WHITE PAPER

Using MATLAB with Big Data from Sensors and IoT Devices

MathWorks®

Introduction

Using smart sensors and other Internet of Things (IoT) devices, business and engineering teams can collect vast amounts of data from scientific instruments, manufacturing systems, connected cars, and aircraft.

With the right tools and techniques, you can use this data to make rapid discoveries and build more intelligence into your products, services, and manufacturing processes. But gaining access to and working with the data may sound like a daunting task.

Not for ***MATLAB***® users.

They can now accomplish what used to require computer scientists experienced with distributed processing systems and data scientists skilled at machine learning. This paper will explore the MATLAB workflow for working with big data from sensors and IoT devices:

1. Access historical data in files, databases, or the Hadoop Distributed File System (HDFS)
2. Visualize, process, and analyze this data to understand trends
3. Use machine learning to create models and algorithms for use in embedded systems, business applications, and other services

Let's look at an example of how engineers are using MATLAB with big data to manage their heavy equipment.

Case Study: Predictive Maintenance at Baker Hughes

Engineers at ***Baker Hughes***, a service provider for oil and gas operators, needed to develop a predictive maintenance system to reduce pump equipment costs and downtime on their oil and gas extraction trucks. If a truck at an active site has a pump failure, Baker Hughes must immediately replace the truck to ensure continuous operation. Sending spare trucks to each site costs the company tens of millions of dollars in revenue that could be generated elsewhere if the trucks were in active use at other sites. The inability to accurately predict when valves and pumps will require maintenance underpins other costs. Too-frequent maintenance wastes effort and results in parts being replaced when they are still usable, while too-infrequent maintenance risks damaging pumps beyond repair.

Baker Hughes collected terabytes of data from the oil and gas extraction trucks, and used this data to develop an application that predicts when equipment will need maintenance or replacement. MATLAB provided Baker Hughes engineers with the functionality they needed to develop predictive models and to combine multiple kinds of data, including sensor data from a proprietary file format, into one analysis application.

Baker Hughes projects savings of $10 million using this predictive maintenance application. They also reduced development time ten-fold.

"MATLAB gave us the ability to convert previously unreadable data into a usable format; automate filtering, spectral analysis, and transform steps for multiple trucks and regions; and ultimately, apply machine learning techniques in real time to predict the ideal time to perform maintenance."

