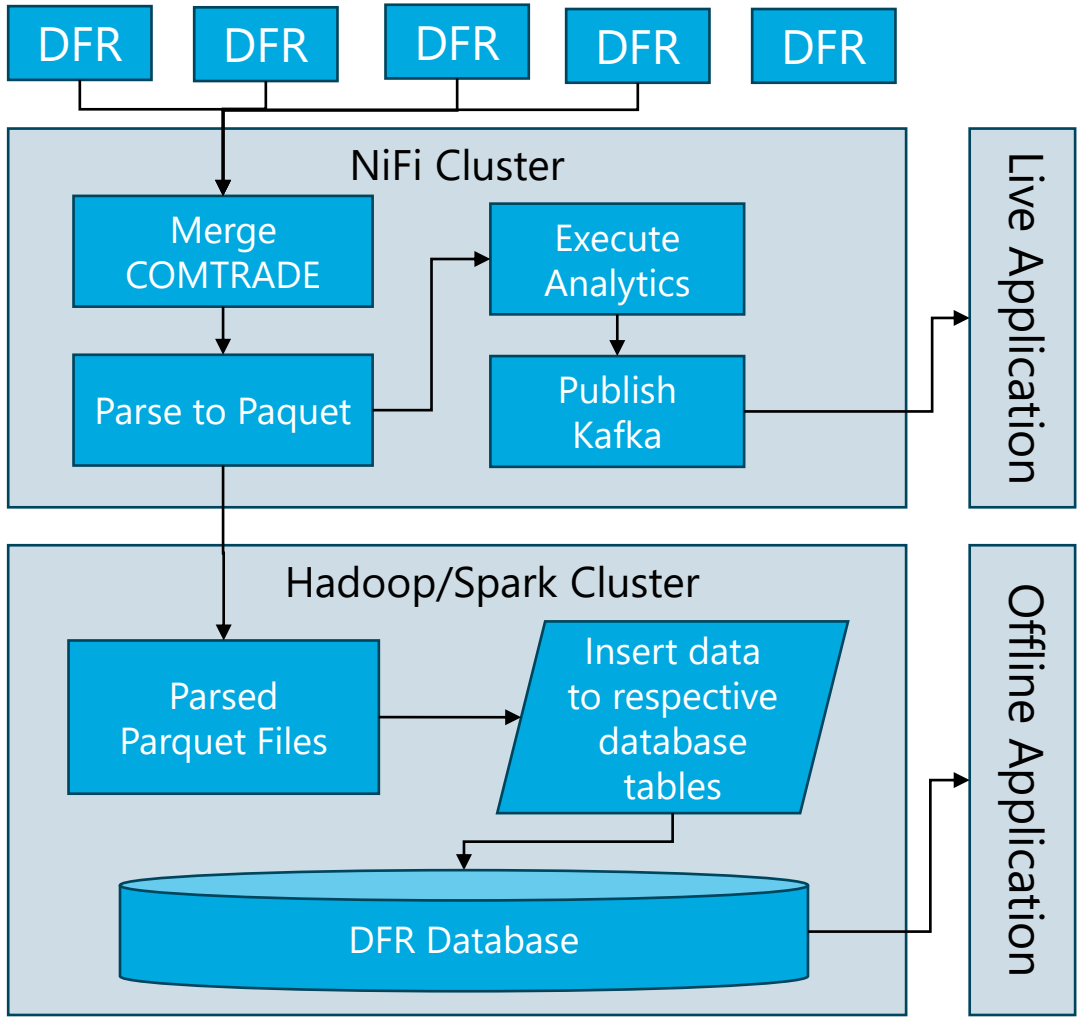
DFR Data Ingest: Write then Parse Later

- NiFi interfaces between DFR and HDFS to route files
- Batch Spark job reads new events, parses to parquet, inserts to database tables
- This approach adds time to discovering high priority anomalies
- Challenges with COMTRADE



