Data Management and AI Tools for the Utility Environment

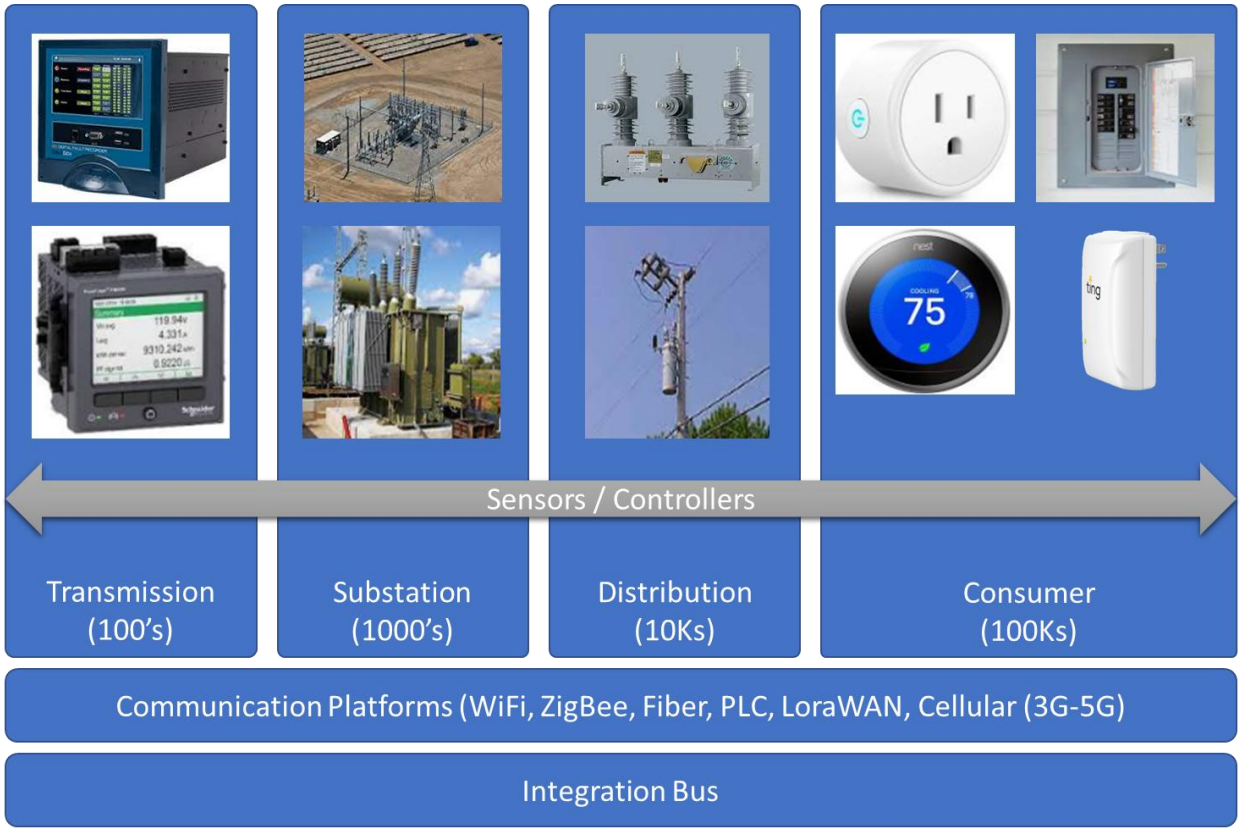
Prepared By: Theo Laughner

Introduction

The grid of today is far more complicated than the historical grid where there was unidirectional power flow. The drivers for these complications are multi-dimensional: technical, business, and policy based. The technical drivers include aging infrastructure, creating a digital twin of the infrastructure, diverse generation types, diverse generation locations, and electrification of new loads on the system. The business drivers include reducing unplanned maintenance costs, deferring infrastructure replacement, and increasing customer satisfaction. Meanwhile, there are political drivers that are forcing some of these changes such as safety, decarbonization and electrification.

Historically, large thermal generation plants made power, the generation was connected to the transmission system, which, in turn, was connected to distribution system via substations and finally connected to the consumer. The power moved in only one direction. Consider the figure below, generation is everywhere. Moreover, the system has storage that is sometimes a generator and other times a consumer. Finally, the system relies upon generation that is variable with weather conditions.