— Gulshan Singh, Baker Hughes

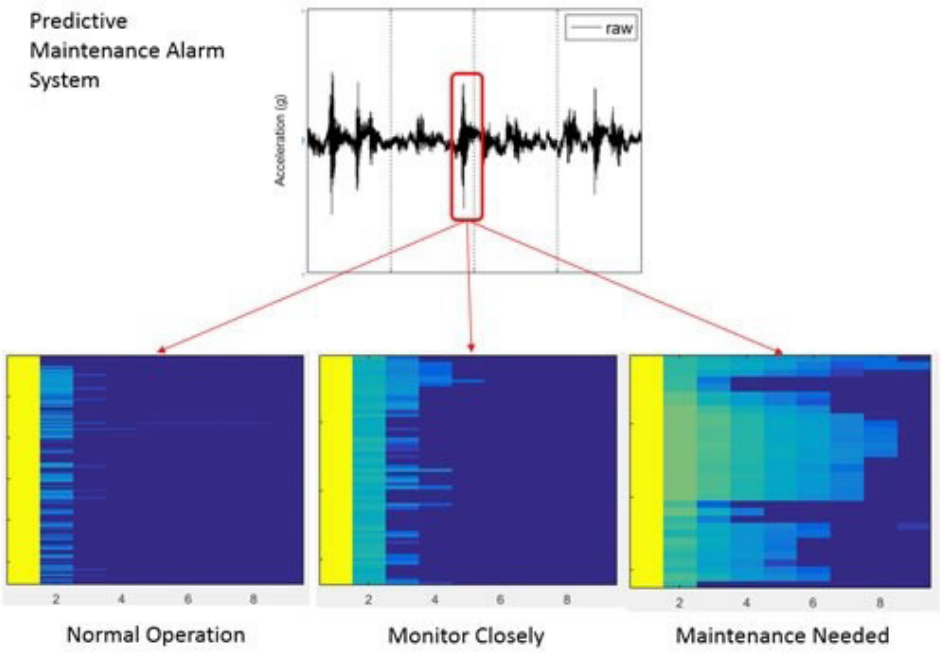


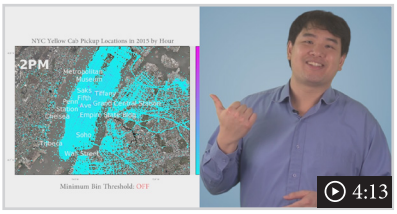
Figure 1. Baker Hughes' predictive maintenance alarm system, based on MATLAB.

Accessing Large Sets of Data

The first challenge in working with big data is determining how to access large data sets, which typically come in many different forms and are stored in various types of systems, such as files, databases, and Hadoop systems.

Files

Many big engineering and scientific data sets consist of a large number of small or medium-sized files. As the collective data and file sizes increase, data won't fit into the memory of a single computer. These files typically reside within one or more directories on a shared drive and may consist of delimited text, spreadsheets, images, videos, and various proprietary formats. MATLAB supports ***a broad range of file types***. MATLAB also supports tall arrays for processing data too large to fit in memory.



» *See MATLAB Tall Arrays in Action* – includes downloadable MATLAB code and data set

Databases

MATLAB supports many database types used to store and manage big sets of data:

- **Relational (SQL):** Widely used for business applications, popular among IT developers
- **Data Warehouse:** Based upon relational (SQL) databases, houses business-critical data, and provides analytical capabilities and fast access for business-critical applications
- **NoSQL:** Optimized for data that doesn't fit into relational databases
- **Data Historians:** Optimized for time-based, production, and process data commonly collected from industrial equipment
- **IoT Data Aggregators:** Typically includes cloud-based services for aggregating time series data from connected sensors and devices; typically accessed via web service calls

Hadoop

MATLAB supports Hadoop, a system for storing and processing big data sets based upon distributed computing and storage principles. Hadoop comprises two major subsystems that coexist on a cluster of compute servers:

- **HDFS, or Hadoop Distributed File System:** A large, failure-resistant file system
- **YARN, or Yet Another Resource Negotiator:** Manages applications that run on Hadoop, including batch processing frameworks, such as MapReduce and Spark, and SQL interfaces, such as Hive and Impala

Sensors and IoT Devices

With IoT big data, engineers and scientists need to be able to work with variety of storage systems and data formats used to store and manage data. For example, sensor or image data stored in files on a shared drive may need to be combined with metadata stored in a database. MATLAB lets you assemble and work with larger-than-memory data sets and to run on multiple computers without special programming. As we saw with Baker Hughes, data of many different formats must be used together to understand the behavior of the system and develop a predictive model.