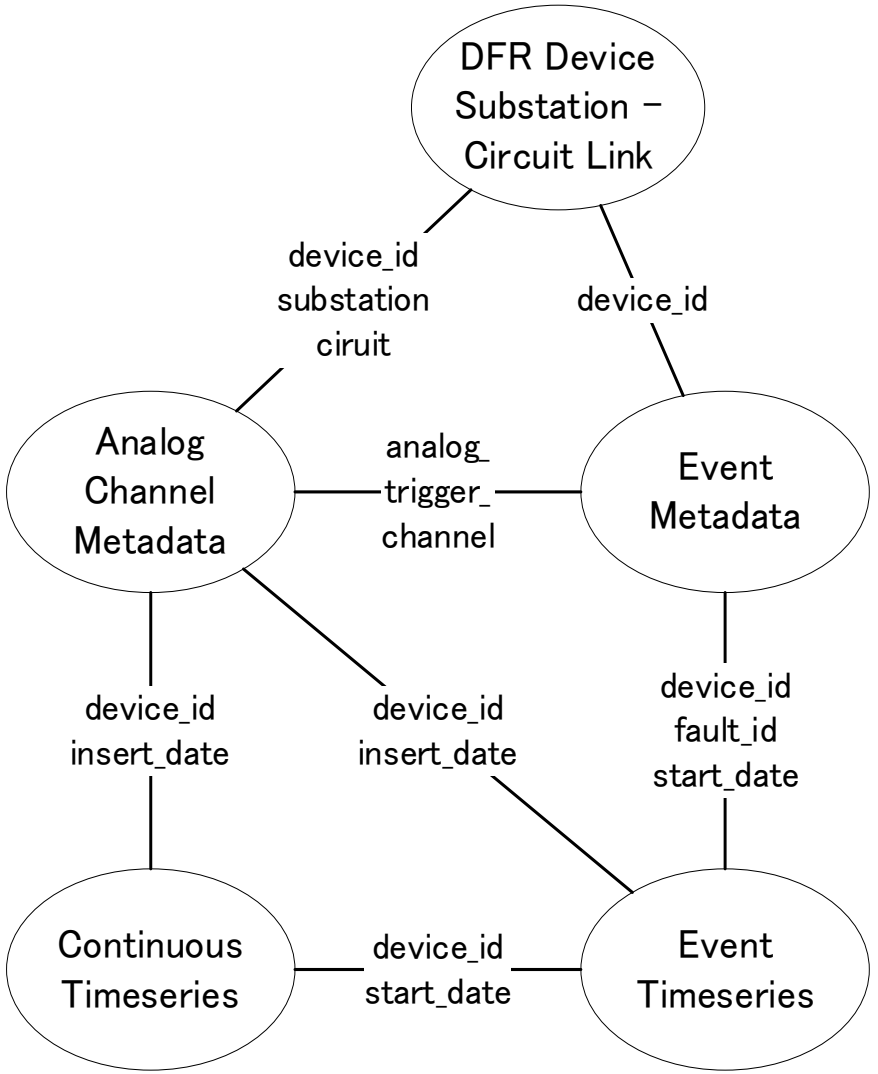
DFR Data Ingest: Parse and Analyze in Motion

- Raw files are parsed to parquet format
  - COMTRADE has pair/triplet files
  - PQDIF all-in-one file
- Parquet files are sent to database or consumed immediately for live use-cases
- Limitations exist for processing capabilities in NiFi
- *…why Parquet?*



DFR Database Schema Design

Table	Description
DFR Device Substation-Circuit Link	Links substation and circuit names to DFR device ID
Analog Channel Metadata	Links DFR channel ID's (A1, A2, A3…) to channel name, units, phase, and insert date.
Event Metadata	Contains all metadata associated with COMTRADE standard
Event Timeseries	Contains the analog channel timeseries from COMTRADE records
Continuous Timeseries	Contains the analog channel timeseries from continuous records



Exploiting Python for Usability

```
> import dev.dfr_ckt as dwa
```

Initialize the Dft_Ckt object with a circuit name

```
> ckt = dwa.Dfr_Ckt(ckt_name='BURNT MOUNTAIN')
```

Load the circuit DFR metadata

```
> ckt.load_meta()
```

Now the metadata is loaded as object attributes

```
> ckt.current_ids
```

```
['A20', 'A21', 'A22', 'A23']
```

```
> ckt.current_phases
```

```
['Ia', 'Ib', 'Ic', 'Ir']
```

