

How to Use Streaming Analytics to Create a Real-Time Digital Twin

Brad Klenz, SAS Institute Inc.

ABSTRACT

As the Internet of Things (IoT) expands, connected devices are frequently located in remote places, operating in different physical environments. These devices communicate with control systems and with each other.

It can be challenging to know and understand the environment in which these devices operate, and whether devices are operating properly and efficiently. To meet these challenges, you can create a “digital twin” of a device: a virtual representation of it in real time. A digital twin tells you how a device is operating, no matter where it is physically located.

IoT devices have a number of sensors installed on them, as well as sensors for the environment around them. Analytics can bring this sensor data together to create a true real-time digital twin. This paper explains how streaming analytics in SAS® Event Stream Processing can be used to create a real-time model of a remote device.

Analytics enable a digital twin to do the following:

- Fill in gaps where sensor data is not available.
- Notify when a device is not operating efficiently.
- Provide advance notice when a device is failing.
- Detect when devices are not interacting properly.
- Forecast future operating conditions.

INTRODUCTION

IoT is bringing the same type of understanding and control to the physical world that have become familiar in the digital world. In the digital world, financial transactions can be processed in real time to assess portfolio risk, or detect fraudulent transactions. Network traffic can be monitored for cyber security intrusions. Social media can be studied and influenced. But moving to the physical world surpasses another boundary.

With IoT, data is collected from sensors on a device, on neighboring devices, the environment around a device, and whatever interacts with the device. The speed is real-time, and connectivity allows us to span distances instantly in many cases. Advances in streaming analytics now enable us to process this real-time data using machine learning and artificial intelligence.

Connecting to the physical world introduces challenges, some obvious, some not so obvious. First, we must obtain data from the physical objects themselves. In many cases, sensors are incorporated into the device and take direct measurements of the factors that interest us, such as the oil temperature in a vehicle. We can also have sensors that measure the surrounding environment that has a physical effect on the device. Air temperatures and humidity affect how a combustion engine burns fuel. In addition, there are indirect measurements that easily capture a condition of interest. Motor bearings can be measured with vibration sensors and audio noise. These readings can indicate whether the bearing is not running smoothly and could be degrading.

In some cases, measurements cannot be taken without affecting the operation of the device. Internal to the engine, the air flow determines how efficiently the engine burns fuel. But you might not be able to put the sensor in a place that does not disturb the airflow. In some cases, the environment can damage the sensors that take the measurements.

The cost of sensors, especially in relation to the device, can also have an effect. A vibration sensor that would be inexpensive in the application of an aircraft engine might be too expensive for an automotive application.

HOW CAN ANALYTICS BE USED TO CREATE A DIGITAL TWIN?

Without analytics, a digital twin simply reports data that is collected remotely. If you want to know the temperature of a building in another city, you read a temperature sensor and report a value on a dashboard. You can also save measurements in a database and review previous values. You can summarize data and create reports with the average temperature and the observed range of temperature values. This information can provide a sense of the general status of the building. However, it leaves out information that could give you a better understanding of the specific operation of the building.

With analytics applied to the sensor data, you can obtain a more comprehensive view of device. Here are areas where analytics can help:

FILLING IN THE GAPS

Although the amount of sensor data can grow tremendously, not everything is measured all the time. There could be places where sensors are not present or not possible to use. There could be different time intervals than the sensor supports. Filling in these gaps with analytics is necessary to gain a full understanding of the device's operation.

ANOMALY DETECTION

A real-time dashboard of sensor data can report the current status of a device. A more interesting view would be to show when and where a device is not behaving normally and indicate what is different. What is defined as normal operation can depend on many factors. The device could have different configurations, it might operate differently due to expected changes in the surrounding environment, or it might have a known degradation pattern.

LOOKING FORWARD

A database of previous sensor measurements can be used to review all previous operating states of a device. In some cases, though, you might want to know the future operating state, or to determine the operating state under a set of conditions that have not been seen before. Using an analytic model of the device can estimate the results under these unknown conditions.

LEARNING FROM ALL YOUR DEVICES

It is common to have the same type of device installed in many places, and to have new devices installed over a long period of time. Each device might itself have unique characteristics relating to its usage and environment, but you can learn something from each device and apply that knowledge to other devices. When building analytic models of our devices, the model errors can tell us what characteristics are well understood and what characteristics need further research.

SOME USE CASES DISCUSSED IN THIS PAPER

In order to clarify the various topics that I cover in this paper, I will illustrate with examples from various projects I have worked on. Here is an overview of these use cases to get you familiar with them. Later, we will go into details to explore each topic.

SMART GRID

The power grid represents an industry that is adopting new methods of operation, which require new data sources to be successful. The existing power grid is a well-understood system with much historical data and analysis under many operating conditions. The addition of new distributed energy resources (DER) like renewable (solar, wind) generation and energy storage is changing how the power grid is operated. These new generation sources are less reliable than historical power sources. Solar and wind generation only produce power when solar and wind resources are available. A solar farm's output drops dramatically with each cloud that passes by. This is a localized condition that might apply to a remote location with only basic weather data available. These new power sources are being incorporated into a system that has a very high standard of reliability, the power grid on which we all depend. New sensors model the power grid in much greater detail, including more detail on load conditions.

