Using MATLAB on Apache Spark for ADAS Feature Usage Analysis and Scenario Generation

Sanjay Abhyankar

Supervisor & Technical Expert

ADAS **C**ustomer **I**nsight & **D**ata **A**nalytics (CI&DA)

Ford Motor Company

Abstract

- In the past, engineers download terabyte-sized ADAS datasets to look for edge cases. This approach consumes huge amount of network bandwidth and local storage space. We created a new and more efficient way, which utilizes MATLAB to access Apache Spark resources to decode, analyze data, and search for edge cases right on the Hadoop file system. It dramatically improves throughput and reduces the amount of data downloaded to the engineer's workstation.
- This approach was successfully used to analyze ADAS feature usage from the CAN traffic on Ford's **Big-Data-Drive** fleet of vehicles. It has been deployed for all future Big-Data-Drive vehicle analysis.

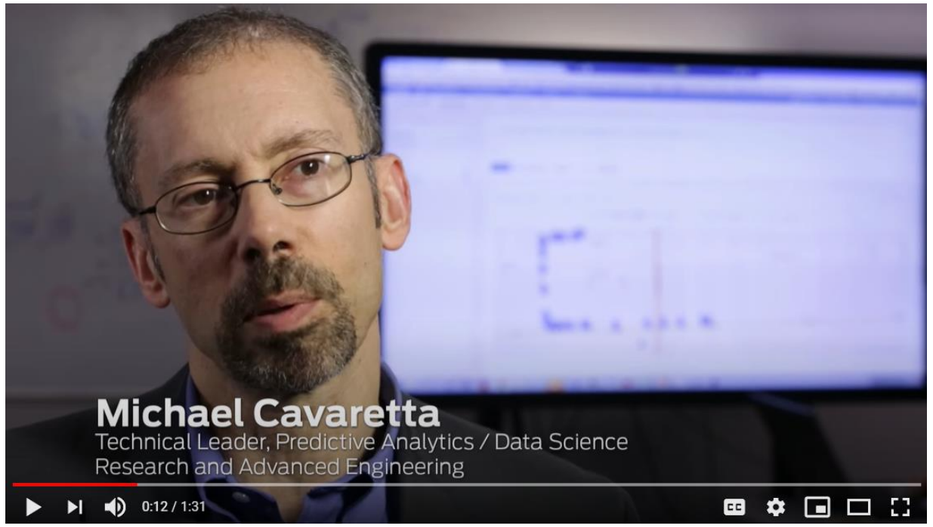
The Team

- Ford :
  - Sanjay Abhyankar, Supervisor and Technical Expert ADAS Analytics
  - Heesuk Kang, ADAS Data Analytics Engineer
- MathWorks:
  - Will Wilson – Senior Application Engineer, MathWorks

Our Internal Team

- The ADAS Customer Experience & Data Analytic Team was formed by engineers from various disciplines such as NVH (Noise Vibration Harshness) and Powertrain Engineering
- Team had decades long experience using MATLAB and other analytic tools, however they did not come from an IT/Big Data background
- Needless to say there were a lot of learnings along the way !
- Today's presentation will focus on the application of MATLAB Big-Data-Toolset to the Big-Data-Drive effort begun at Ford in 2015

What is the Big-Data-Drive ?