To better understand this complexity, the utility industry has deployed more sensors. These sensors serve to provide a digital twin of the utility system. In the figure below, consider the vast number of sensors available on the grid today. Traditional utility systems were able to manage data at the scale of hundreds or thousands of sources at a time. Now, there are millions of devices that can provide information about the power system in real time. Therefore new data processing techniques are required.



Analytics and Artificial Intelligence

Enter data analytics and artificial intelligence (AI) systems. These systems are designed to automatically analyze and act based on the inputs received from the sensors in the field. However, these systems are not without challenge to deploy. Analytics is the systematic computational analysis of data. Artificial Intelligence by contrast is teaching a computer system to perform tasks that normally require human intelligence. While there is an important difference, they share many common pipeline elements: capturing, accessing, acquiring, wrangling and cleaning, analyzing and modeling, and deploying results.

AI Failures

While AI is an emerging field with many successes, there have been some notable failures. These failures have served to create risk. Consequently, utilities have been slow to adopt the technology. These failures often stem from models being corrupted by interaction with data that is not well

vetted. For example, in less than 24 hours, Microsoft Tay, an AI chatbot, started making offensive and inflammatory tweets on its Twitter account. This was due to the underpinning model being inappropriately influenced by other users in Twitter. Obviously, great care should be taken to manage the training data used by AI systems.

AI Analytics Pipeline

For AI to work, a model must be built. For a model to be built, data must be collected. After the data is collected, it needs to be labeled so that a model can properly learn the condition described by the data. Once the model is trained, it can be deployed to analyze additional new data. Many of the publicly available models can quickly and accurately identify objects, people, or faces. These models have been built using large training data sets. The data sets are collected through the AI pipeline.

The AI pipeline includes the following activities: Capture, Access, Acquire, Wrangle & Clean, Analyze & Model, and Visual Report and Propagate. These elements are necessary to properly train an AI system. Once the system is trained, then the model can be deployed to a variety of places that can enable the system to act.

Capture

Capturing data has several hurdles. For example, the data is not always captured. This can be the result of a data collection device not being configured properly or activated. Special consideration should be given to how fast data is collected. For example, if the phenomenon being measured occurs at 100khz then a 10khz sampling rate will not be sufficient to capture or describe the phenomenon. Other considerations include timestamp accuracy, communications bandwidth, device lifespan. These all factor into the cost of the system.

Access

Accessing the data can be difficult in the utility environment. For example, devices are often connected to an operations network. However, the users of the system are often in the corporate network. These systems have limited connectivity often through a firewall that intentionally prevents traffic between the two networks. Consequently, access to device data can be very limited. It is often beneficial to set up an automated data collection system that connects to the device on the operations network and stores the data in the corporate network.