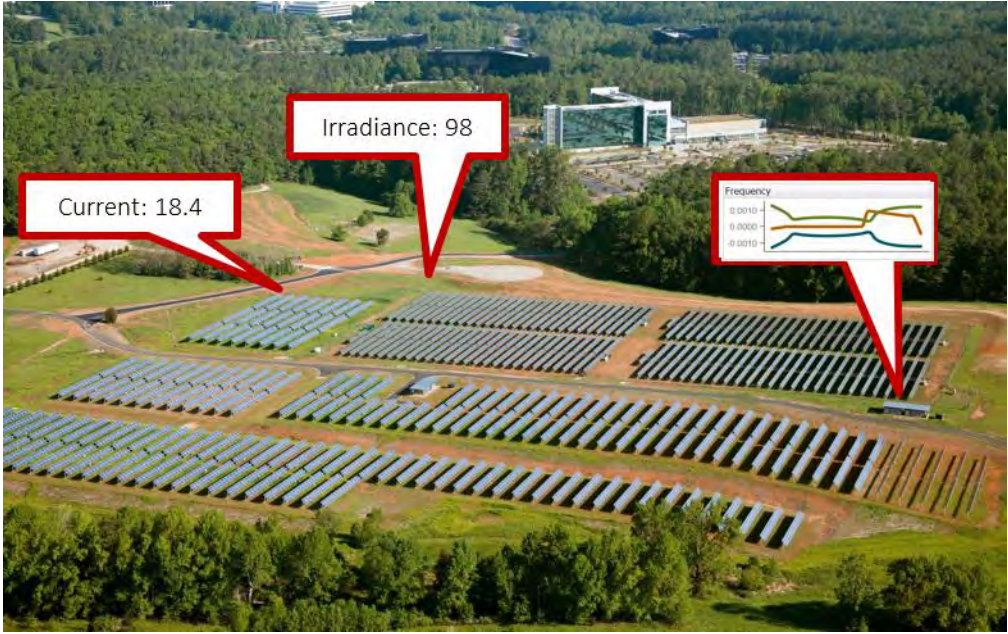


Figure 1. Solar Farm

CONNECTED VEHICLE

Modern cars and trucks have approximately 400 different sensors taking measurements. This includes mechanical measurements of the main systems (engine, transmission, brakes) like temperatures, pressures. It also includes emissions measurements, and operating conditions like number of passengers or environmental data. One of the most challenging aspects of connected vehicle data is the need to run the analytics on the vehicle. Although there is much data available, the cost to send the data from the vehicle is prohibitive.

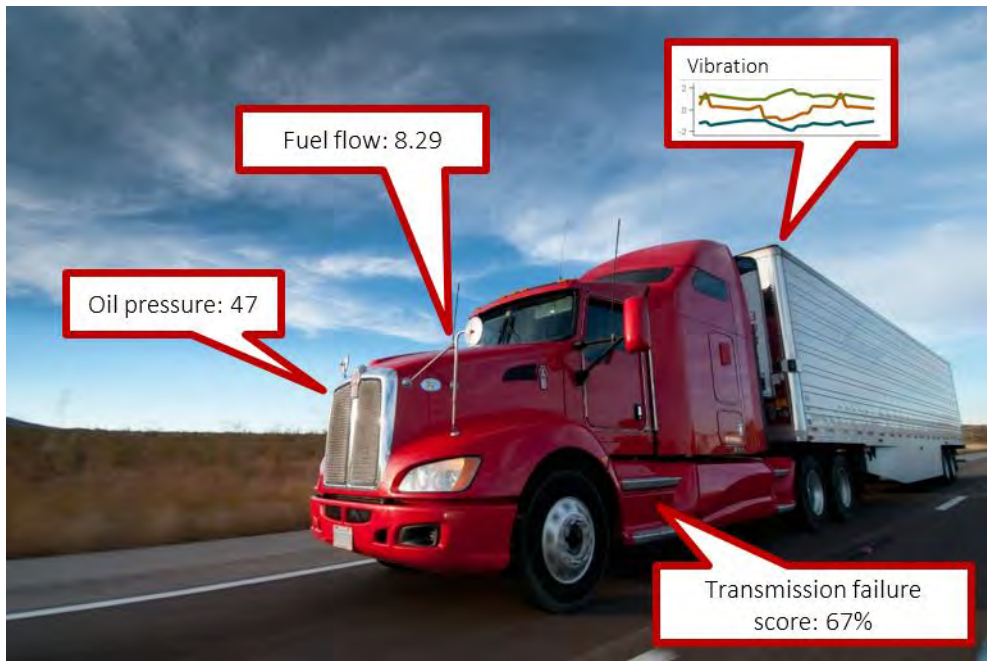


Figure 2. Connected Truck

SMART BUILDING

Buildings are large consumers of energy. Many sensors are already installed to support the building's automation systems. These sensors capture data on the state of systems such as HVAC, water, and lighting, as well as their control systems. Additional data can be added to capture the environment (weather), occupancy, and usage conditions. The objective is to operate the building at a comfortable level for the occupants in an energy-efficient manner. In addition, equipment in the building can be monitored for potential failure or performance degradation. Analytic models are used to determine when a part is likely to fail so that it can then be serviced ahead of time, avoiding an unplanned outage.



Figure 3. Smart Building Data Sensors

FILLING IN THE GAPS

Although it seems with IoT that sensors are being installed everywhere, there are actually many places where they are not available. Here are some examples where sensor measurements might be lacking:

A SENSOR MIGHT BE TOO EXPENSIVE TO INSTALL ON EACH DEVICE AT EACH POSITION

Just because there is a sensor available in the market to capture a desired measurement, it doesn't mean that it is a good business decision to install it. In the smart grid use case, one of the available sensors is called a phasor measurement unit (PMU). These sensors cost about \$2500 each, with additional cost for the installation and network communications. This cost is justifiable for installation at power generation facilities and larger substations. But until technology costs drop, PMUs are not justifiable at something like the household or building level.