Set some start & end timestamps

```
> start_ts = '2022-12-11 22:06:42'
> end_ts = '2022-12-11 22:06:44'
```

Load the timeseries data

```
> ckt.load_amps(start_ts, end_ts)
> ckt.timeseries[:3]
```

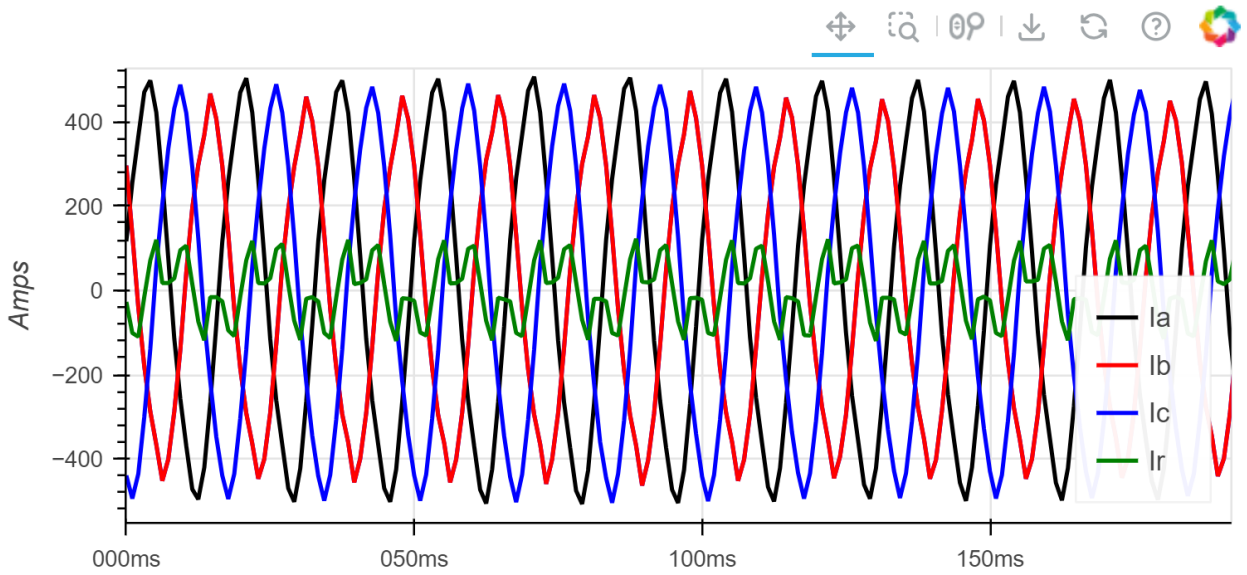
```
array([datetime.datetime(2022, 12, 11, 22, 6, 42),
       datetime.datetime(2022, 12, 11, 22, 6, 42, 1041),
       datetime.datetime(2022, 12, 11, 22, 6, 42, 2083)], dtype=object)
```

```
> ckt.amps[:3,:]
```

```
array([[272.959108, -51.360167000000004, -222.980066, 7.517735999999999],
       [206.59749, -111.48914300000001, -86.436093, 12.52956],
       [169.03431, -192.913798, 30.064728, 21.300252]], dtype=object)
```