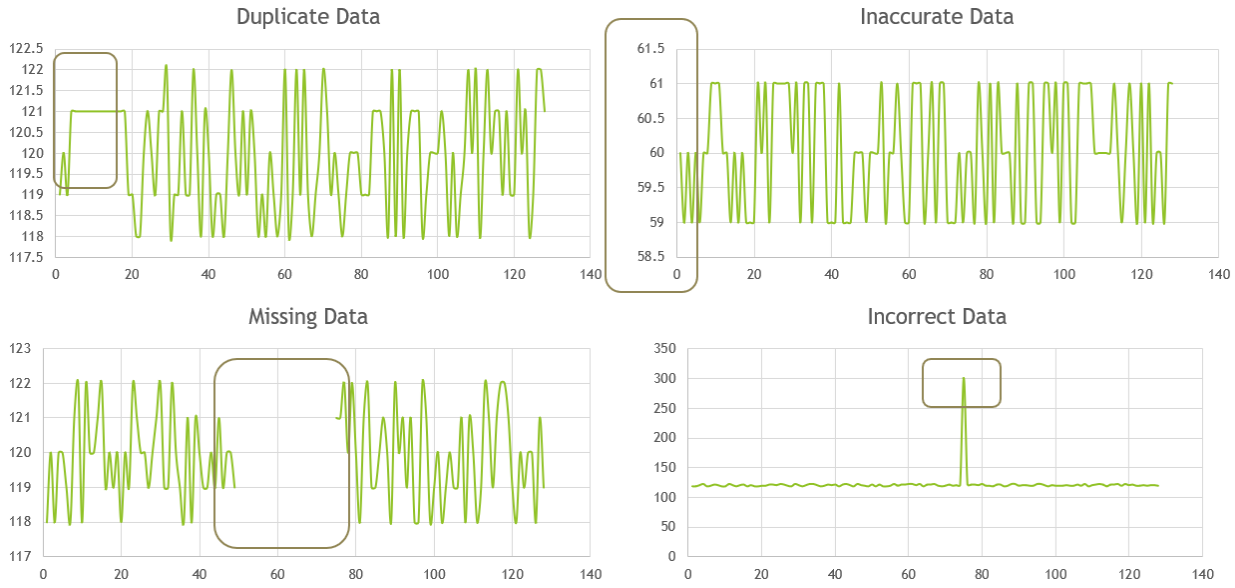
Acquire

Once the data has been captured and collected from the remote device, the data needs to be stored for processing. Consider a single digital fault recorder with 80 channels. For each event captured by the device, the data stored is about 3MB of data. If a utility deploys, 250 such devices and each device has 1 event per day, this amounts to nearly 750MB of data per day and over 20,000 channels of data to review. Similarly, a single PMU with 1 channel collects about 10.3 million points of data per day. This translates to over 60GB of data per year.

Collecting and storing the data can be complicated by the fact that each vendor uses a proprietary file format, proprietary protocols, and proprietary storage systems. This will make it difficult to perform machine learning or AI on later. Care should be given to consider, how long data will be kept, where the data will be stored and what format the data will be stored in.

Wrangle and Clean

If the data captured and stored by the system is of poor quality, then much like Microsoft Tay, the system will make bad decisions. Therefore, it is important to make sure that the data is of sufficient quality to train the system. There are several examples of bad data: duplicate data, inaccurate data, missing data, and incorrect data. These are shown in the figures below.



These errors are often introduced through poor configuration management. Examples include misconfigured thresholds and system configuration changes that aren't reflected in monitoring systems. These issues are sometimes exacerbated by devices that are difficult to configure.

Analyze and Model

Once all of the data is collected, the data needs to be tagged. The tagged data is used to help train the AI model. Once the model is trained, the system can be used to identify other similar issues. Important to remember is just how much data is required to train a model. Performing some of this analysis by hand can be labor intensive and take a lot of time especially if the analysis and modeling tools aren't scaled to deal with the quantity of data that is required.

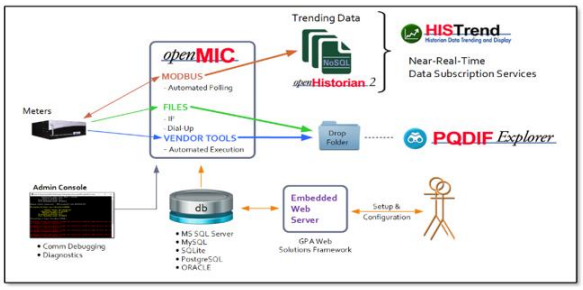
For example, if a system as 1 million data points and it takes 1 second per minute to do the analysis, then the analysis will take nearly 12 days to complete.

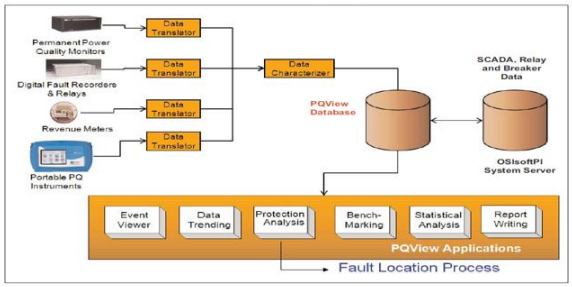
Visualize, Report, And Propagate

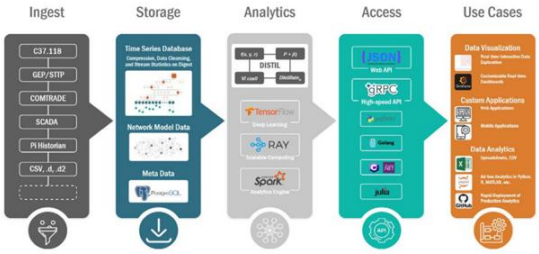
Once the model has been trained, it can be deployed to the detection system for use. The analytics pipeline is similar. However, the primary outputs of an analytics system is generally a dashboard, email, or report of some kind.