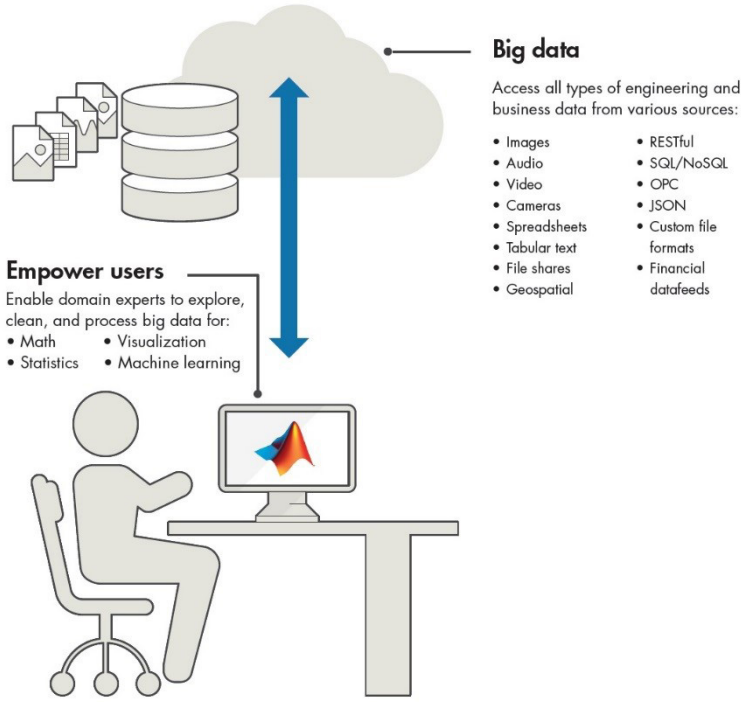


Figure 2. MATLAB users access and analyze a wide range of big data.

Once you have access to the big data collections, you can analyze and process the data, and generate models for use in applications.

Learn how to access and work with:

- *Images*
- *Spreadsheets*
- *Tabular Data Files*
- *Custom Files*
- *Databases*
- *Hadoop/HDFS*

Analyzing, Processing, and Creating Models

You can generate models from the enormous sets of data collected from smart sensors embedded in measurement instruments, manufacturing equipment, medical devices, connected cars, aircraft, and power generation equipment. These models can predict outcomes to make decisions on when to perform maintenance. The goal is to reduce cost on unnecessary maintenance or unplanned downtime. You can use these models to forecast, for instance, when to optimally turn on expensive power generation plants. Models can also be embedded into medical devices or vehicles to increase their efficacy and performance.

Such predictive models can be a game changer. But the data collected from these systems and devices is far from perfect. MATLAB helps you to identify trends in your data, clean and correct dirty data, and apply algorithms to determine the most influential signals in large data sets, so you can create and implement a practical and effective model.

Case Study: BuildingIQ Optimizes Building Energy Use

A valuable application of big data through predictive analytics lies in controlling energy costs by optimizing consumption based on weather sensors and cost data. ***BuildingIQ*** used MATLAB to develop a real-time system to minimize HVAC energy costs in large-scale commercial buildings with proactive, predictive optimization. The application analyzes billions of data points from power meters, thermometers, and HVAC pressure sensors, as well as weather and energy cost data. The team removed noise produced by sensor failures and created a mathematical model of the building's thermal and power dynamics. Algorithms use this calculated model to optimize occupant comfort while minimizing energy cost. The system is reducing energy consumption by 15-25%.

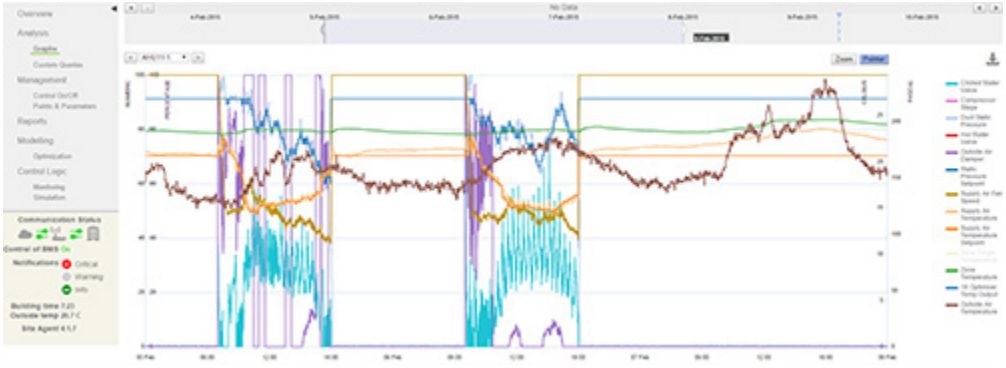


Figure 3. A plot from BuildingIQ's predictive energy optimization platform. The platform optimizes energy consumption by monitoring and controlling several variables.