Plot the time series

```
> ckt.plot_amps().show()
```

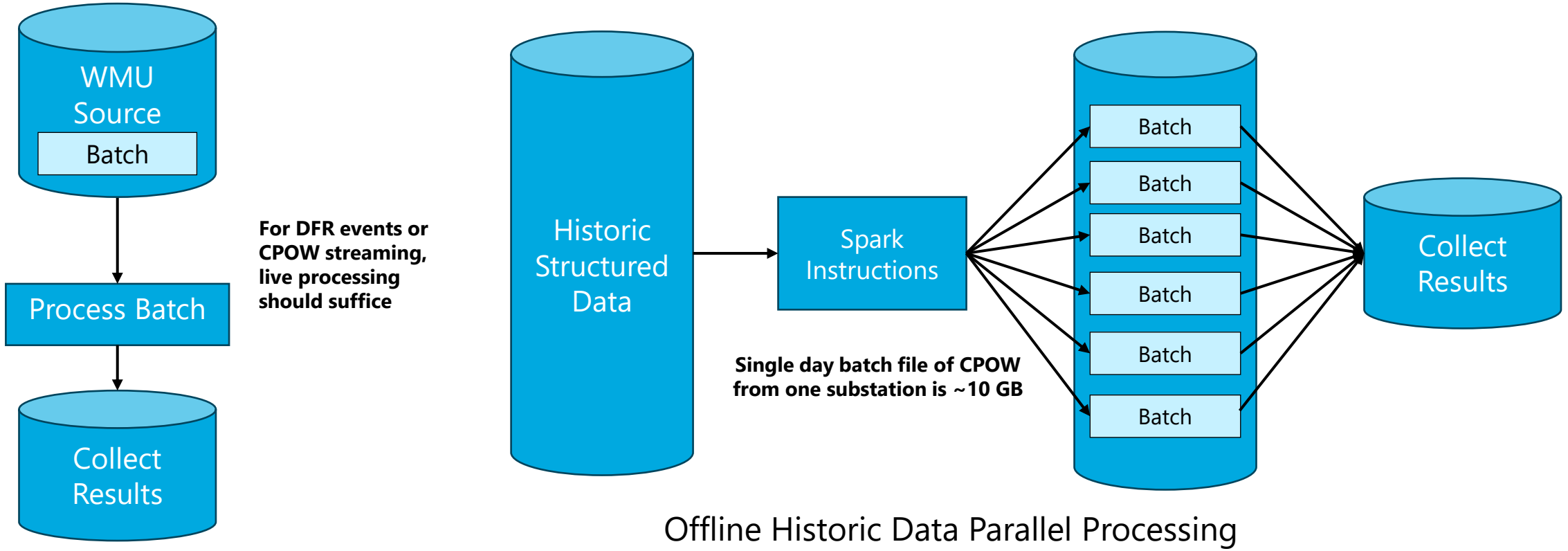


Python Modules in Use:

Impyla:	Impala Database API	**SciPy:**	Signal Processing
Pandas:	Python Dataframes	**PySpark:**	Big Data Processing
NumPy:	Numic Array Operations	**Bokeh:**	Interactive Plotting

Utilizing Spark for Waveform Data Processing

Small batch or stream processing may be able to keep up with real-time data creation, while Spark helps parallel process large historic datasets.