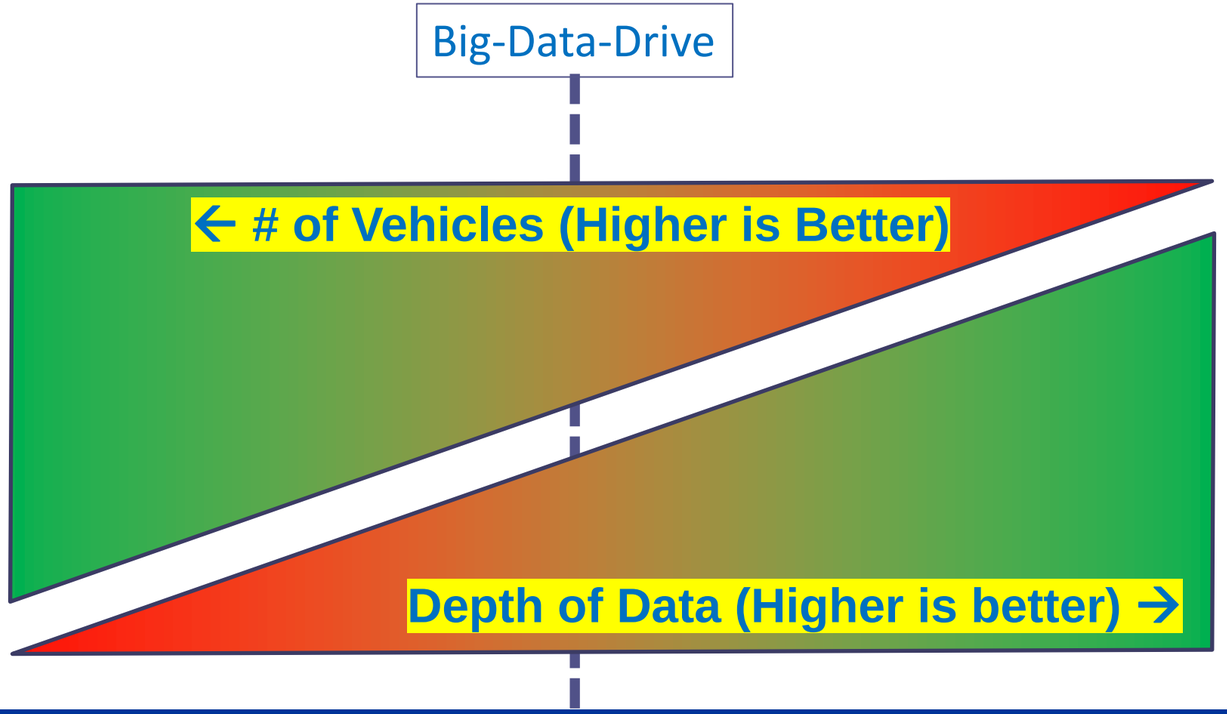


https://www.youtube.com/watch?v=uovijOpB7Jw

- Big-Data-Drive is one of the several connected vehicle data collection efforts at Ford. It expanded significantly since it's inception in 2015
- Full CAN Traffic data is collected and sent to the Ford Cloud.
- Data from drivers that have signed an agreement to share data and have data logger installed was analyzed

Vehicle Data Collection – Big Picture

- Most Vehicle Data Collection efforts balance # of Vehicles and Depth of Data
- BDD falls in the middle of this balance



A common need among engineering groups is to look for edge cases in BDD. Downloading terabyte-sized datasets to look for edge cases would consume huge amount of network bandwidth and local storage space.