Automated Systems

There are several systems that can aid the AI pipeline. This report provides three examples – a commercial system that is closed source, an open-source software system, and a hybrid system that is commercial, but provides interfaces through open application programmatic interfaces APIs. They all follow the some basic model as described above – get the data, store the data, analyze the data, and provides reports/analytics/models. The architectures for each are shown in the figure below.







Deployment Methodology

These systems are heavily operations technology (OT) and information technology (IT) involved. As such, deploying them often requires a cross functional team of experts from the operations, OT, and IT organizations. A typical approach to deploying a system of this type follows the software development lifecycle (SDLC) process . The process includes the following phases: initiation, definition, design, development, implementation, operations, and testing.

Initiation

During the initiation phase, a project plan is developed. The project plan includes the scope, stakeholders, risks, assumptions, success criteria, deliverables, service level agreements, deployment plans and success measures. These may be summarized in a project charter.

The project timeline is a key component of the project plan. The timeline often takes longer than planned due to resource constraints that aren't always identified early in the project. So allow plenty of time.

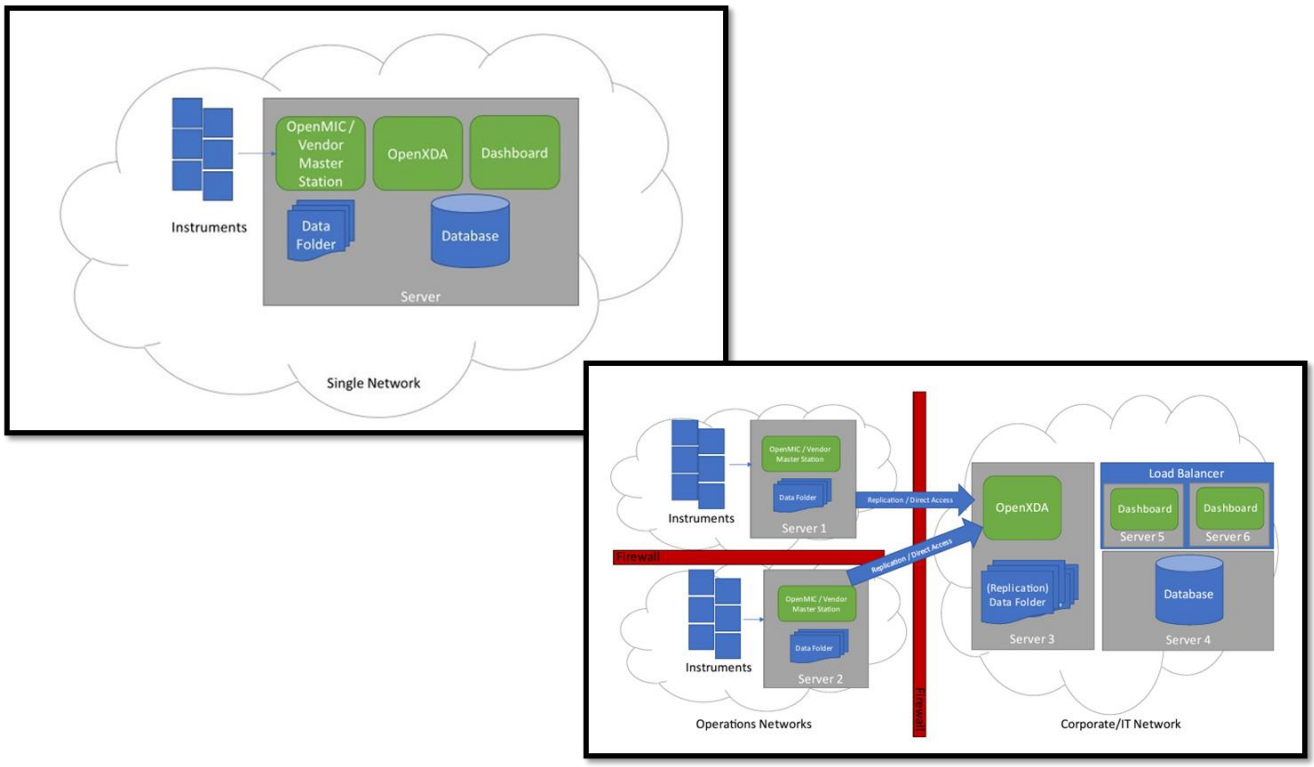
Definition

During the definition phase, the requirements are identified. The requirements should include business requirements, but also should include technical requirements such as speed and volume. An example table of requirements is shown in the figure below.