Even though it is cost-prohibitive to install an advanced sensor on every device, the expense will be justifiable for some devices. Selectively installing advanced sensors serves a couple of purposes. First, you will start collecting detailed data at selected locations. This data is then used to model these selected devices, and you can apply that model to devices that lack the advanced sensors. Second, the detailed measurements from some selected locations can be used to build a more detailed model of the entire system. In the power grid use case, there are relationships with power generation and load across the grid. Having the advanced sensors in some locations captures important details on these interactions.

THE SENSOR MAY INTERFERE WITH THE FUNCTION OF THE DEVICE

Although a sensor might be available to capture a measurement, having the sensor installed can affect the operation of the device. One example is an airflow sensor. In a combustion turbine, airflow is critical to the efficient operation of the engine. Unfortunately, the presence of the sensor in the airflow will alter the function of the turbine. Other examples of sensor interference include cases where the weight of the sensor affects the operation or balance of the device.

THE COMMUNICATION CONNECTION WITH THE SENSOR MIGHT NOT BE POSSIBLE, RELIABLE, OR COST-EFFECTIVE

You might find situations where communication with the sensor presents challenges. With devices in remote locations, high-speed, reliable network bandwidth might not be available, or is cost-prohibitive. In addition, devices that are in motion can also have sporadic connections. In connected vehicle applications, there is a large amount of data on the vehicle being generated at high frequency, but the cellular connection costs to do the data analysis off the vehicle are not economical. In this case, we run the analytic model on the vehicle, with the full fidelity of the data available. We use the analytic model to determine the key characteristics of the vehicle's operation, and send the selected results over the cellular connection. A larger system model looking at vehicle traffic flows would then benefit from the detailed data captured on the vehicle, and have a cost-effective means to model the traffic flows.

DATA COLLECTED AT DIFFERENT TIME INTERVALS

Some sensors are designed to collect data at different time intervals. This might be a simple way to reduce the cost of the sensor. The measurement being taken might be known to slowly change values, and a longer interval reduces the network traffic requirements for the system. In a smart building application, some equipment, like chillers, is monitored closely for vibration at higher data rates. Other measurements, like building air temperatures, change more slowly. Another example of data that are sampled at different time intervals is when various sensor technologies have been installed over time. Older sensors can have a slower data rate than newer sensors.

ANALYTIC TECHNIQUES TO FILL IN THE GAPS

I find it helpful to think of a physical system as a fabric or surface over which a large number of measurements and relationships flow. This includes thinking of the physical system in spatially and temporally. As discussed above, there are many reasons why we have data measurements only at specific points on this surface, even though we know that the true physical system is continuous.

For the points in space and time where we have gaps, we create an analytic model to determine the relationships. These analytic models use predictive model algorithms like regression, neural networks, or gradient boosting.

In one of our smart building projects, we have over 100 separate analytic models to model the HVAC airflow within the building.

ANOMALY DETECTION

Anomaly detection is a pressing IoT topic. As we build the real-time system model for our digital twins, we start to see that most of the time we are collecting data, and that the model tells us the system is operating normally. This is important information, but it also is repetitive. We would prefer for our system model to tell us when the system is not operating normally. These are the more interesting situations. This knowledge will direct our attention to places where we need to improve the system. The concept of looking for times when the system is not operating normally is called anomaly detection.

As you start to look at the data, you will see that many different conditions could be called anomalous. One example is when your system model is estimating particular conditions in the system or system output, but you are not seeing those conditions in real time. Another example of an anomaly is where you have determined all the different operating modes that the system can be in, and now you find that it is in a mode that should not be possible. Yet another type of anomaly involves looking at the change of state

over time. You might see a pattern of states where each state is in normal conditions, but the pattern of the states over time is not normal. An abnormal pattern is another type of anomaly.

One important aspect of our definition of an anomaly is that an anomaly does not necessarily signal a problem. All we are saying is that the building is not operating as expected. The causes could be some state that you have not previously seen before, or it could be something, such as an equipment issue, where there is a problem that needs to be addressed.

As your system model improves over time, it reflects a better understanding of the system. This will result in flagging fewer and fewer situations as anomalies, as they are now explained phenomena.

USING YOUR SYSTEM MODEL FOR ANOMALY DETECTION

Let's first look at using your system model to detect anomalous behavior. With your system model, you can incorporate many different inputs from your system. This model is targeted toward higher-level outcomes. For example, in our smart building application we can build the system model to estimate energy consumed, or maintain a comfort level based on temperature, humidity, and air mixture. The system model would use all the different inputs we have available. In the smart building case, there are energy sensors on various pieces of equipment, sensors monitoring the air flows on the HVAC system, and sensors that incorporate environmental conditions, such as the weather, temperature, and humidity. There are also input variables to capture the building usage at a given moment.

Your system model uses those inputs to estimate the higher-level target variable, such as energy usage. With the system model now deployed on the building, you can monitor all those input variables and estimate the target variable. Then you compare the estimate to some actual measurement of that variable, such as the power meter coming into the building for energy usage. If the target variable is temperature from an inside air sensor, you compare the actual measurement from the sensor to the estimate from the system model. In many cases this estimate will be very close to the actual measurement of the target variable. What you are looking for is cases where the estimate is off. In these cases, the system is not reacting as it previously has to this set of conditions.

Note that this model will be a continuous model, so the estimate will be off by a factor and is not an exact model. But it does give us a set of confidence limits, and you will use these confidence limits to determine when you actually send a notification or alert, or flag an anomaly condition. We may choose the warning limits, based on our previous experience, around the level of 90% or 95% confidence. This will give us an initial set of notifications.