Developing the workflow

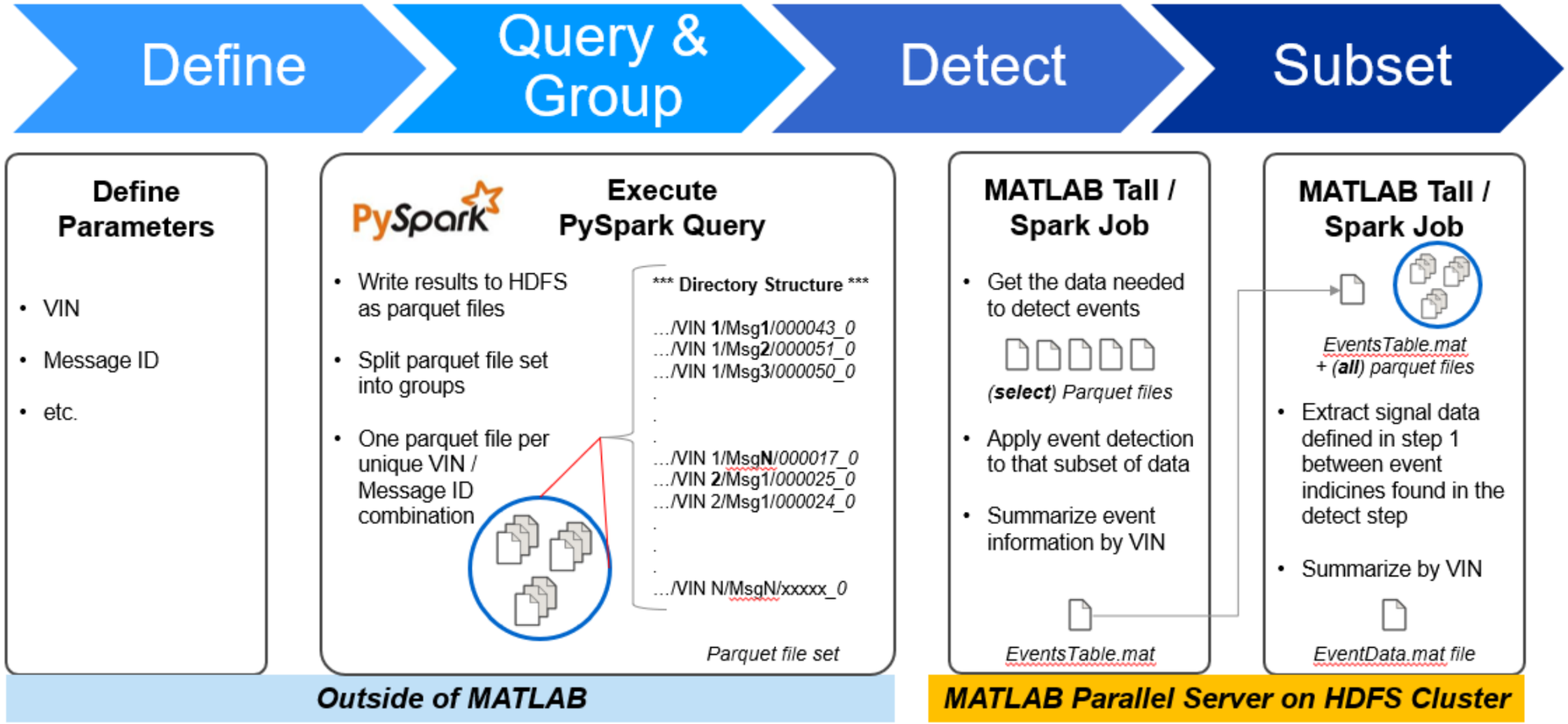
Initial “Bucket” of Tools

PySpark MATLAB Qlik

Streamlined Workflow

Team tried several iterations of tools and settled on the final Workflow. The next few pages will take us though the key steps in the workflow

Final Workflow



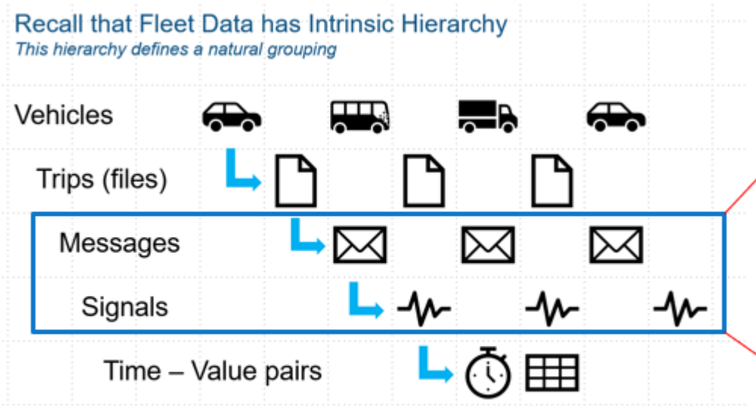
MATLAB programs decode, analyze data, and search for edge cases right on Hadoop file system.