> *"Developing algorithms in MATLAB is 10 times faster and more robust than developing in Java. We need to filter our data, look at poles and zeroes, run nonlinear optimizations, and perform numerous other tasks. In MATLAB, those capabilities are all integrated, robust, and commercially validated."*
>
> **— Borislav Savkovic, BuildingIQ**

Exploring and Processing Large Sets of Data

Before creating a model or theory from your data, it's important to understand what is in your data, because it may have a major impact on your final result. Steps include:

- Consider slow moving trends or infrequent events spread across your data
- Clean bad or missing data
- Derive additional information for use in later analysis and model creation
- Find the data that is most relevant for your theory or model

Let's look at some of the capabilities that can help you easily explore and understand data.

Visualization

Summary visualizations, such as the MATLAB **`binScatterPlot`** shown below, provide a way to easily view patterns within large data sets. The plot highlights areas of greater concentrations of data points, with changes in color intensity. Using a slider control to adjust color intensity lets you interactively explore data to rapidly gain insights.

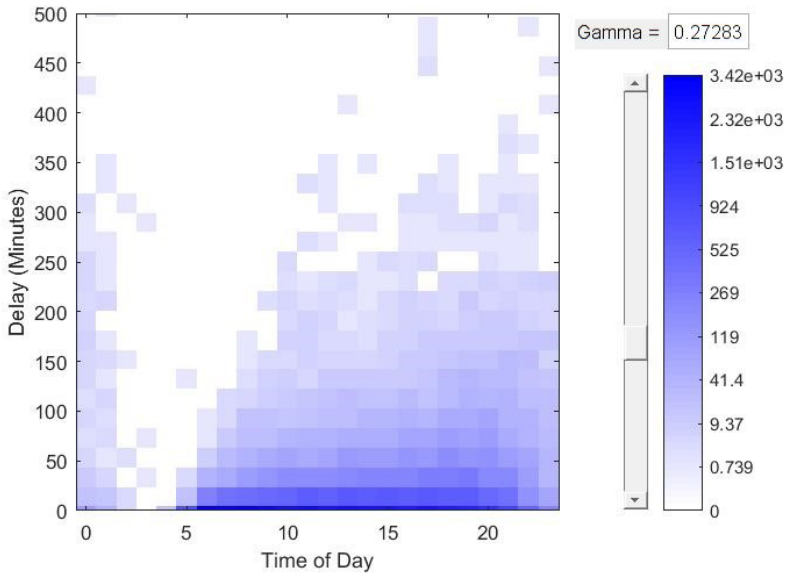


Figure 4. binScatterPlot in MATLAB.

» Get code example: ***Working with Big Data in MATLAB Using Tall Arrays***

Data Cleansing

All data contains outliers, or bad and missing entries. Before you can properly understand or interpret data, you need to remove or replace these entries. By programmatically cleaning this data, you can scale to manage new data as it's collected and stored.

```
                                                    % Filter out trips with…
idx = tt.TripDurationMinutes <= .5 | ...            % …really short times
          tt.TripDurationMinutes >= 60 * 4 | ...    % …unfeasibly long times
          tt.TripDistance <= .25 | ...              % …really short distances
          tt.TripDistance >= 1000 | ...             % …unfeasibly long
                                                    %  distances
          tt.Fare <= 0 | ...                        % …negative fares
          tt.Fare > 1000 | ...                      % …unfeasibly large fares
          any(ismissing(tt),2);                     % …missing data
```

Figure 5. Example of filtering big data with MATLAB using NYC taxi trip data (21.3 GB data set).

Data Reduction

The large number of signals collected from your systems makes it difficult to find important trends and behaviors in your data. Much of the data may not be correlated with the behavior you are looking to predict or model. Being able to calculate correlations across your data, as well as using techniques such as principal component analysis (PCA), lets you reduce your data to only the signals that most influence the behavior you are modeling. By reducing the number of inputs to your model, you create a more compact model and require less processing when the model is embedded into your product or integrated within a service application.