At this point you have notifications, but the next step will be to determine the root cause of the anomaly. Because you are monitoring all the input variables in real time, you can snapshot the state of the system when the anomaly occurred. You can even snapshot a time window before and after the anomaly to get more context. You can use the snapshots to then do root-cause analysis to identify what condition triggered the anomaly.

The algorithms to use for the system model are predictive model algorithms, such as neural networks. You should use these models due to the large number of input variables that you are modeling. These system models are also deep-learning models, which can handle very large numbers of interactions among the input variables.

Now that you have some very high-level, or macro-level, system models that are monitoring the system, you can further refine these models by adding more specific predictive models for individual systems within the overall system. In our smart building application, we have chosen to create predictive models for many of the elements in the HVAC system. These are elements such as the air handler units (AHU), the fan power boxes, and the variable air volume (VAV) boxes.

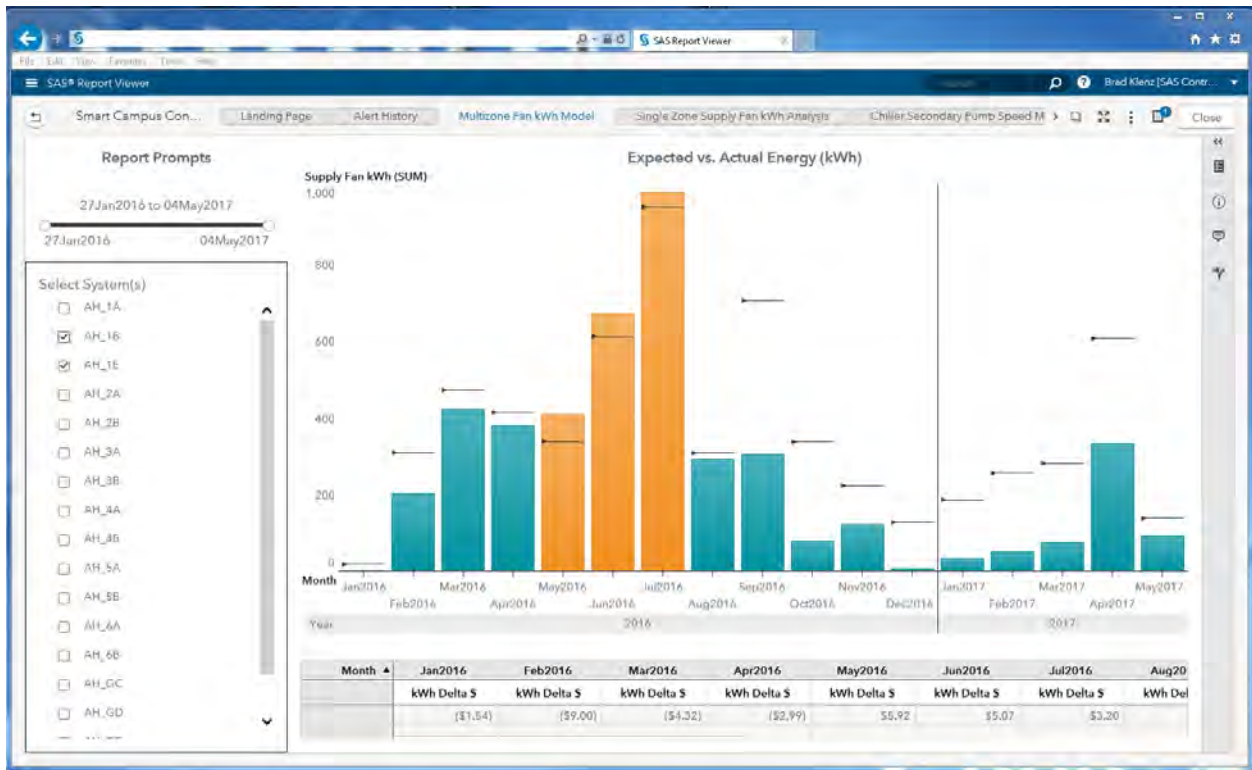


Figure 4. System Model Estimate versus Actual

By creating these additional device-level models, you get more fine-grained granularity for determining where an anomaly occurs. You also capture anomalies that might be lost in the larger variations of the overall system.

Another example of using the system model is on our power grid application, where we are looking for situations that threaten grid stability. In this scenario we have modeled various grid sensors like voltage and current, as well as the operating state of equipment, such as static VAR compensators. The system model then detects and alerts us when we are in a situation where we are vulnerable to a grid collapse.

OPERATING MODES FOR ANOMALY DETECTION

Now consider a different type of anomaly detection. In some cases, your historical data can determine how many different operating modes your system assumes. A good example is energy usage within our smart building application. To capture the complexity of energy usage, you can place sensors on the different subsystems of energy consumption in the electrical system in the building. You can also have sensors on the HVAC and subcomponents. You can have sensors on the lighting. You can have sensors on the office computer systems and on the non-critical plug loads, like the breakrooms and appliances.

To perform this type of anomaly detection, use an algorithm like Support Vector Data Description (SVDD). This algorithm uses a distance-based method similar to clustering. It works across many different dimensions simultaneously and determines a binary categorization. You train the model using data from known good operating modes. The model determines how much the different dimensions contribute to that mode. Once you have built a model, you can then use the model in real time to monitor the building. When monitoring the building, you take all the input variables and their current mode, and compare it to your model. The model will detect any combination of states that you have not seen before. This situation is then flagged as an anomaly.