High-level Requirement ID	Business Topic	High-level Requirement Name	Priority	Long Description	Comments
1	Business Requirement				
1.1	Business Requirement	Device Variety		Demonstrate the ability to integrate data from a representative sample of field devices both at transmission and distribution levels.	Could include monitoring or operational
1.2	Business Requirement	Remote Monitoring T&D		Remotely monitor steady-state and event-driven T&D system events for trending of voltage regulation, harmonics, voltage sags, transients, and others.	Manual
1.3	Business Requirement	Automatic Monitoring T&D		Automatically monitor T&D system events for trending of voltage regulation, harmonics, voltage sags, transients, and others.	Automated
1.4	Business Requirement	Event Driven Monitoring T&D		Event-driven monitor T&D system events for trending of voltage regulation, harmonics, voltage sags, transients, and others.	

Design

Once the requirements and specifications are established, then an architecture can be selected. Two parameters that help inform architecture are scalability and redundancy. Example architectures are shown in the figure below.



Development

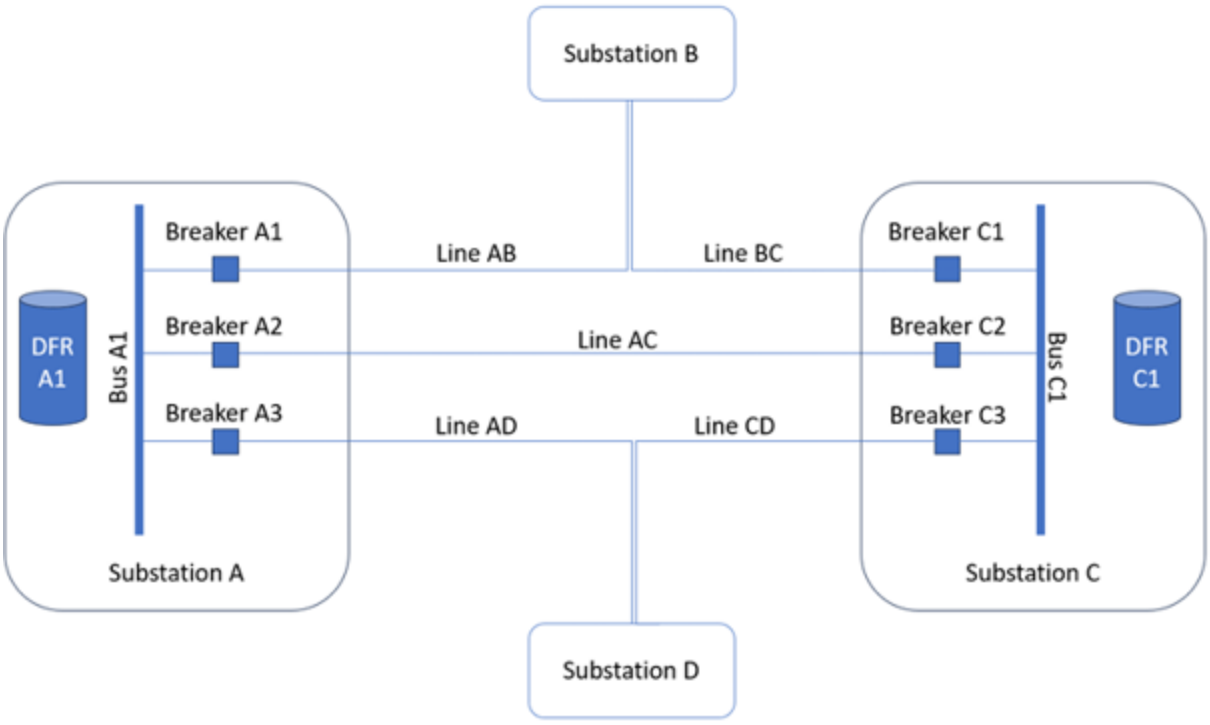
Once an architecture is selected, then the infrastructure (hardware) and software tools can be selected, procured, and installed. In some cases, a utility may choose to develop their own custom solution. In the case of a custom built solution, there are additional steps to that will likely occur. During the development phase, the hardware will usually be provisioned.