» Get code example: ***Principal Component Analysis***

Data Processing at Scale

As an engineer or scientist, you may find that you are most efficient when working on your local desktop workstation using tools you already know. However, efficiency with big data also requires data analysis and modeling beyond your desktop workstation; you may also need to use your analysis pipeline or algorithms on an enterprise-class system such as Hadoop, without changing your code.

» Learn more: ***Working with MATLAB, Hadoop, and Spark***

Creating Models

Assume you have collected months' or even years' worth of data. What is so valuable in this data? For ***Baker Hughes***, information from temperature, pressure, vibration, and other sensors was collected over the lifetimes of many pumps. Engineers analyzed this data to determine which signals had the strongest influence on equipment

wear-and-tear. This step included performing Fourier transforms and spectral analysis, as well as filtering out large disturbances to better detect the smaller vibrations of the valves and valve seats.

The team discovered that using data captured from pressure, vibration, and timing sensors enabled them to accurately predict machine failures. They used machine learning to create the models that eventually predicted failures from these large sets of data. Machine learning is commonly used in such situations due to the large number of observations (samples) and the possibility of many variables (sensor readings or machine data) present in the data.

Machine learning techniques use computational methods to learn information directly from data without relying on a predetermined equation as a model. This ability to train models using the data itself opens up a broad spectrum of use cases for predictive modeling—such as predictive health for complex machinery and systems, physical and natural behaviors, energy load forecasting, and financial credit scoring.

Machine learning is broadly divided into two types of learning methods: supervised and unsupervised learning, each containing several algorithms tailored for different problems.

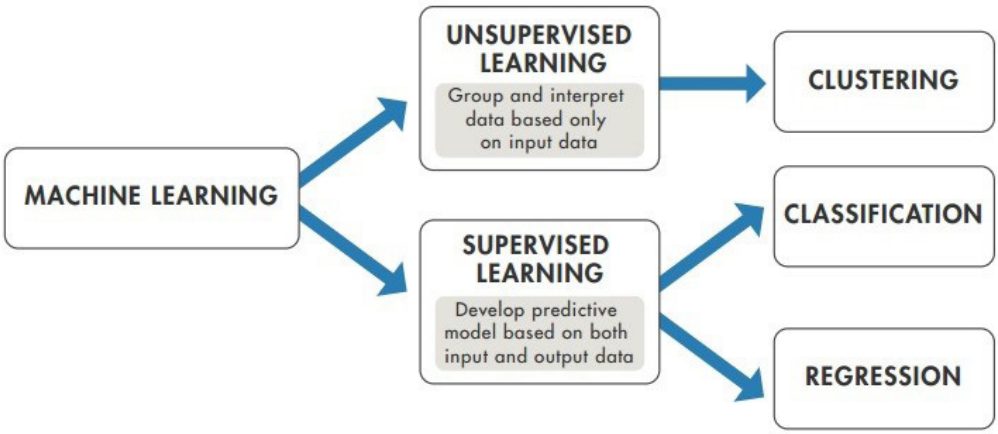


Figure 6. Two types of machine learning methods, each with different algorithms tailored for solving different problems.

Supervised Learning

Supervised learning is a type of machine learning that uses a training data set which maps input data to previously known response values. From this training data set, the supervised learning algorithm seeks to build a model that can make predictions of the response values for a new data set. Using this technique with large training datasets often yields models with high predictive power, which can generalize well for new data sets. Supervised learning includes two categories of algorithms:

- **Classification:** For response values where the data can be separated into individual classes
- **Regression:** For prediction when continuous response values are desired

Getting Started with Supervised Machine Learning

- **Select features:** Reduce variables or calculate new variables
- **Specify training and validation data:** Split your data
- **Assess the results:** Obtain measures of accuracy

Unsupervised Learning

Unsupervised learning is a type of machine learning used to draw inferences from data sets that have input data that does not map to a known output response.