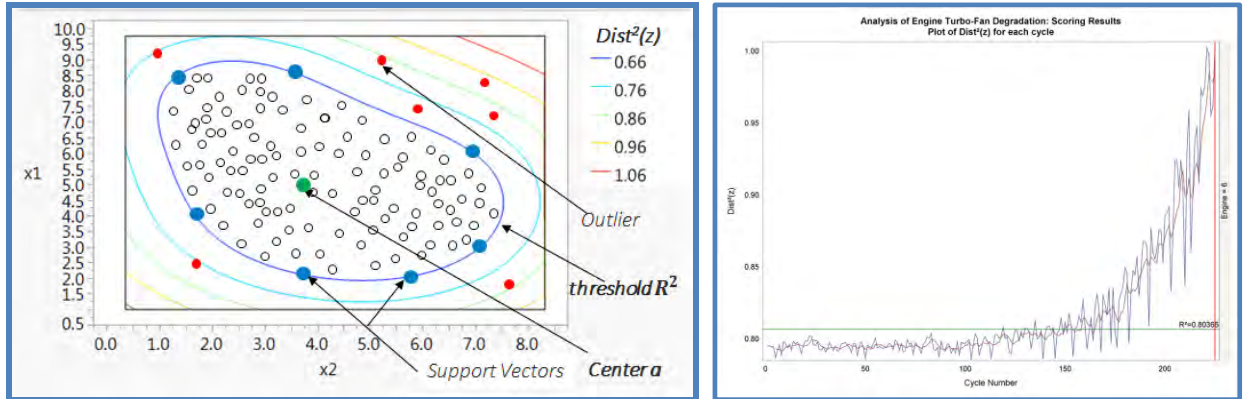


Figure 5. Support Vector Data Description Model and Degradation

In implementing our smart building energy model, we were able to see a relationship between the lighting in the building and the computer usage. This makes sense in that typically employees come into the building, turn on their lights, and then use their computer. The model detects these measurements as ones that are correlated. An example of an anomaly would be a case where a lot of the building lights are on, but there is very little computer usage. This indicates a mode not seen before. As discussed earlier, these anomalies might not necessarily be problems, but just unusual operating conditions. In the example where the lights are on, but computer usage is low, it is possible that building maintenance is occurring. The engineers doing the building maintenance have come in and turned on the lights to be able to do their work, but they are not using the computers within the offices. This would be an unusual condition, but it would not necessarily indicate a problem.

CHANGES IN RELATIONSHIPS BETWEEN PARTS OF YOUR SYSTEM

Your digital twin might have relationships between parts of your system that can help detect a different type of anomaly. One case is when you have similar machines that are operating in a similar environment, and it is expected that these machines would have similar performance in those environments. Examples would be a power plant where multiple generators are working in parallel, or multiple chillers in a building HVAC system. A second case is where your digital twin has a set of similar parts that function in parallel. An example would be individual cylinder measurements in a combustion engine. These represent more complex problems than a part that always operates in a controlled environment.

Because these parts are expected to perform in a similar fashion, an anomaly can be detected when one part has changed its characteristics in relation to the other parts that are operating in the similar environment.

For this type of anomaly detection, you use an algorithm called Moving Window Principal Components Analysis (MWPCA). This method starts with computing the principal components from the measurements of the related parts. It takes the correlated measurements and creates linearly uncorrelated values that are the principal components.

These principal components capture the majority of variability in the measurements. These are the environmental effects. In our power generation example, the variability is the fluctuating power load on the generators. For the building's HVAC chillers, it is the temperatures of the coolant.

Then we add a moving window to account for the time dimension of the problem. As the environment changes over time, the principal components for the monitored parts should change in a related manner. A change in the principal components would then indicate changes to the relationships among the parts, after accounting for the changes in the environment.

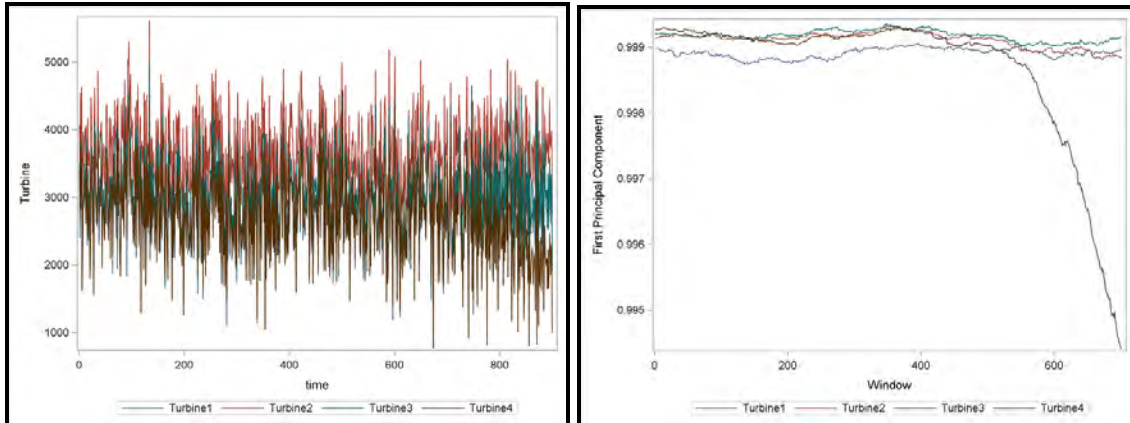


Figure 6. Measurements with Environmental Effects Compared with Principal Components

Another benefit of this algorithm is that you don't necessarily need measurements of the environmental conditions. If you know from the system design that the parts operate in a similar environment, the part measurements are sufficient to detect anomalies.

The algorithm also makes it easy to determine the part of the system that is causing the anomaly. The loading value for the principal components will point to the measurements for the bad part.