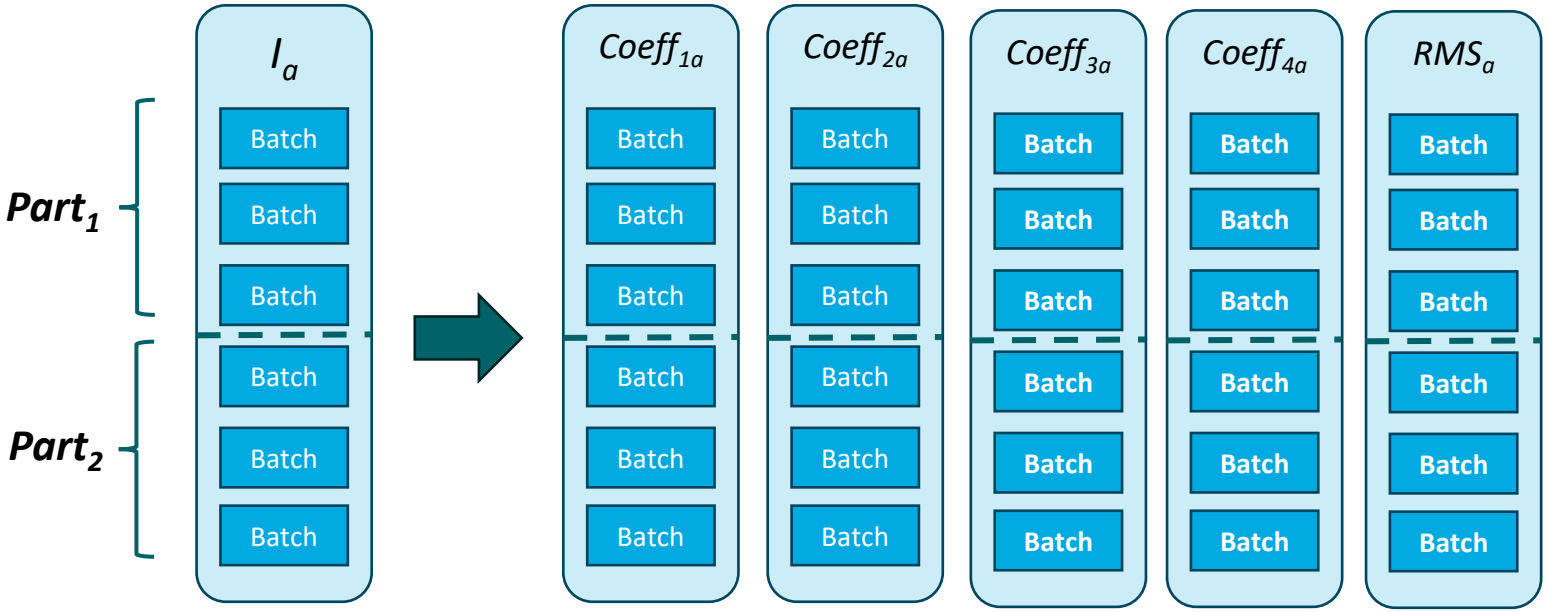


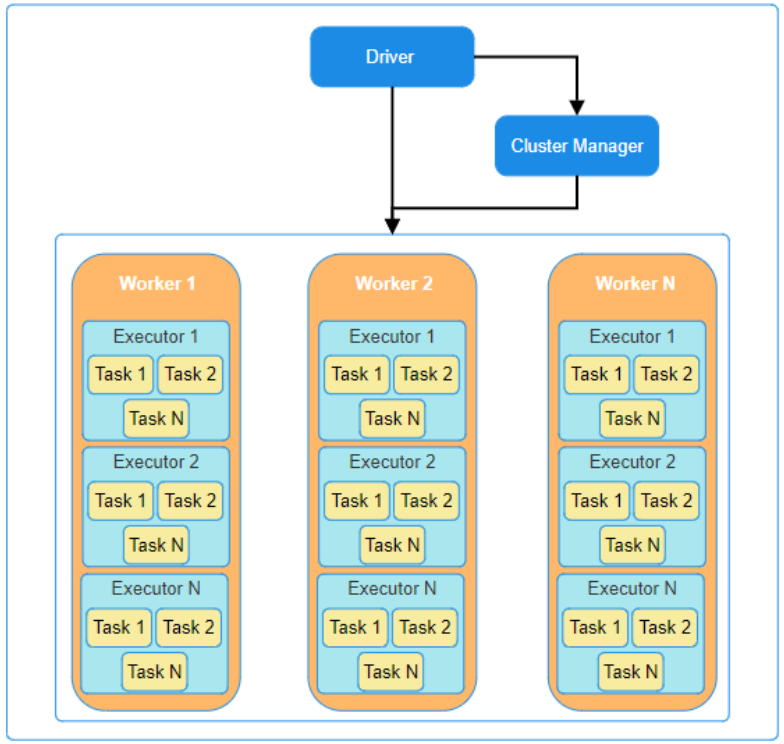
Online Processing

Offline Historic Data Parallel Processing

Why Spark

- When performing feature extraction on bulk CPOW data, we are likely dealing with terabyte scales of data
- Signal transformations make data grow significantly in memory
- Spark splits a large dataset into independent “partitions”, divides a compute cluster into “executors”, then allocates “tasks” to each executor such that tasks run in parallel
- *Analogous to Python’s ProcessPoolExecutor*