- **Cluster analysis:** Common unsupervised learning method used to find hidden patterns or groupings in data

Using Models

To truly take advantage of the value of big data, you must be able to incorporate the models and insight gained from the data into your products, services, or operations.

MATLAB provides a direct path from development to the integration of an algorithm or predictive model into a device, vehicle, IT system, or web-based service, for applications such as:

- **Connected cars:** Large amounts of real-world driving data is used to develop and implement algorithms for use within embedded systems to support ADAS and automated driving capabilities. *Automated Driving System Toolbox™* lets you automate ground-truth labeling, generate synthetic sensor data for driving scenarios, perform multisensor fusion, and design and simulate vision systems.
- **Manufacturing and engineering operations:** Sensors on machinery are providing up-to-the-second information on the health and operation of refining, energy production, and manufacturing systems. Data helps to optimize the operation, yields, and up-time of these systems and requires integration as part of an enterprise IT application.

» Watch video: *Using MATLAB with PI System for Analysis and Process Monitoring*

- Design and reliability engineering: Engineering and operations groups are using data captured from aircraft under test and real-world flight conditions, and from mobile and medical devices to improve the reliability, performance, and capabilities of these devices and systems.

» Get ebook: *Wearable Healthcare Technology: Accelerating Development of Smart Algorithms*

Deploying and integrating MATLAB analytics

- *Spark Applications*
- *Models*

IT, Enterprise Applications, and Big Data

When you are ready for production, you need to work more closely with your organization's IT teams to gain access and to set up a workflow that enables you to process your data. In this new environment, using a software analysis and modeling tool that you are familiar with, and that already works with the systems your IT teams are using to store, manage, and process big data, enables you to effectively use this data in everyday activities.

At many organizations, the CIO has a focus on the proper use and protection of their big data resource, and works with IT teams to implement new policies and processes for data, including:

- **Governance:** Ensuring the integrity of the data by controlling the storage, access, and processing of data
- **Access:** Making data available to engineering, operations, warranty, quality, marketing, and sales groups
- **Processing:** Employing a specialized processing platform once the amount of data is large enough, to eliminate delays in transferring data and decreasing the time to process data

To comply with these new requirements, IT organizations adopt new technologies and platforms for storing and managing these vast and ever-increasing sets of data.

Big Data Platforms and Applications

Generally speaking, there are two categories of big data applications and platforms:: those for batch processing of large, historical sets of data, and real-time or near real-time processing of data continuously collected from devices. This second case is often referred to as streaming, and is found in most IoT applications. MATLAB supports both categories of applications – batch processing and streaming or real-time processing.

Hadoop

Hadoop is designed around distributed storage and distributed computing principles. It comprises two major subsystems that coexist on a cluster of servers, enabling it to support large data sets:

- **HDFS, Hadoop Distributed File System:** Provides a large and fault-tolerant system for storing data
- **YARN, Yet Another Resource Negotiator:** Manages the highly scalable applications that run the Hadoop cluster and process data stored in HDFS

» Learn more: ***MATLAB support for Hadoop and Spark***

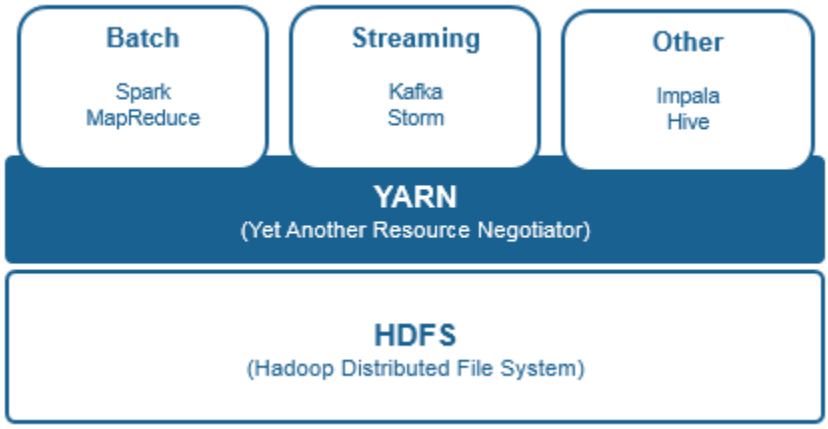


Figure 7. Main building blocks of Hadoop and some common applications that run on it.