The optimized workflow was developed in collaboration with MathWorks

Workflow Step # 1 Define

The analysis begins with the Engineer setting up a Configuration Table that defines what data is to be extracted.

- This step defines things such as the list of VIN's, the signal set & relevant message ID's, VIN's, etc.



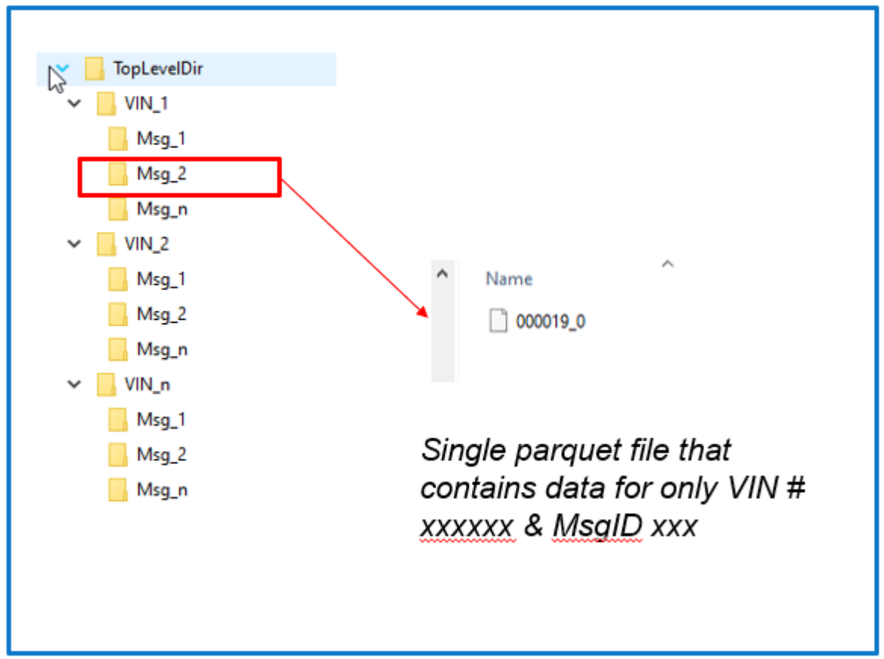
MessageID	SignalSet
118	{1×2 string }
119	{1×3 string }
126	{1×2 string }
374	{["TrnRng_D_Rq"]}
514	{["Veh_V_ActlEng"]}
936	{1×2 string }
938	{["ApaMde_D_Stat"]}
947	{["Ignition_Status"]}
1072	{["OdometerMasterValue"]}
1122	{1×6 string }

Workflow Step # 2 Query & Group

The Engineer then queries the Hadoop based Big-Data-Drive

- The query is executed in PySpark and the results are stored in a VIN-Message Pair Structure on HDFS