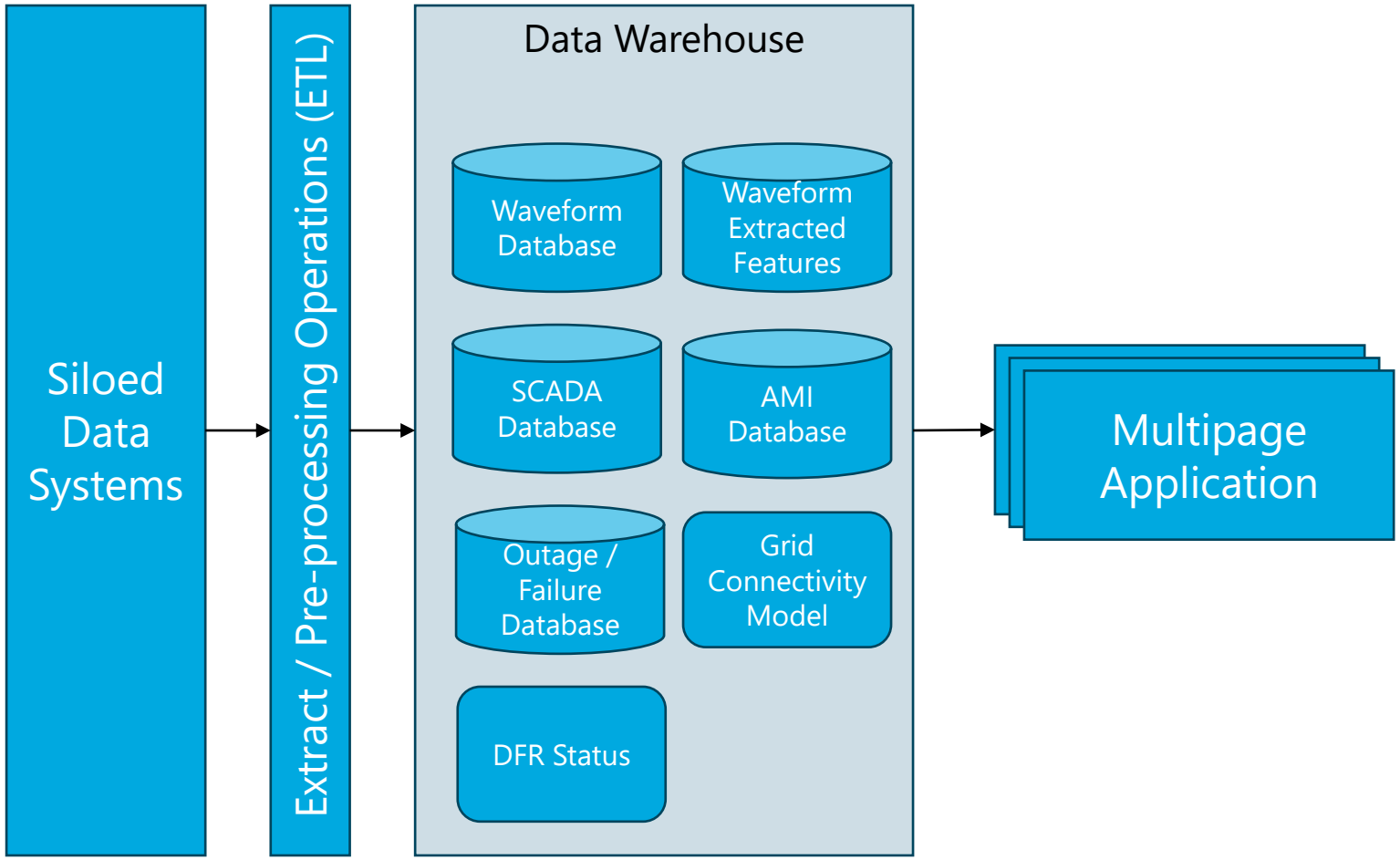




Above figure: Understanding Apache Spark Architecture | by Shobhit Tulshain | Medium
https://medium.com/@shobhittulshain/understanding-spark-architecture-6003184a12ec