Batch Applications and Creating Models

With batch applications, you can analyze and process historical data that has been collected over long periods of time or across many different devices or systems. Using these batch processing applications enables you to look for trends in your data and to develop predictive models that were not possible in the past with large sets of data.

Popular batch processing applications that operate on Hadoop include:

- **Spark:** A more generalized framework that optimizes in-memory operations, which is highly desirable for machine learning applications

» Get code example: ***Using MATLAB on a Spark Enabled Hadoop Cluster***

- **MapReduce:** A highly structured framework consisting of map and reduce functions, useful for large data analysis and data transformation applications

» Get code example: ***Using MapReduce with MATLAB***

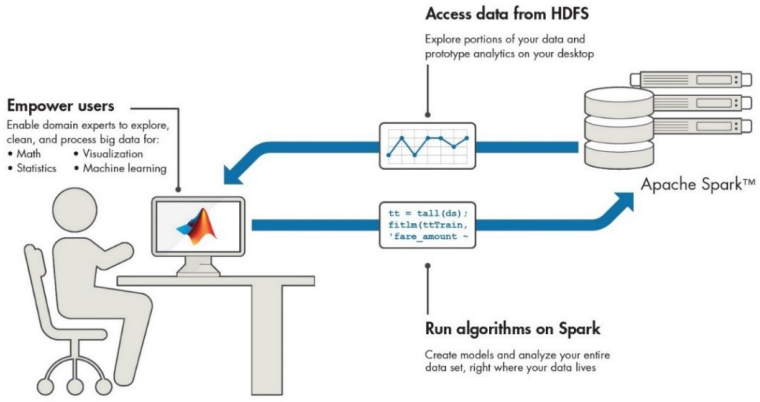


Figure 8. Using MATLAB with Hadoop and Spark

Streaming Applications and Model Integration

Using models developed from sets of historical data, along with a streaming application such as *ThingSpeak*, can add more intelligence and adaptive capabilities to your products and services.

Case Study: Analyzing Streaming Energy Data at Cadmus

Consulting firm *Cadmus* provides full-spectrum energy-efficiency support services to energy utilities throughout North America. These services include studies of energy efficiency that require extensive data collection and analysis. Traditionally, energy analysis is done by collecting data from sensors at the physical site every few months, so device malfunctions were only found when it was too late to correct them. Cadmus engineers used MATLAB and ThingSpeak to develop and deploy two systems of cloud-connected sensors for the near-real-time measurement and analysis of energy data in residential homes and energy load on building systems.

Figure 9. Internet of Things system using ThingSpeak for collecting and analyzing energy data.

In many cases, these kinds of services are usually developed in conjunction with enterprise application developers and system architects. But the challenge is how to integrate your models into these systems effectively. Porting models to another language is time consuming and error prone, requiring extensive work each time a model is updated. Developing predictive models in typical IT languages is difficult. Engineers and scientists who have the domain expertise required for developing these models are not familiar with the languages, and these languages don't always include the functionality needed to adequately process and create models from engineering and scientific data.

Enterprise application developers should look for a data analysis and modeling tools their engineers and scientists are already familiar with, and that also provides the domain-specific tools they need. These tools must also scale for use in developing models and large data sets using Hadoop-based systems that provide capabilities such as a highly robust application server and code generation, enabling a direct path for deploying models into enterprise applications. MATLAB provides a processing platform that scales from your local desktop to big data systems such as Hadoop.