CHANGES IN PATTERNS OVER TIME

When collecting measurements for your digital twin, many measurements will be time series of individual values taken from a sensor. Examples mentioned earlier include duct air pressure in an HVAC system, or power consumed by an HVAC component like a chiller. Looking at these measurements over the time series, you will see regular patterns in the data when the system is operating normally. When an anomaly occurs, it might show itself as a different pattern in the time series that hasn't been seen before, or that is a rare occurrence.

We can detect this type of anomaly using an algorithm called motif analysis. In motif analysis the patterns are also called subsequences. The method analyzes the time series and starts to identify small repeating subsequences. These small subsequences might repeat over and over during normal operation, which can then identify larger subsequences.

An anomaly is then detected by ranking the subsequences that are identified. Normal operations are characterized by long subsequences, or a subsequence that is seen repeatedly with a short interval between occurrences. An anomaly subsequence is unique within the time series, or has a much longer interval between occurrences.

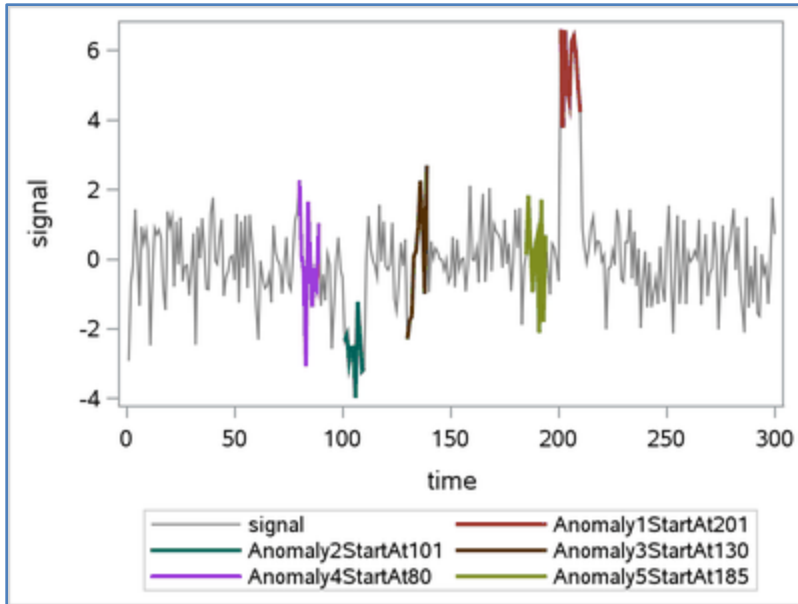


Figure 7. Time Series with Anomaly Subsequences Identified

Note that motif analysis is done on univariate time series, although the input time series could be the result of a multivariate model that captures a particular relationship that you want to analyze.

LOOKING FORWARD USING YOUR DIGITAL TWIN MODEL

To this point, the analytics model of your digital twin monitored your IoT devices in real time. You are able to fill in gaps where measurements were lacking. You were able to detect anomalies when the system was not behaving normally. Another important use for your analytics model is to look beyond the real-time, current state of your IoT devices. You can also use your analytic model to predict different states from the current state, to forecast a future state of a device, and to simulate the device in large numbers of possible conditions.

Using your analytic model to predict different states for your device, you now can perform "what-if" analysis. In this case, you feed your model different inputs for the state that you are interested in. The model then gives you the output state based on those inputs. This enables you to check different conditions that you might have, or that you think you might encounter. This enables you to learn more about your device using the digital twin. Note that these states are not just going back in history and finding a comparable time. You can look for new combinations of state that you have not seen directly before.

Another example of using an analytics model to look beyond real-time monitoring is to use your model to forecast future states. In this case, you can obtain estimates of various input variables that are anticipated for your model in the future. For example, in the smart building application, you can look at the expected usage and occupancy of the building at a future date. You can also get the weather forecast for that future date. Then, using the system model, you can forecast output results, such as energy usage or comfort level.

USING YOUR DIGITAL TWIN MODEL FOR SIMULATIONS

The third example of using your analytic model for non-real-time monitoring is to perform simulations of the device. Device simulations let you examine a range of possible input values and understand the resulting outputs. The simulations are useful for understanding the range of inputs that can be seen. Simulations also help you to understand various congestion points, or stress points, on the system. The simulations are most helpful in cases where you have built subsystem-level models.

The simulations enable you to understand the results at various points in your overall system using the subsystem-level models. By running the simulation over a wide range of possible input states, you can see the output state for many conditions that have not previously been observed. Because your model captures interaction effects between the states that you can see and combinations where the interactions cause undesirable results, running the simulation lets you identify undesirable situations and explore them further. Examining these undesirable situations, you can review your historical data and see whether they have been encountered in the past. You can gain a level of insight into how likely it is that you will see this undesirable situation. The simulation also gives you information to identify shortcomings in your system. Knowing those shortcomings, you can improve the system specifically to remediate known problem areas.

BUILDING YOUR DIGITAL TWIN MODEL

Now that you understand what your digital twin model is and how you use it, let's look how to build it. To build your digital twin model, you need to add to your knowledge in two areas. First, you need to understand the Analytics Lifecycle, historical data, and streaming data. Second, you need to understand the physical and software environment that needs to be constructed for the digital twin model.

One tip for building the system is to construct a learning loop. So the primary objective at first is to get an end-to-end system working, and then incrementally add to this system.

The Analytics Lifecycle provides the framework to create your digital twin models, improve these models, and manage the models.