(Partially Offline) Application Architecture

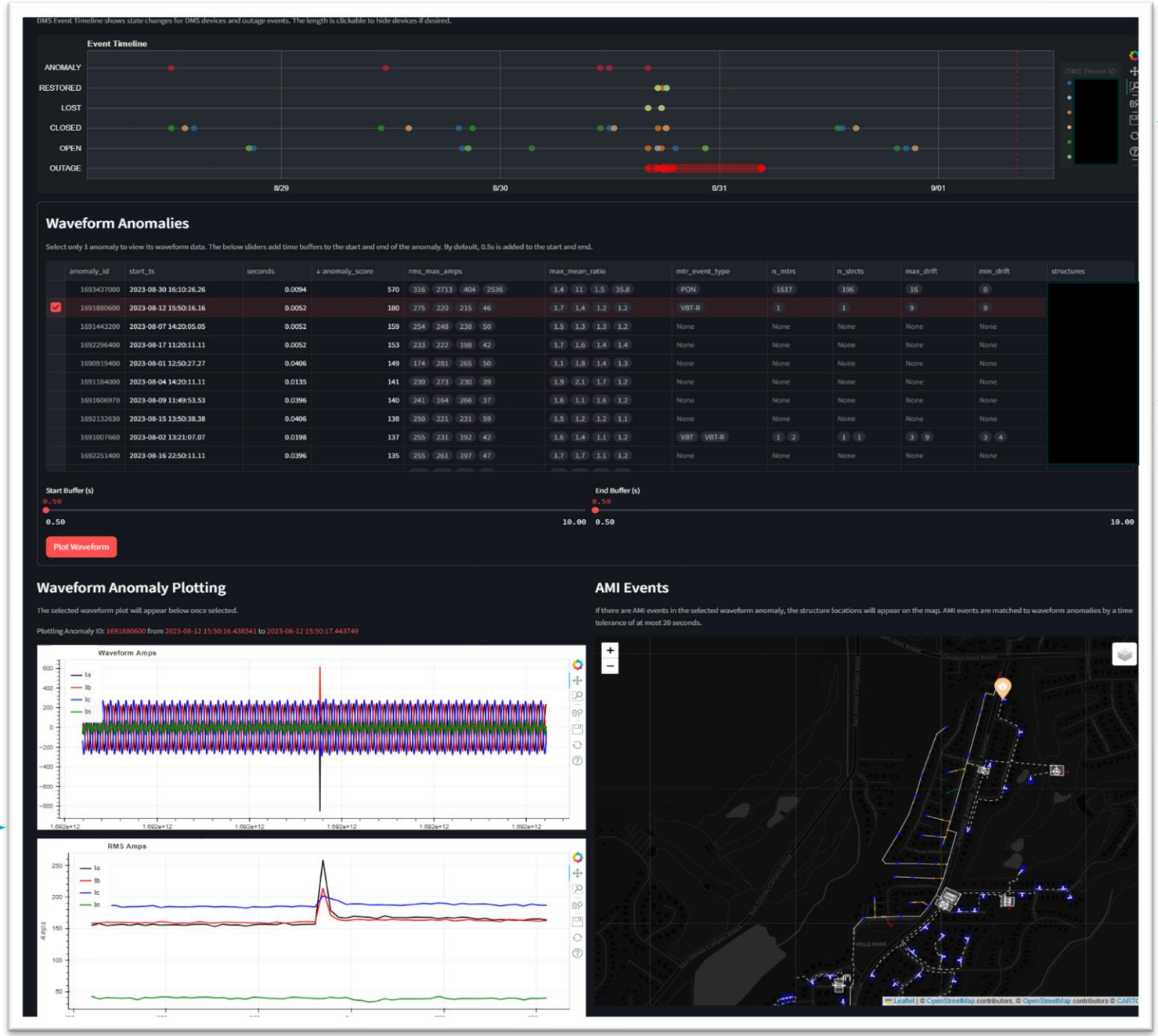
- Preprocess source data as much a possible to de-burden compute within the application
- Application built in Streamlit Python framework
- Impala-Python API interacts with database tables used in the application



Application UI

Supported roles :

- Incipient fault detection (Grid Ops)
- Post-event failure analysis
- Analytic dev / testing (Data Scientist)



Event timeline

Anomalous waveform table with related features

Waveform / RMS plotting

Event location plotting using AMI

Next Steps

- More comprehensive platform for incipient fault detection
  - Live connections to systems outside substation
- Common Substation Platform
  - SCE is designing a system to aggregate and manage data and devices within the substation, including fault records and sequence of event records (SERs) from substation relays
- Digital Substation
  - Sample Value data stream (IEC 61850) 4.8kHz / 15.36 kHz continuous
    - Requires ~ 5Mbps network
  - Analytics subscribe to SV data stream at substation
  - Combination of waveform analytics and protection