Implementation

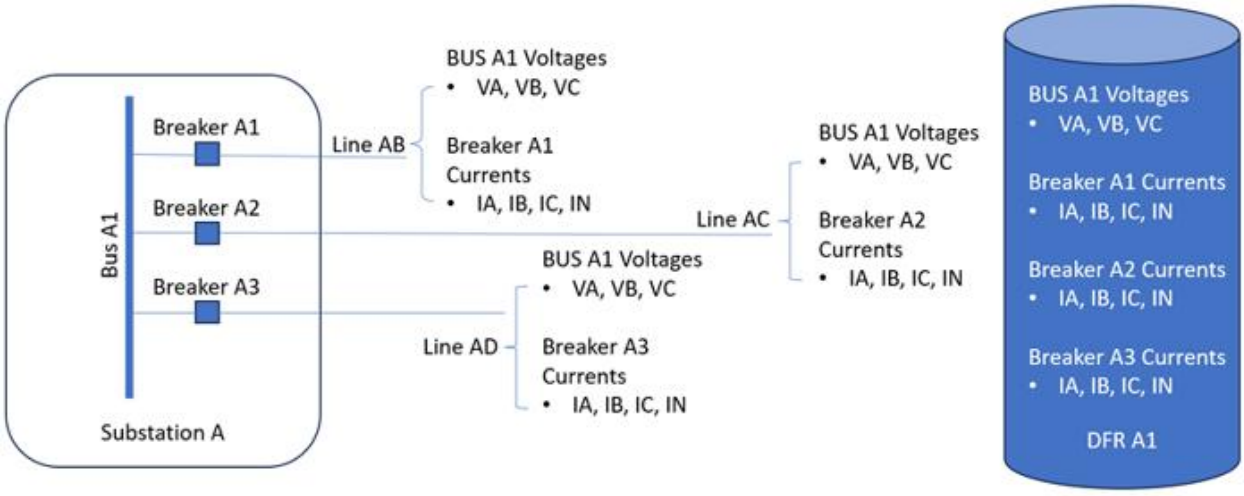
During the implementation phase, the instruments will be configured such that the tools can access and acquire the data from the instruments. The data storage mechanism will be ready for the data collected by the process. The tools used to manage, analyze, and train the models will be leveraged to being doing the analysis.

Operations

During the implementation phase, the instruments were configured. However, maintaining the configuration of the system components for ongoing use is critical to successful operation and training of the system. In the simple network below, there are two digital fault recorders (DFRs) at two of the four substations (Substation A and Substation C). Each element of the system that is monitored should have a unique name or identifier as can be seen in the figure below.



Now, consider the DFR at Substation A. The single DFR is connected to a bus, multiple lines, and multiple breakers. Each breaker likely has multiple currents. So even for this simple substation, there are numerous channels that need to configured and maintained as can be seen in the figure below.



Testing

After the system has been deployed and configured, the system should be tested. There are a number of different types of testing that can be performed: functional testing, integration testing,

user acceptance testing, performance testing, and regression testing. These each test different aspects of the system. For example, functional testing assesses whether the system does what it was expected to do. Whereas performance testing assesses whether the system meets the performance criteria. Finally, regression testing ensures the system still works after a change is made.

Summary

In summary, AI tools can be an effective tool for analyzing large quantities of data. The tools need to be trained using high quality data or have the potential to provide the wrong action. Deploying these tools often requires a cross-functional team of IT, OT, and Business Unit subject matter experts. Finally, having a good project plan to deploy the tools will help the project meet success criteria.