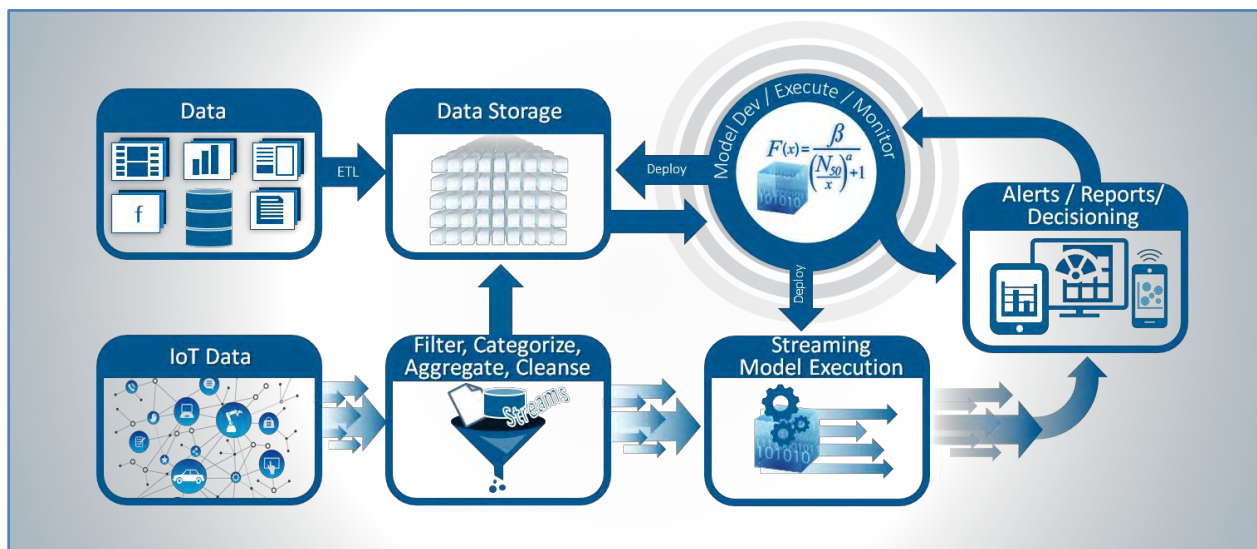


Figure 8. IoT Analytics Lifecycle

The Analytics Lifecycle will take your streaming data that is coming from your sensors, manage this data in real time, and provide the data for your historical context as needed. Even without real-time streaming analytics, many companies start by taking sensor data and collecting it in a large data database. You can use this database to build initial analytic models that describe your digital twin. You can take recurring data in a batch mode and process it to achieve some of your objectives for the digital twin. You can find normal states, summarize observed states, and revise the analytic model as you collect more data to improve the model. If you observe the model over time, you will generate ideas on how to improve your physical system.

To realize the full benefits of your analytic model for your digital twin, you do need to incorporate the model into a real-time streaming environment. The real-time streaming environment will provide the full-time, visible, digital equivalent of your physical systems. The streaming environment must be able to capture the sensor data that you need, and also implement the numerous analytic models that you have for your digital twin. One benefit of your digital twin is to provide real-time status of your remote devices;

therefore, the streaming environment must also be able to integrate with control environments and real-time dashboard systems.

There is, of course, a physical and software environment required to support your digital twin in the physical environment. This environment is called an edge-to-enterprise system. In your physical environment you must have sensors on the remote devices. The sensors will then feed into edge computing gateways to provide analytics right at the source of the sensor data. From your edge computing and remote devices, you then use network communications back to a central on-premises server or cloud environment. Having a cloud environment enables you to create analytic models that include many devices. The following diagram shows how your IoT system can comprise multiple levels.

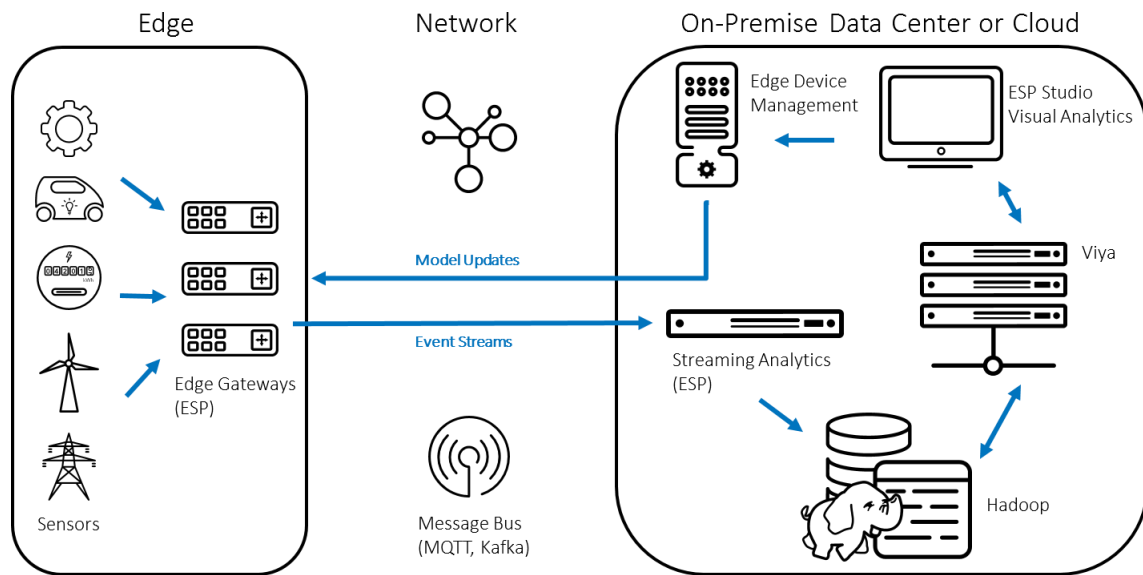


Figure 9. Edge-to-Enterprise Environment

CONCLUSION

IoT data coming from physical devices provides the foundation for a digital twin of your remote systems. The data itself provides basic reporting of the current operating state of the device. Adding analytics to this basic digital twin greatly enhances the benefit of capturing this real-time streaming sensor data.