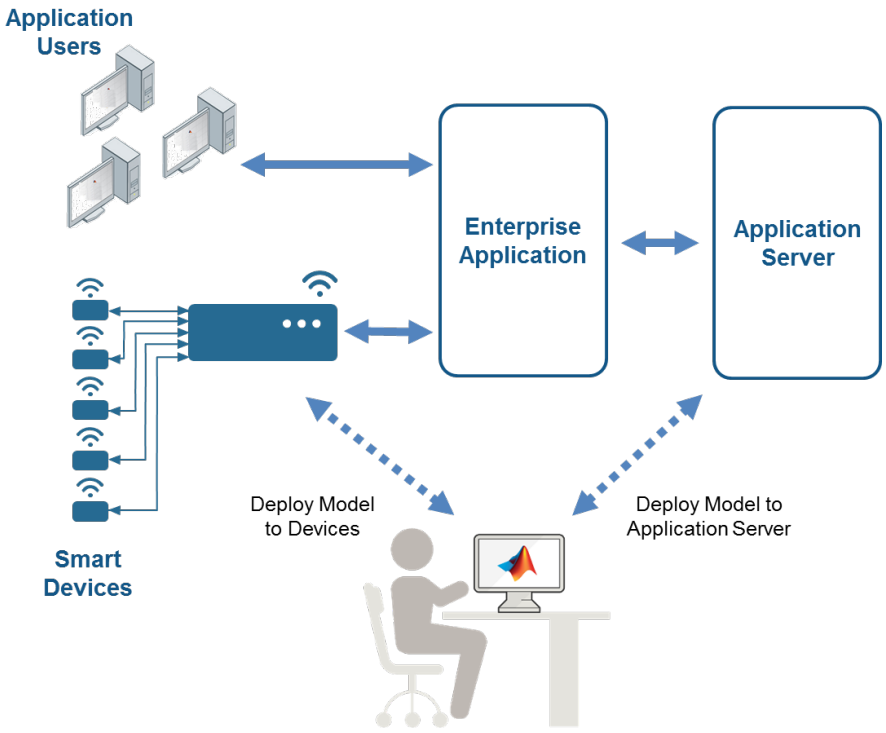


Figure 10. Integrating models with MATLAB.

Scientists, Engineers, and IT

Big data systems employ a variety of ways to store, manage, and process big data. By working closely with your IT team and using a platform such as MATLAB, you can create a workflow that is familiar, enabling you to easily and efficiently work while gaining insight from a vast collection of data.

IT managers and solution architects can use modeling capabilities in MATLAB to enable the scientists and engineers in their organizations to develop algorithms and models for smarter and differentiated products and services. Simultaneously, you are also enabling your organization to rapidly incorporate these algorithms and models into your products and services by leveraging production-ready application servers and code generation capabilities.

The combination of a knowledgeable domain expert who has been enabled to be an effective data scientist, along with an IT team capable of rapidly incorporating their work into the services and operations of their organization, makes for a significant competitive advantage when offering the products and services your customers are demanding.

Work with IT to Enable Your Big Data Workflow

- **Big data platforms:** Find information on the platforms and applications your IT teams are using
- **Data formats:** Understand the structure of your data
- **Processing needs:** Understand what is it that you need to do with your data
- **Enable your workflow:** Work with your IT teams early to enable access to your data and plan your model integration strategy

Learn More

- White Paper: ***Predictive Analytics with MATLAB***
  Use machine learning with big data for engineering-driven analytics.
- White Paper: ***The Manager's Guide to Solving the Big Data Conundrum***
  Learn how to extend or replace Excel for more productive work with big data.
- Video: ***Why Use MATLAB for Big Data?***
  Roy Lurie, VP of Engineering for MATLAB Products, underscores how MATLAB is more convenient and scalable than ever.
- Video: ***Mondi Gronau Develops a Predictive Maintenance and Process Monitoring System Using Machine Learning***
  See a demo of working with big data in MATLAB, and download the code and data sets.