Analytics enable you to fill in gaps that occur due to the limitations of your sensors. Usually, you would like your digital twin to give you information about the overall performance or health of the remote device. Individual sensor measurements can provide some key portions of this overall picture, but analytics will capture the relationships and interactions that are significant as well.

Analytics also alert you when the remote device is not operating normally, and enable you to troubleshoot these anomalies. There are many different ways that a remote device can enter an undesired condition. The relationship between parts of the device might be incorrect. You might see an operating mode that is unusual, or changes in your remote device can occur over time that can manifest themselves as a different pattern than the normal sequence of events. Various analytic algorithms in your digital twin will detect these anomaly conditions.

Analytics also enable your digital twin to provide information about your remote device for time frames other than the current time. Your digital twin can be used to forecast future performance of the remote device. You can do “what-if” analysis to study your remote device under different conditions, and you can simulate a range of possible conditions to find troublesome scenarios that might not have occurred yet.

A digital twin with analytics can be created by using the IoT Analytics Lifecycle to build the various analytic models that are needed. Once the analytics models are built, they can be deployed into the digital environment to reap the benefits of the digital twin. You will now be realizing the return on your investment in IoT.

REFERENCES

1. Kakde, D., Chaudhuri, A., Kong, S., Jahja, M., Jiang, H., and Silva, J. (2017). "Peak Criterion for Choosing Gaussian Kernel Bandwidth in Support Vector Data Description." In 2017 IEEE International Conference on Prognostics and Health Management (ICPHM) (PHM2017). Piscataway, NJ: Institute of Electrical and Electronics Engineers.
2. Kakde, D., Chaudhuri, A., and Peredriy, S. (2017).. "A non-parametric control chart for high frequency multivariate data." In Proceedings of the 2018 Annual Reliability and Maintainability Symposium (RAMS). Piscataway, NJ: Institute of Electrical and Electronics Engineers.
3. Chaudhuri, A., Kakde, D., Jahja, M., Xiao, W., Jiang, H., Kong, S., and Peredriy, S. (2018). "Sampling Method for Fast Training of Support Vector Data Description." In Proceedings of the 2018 Annual Reliability and Maintainability Symposium (RAMS). Piscataway, NJ: Institute of Electrical and Electronics Engineers.
4. Chaudhuri, A., Kakde, D., Sadek, C., Gonzalez, L., and Kong, S. (2017). "The Mean and Median Criterion for Automatic Kernel Bandwidth Selection for Support Vector Data Description." In 2017 IEEE 17th International Conference on Data Mining Workshops (ICDMW). Piscataway, NJ: Institute of Electrical and Electronics Engineers.
5. S. Peredriy, D. Kakde, and A. Chaudhuri. Kernel bandwidth selection for SVDD: The sampling peak criterion method for large data. In 2017 IEEE International Conference on Big Data (Big Data). Piscataway, NJ: Institute of Electrical and Electronics Engineers.
6. Gillespie, Ryan and Saurabh Gupta. 2017. "Real-time Analytics at the Edge: Identifying Abnormal Equipment Behavior and Filtering Data near the Edge for Internet of Things Applications" *Proceedings of the SAS Global Forum 2017 Conference*. Cary, NC: SAS Institute Inc. Available: <http://support.sas.com/resources/papers/proceedings17/SAS0645-2017.pdf>
7. Chou, Timothy. 2016. *Precision: Principles, Practices and Solutions for the Internet of Things*. Cloudbook, Inc.
8. Sico, Kathleen. 2014. "PMU Data Analysis and Short-term Voltage Stability Assessment." North Carolina State University, Electrical Engineering. Available: <http://www.lib.ncsu.edu/resolver/1840.16/9899>
9. Industrial Internet Consortium. 2017. "Industrial Internet of Things Analytics Framework." Available: <https://www.iiconsortium.org/industrial-analytics.htm>
10. Industrial Internet Consortium. 2017. "Industrial Internet Reference Architecture v 1.8" Available: <https://www.iiconsortium.org/IIRA.htm>
11. "Cisco and SAS Edge-to-Enterprise IoT Analytics Platform". 2017. Available: https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/Cisco_SAS_Edge_to_Enterprise_IoT_Analytics_Platform.html
12. SAS Institute Inc. 2017. "Time Series Motif Discovery Package." In *SAS® 9.4 and SAS® Viya® 3.3 Programming Documentation*. Cary, NC: SAS Institute Inc. Available: http://go.documentation.sas.com/?cdclid=pgmsascdc&cdcVersion=9.4_3.3&docsetId=castsp&docsetTarget=castsp_mtf_toc.htm&locale=en
13. SAS Institute Inc. 2017. "The SVDD Procedure." In *SAS® Viya® 3.2 Programming Data Mining and Machine Learning Procedures*. Cary, NC: SAS Institute Inc. Available: http://go.documentation.sas.com/?cdclid=vdmmlcdc&cdcVersion=8.11&docsetId=casml&docsetTarget=viyaml_svdd_toc.htm&locale=en
14. SAS Institute Inc. 2017. "The MWPCA Procedure." In *SAS® Viya® 3.2 Programming Data Mining and Machine Learning Procedures*. Cary, NC: SAS Institute Inc. Available: http://go.documentation.sas.com/?cdclid=vdmmlcdc&cdcVersion=8.11&docsetId=casml&docsetTarget=viyaml_mw pca_overview.htm&locale=en

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Brad Klenz
SAS Institute, Inc.
brad.klenz@sas.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.