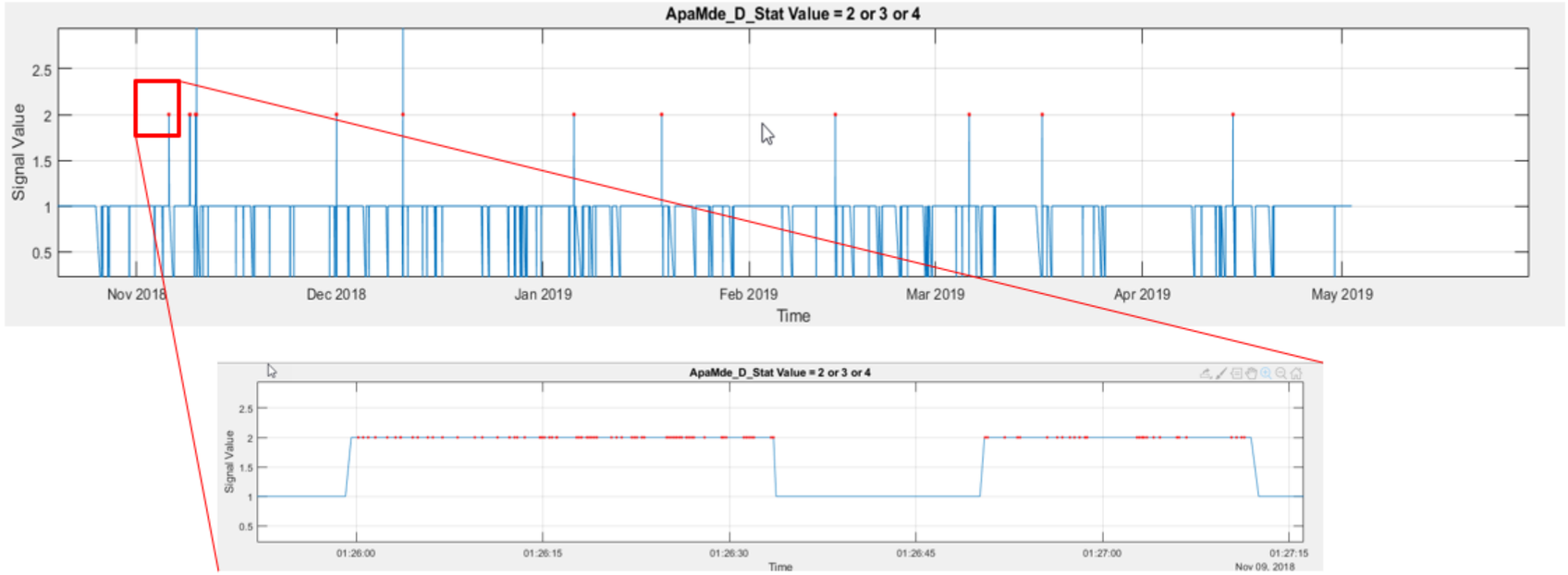
Example output from PySpark query



Workflow Step # 3 Detect

The next step is Event Detection. The engineer specifies a set of event triggers based on multiple signals to extract only the data of interest.

- The CAN data is first decoded using the MATLAB Vehicle Network Toolbox (VNT)



Efficient MATLAB algorithms running on Apache Spark were developed to detect events from CAN logs

Workflow Step # 3 Detect

Step 3 concludes with the creation of an Event Dictionary like the one shown below. This is a high level look at the events of interest.

VIN	EventNumber	EventStart	EventStop	EventDuration_s
"1FMCU9D	0	NaT	NaT	0
"1FMCU9D	1	18-Dec-2019 04:31:10	18-Dec-2019 04:32:42	92.023
"1FMCU9D	2	18-Dec-2019 04:33:11	18-Dec-2019 04:34:21	69.837
"1FMCU9D	3	18-Dec-2019 04:55:09	18-Dec-2019 04:56:44	94.898
"1FMCU9D	4	04-Jan-2020 01:55:28	04-Jan-2020 01:56:08	39.841
"1FMCU9D	5	04-Jan-2020 02:09:00	04-Jan-2020 02:10:11	71.53
"1FMCU9D	6	04-Jan-2020 05:30:51	04-Jan-2020 05:32:26	94.898
"1FMCU9D	7	22-Jan-2020 01:41:14	22-Jan-2020 01:41:56	41.122

Workflow Step # 4 Subset

In Step 4, the event dictionary developed in the previous step is used to extract time logs and create the final EventData file which has all the information needed for further analysis.

8×53 table

VIN	EventNumber	EventStart	EventStop	EventDuration_s	ApaChime_D_Rq	ApaMde_D_Stat	EI
[redacted]	0	NaT	NaT	0	{ 0×0 double }	{ 0×0 double }	{
	1	18-Dec-2019 04:31:10	18-Dec-2019 04:32:42	92.023	{1238×1 timetable}	{60×1 timetable}	{1
	2	18-Dec-2019 04:33:11	18-Dec-2019 04:34:21	69.837	{1118×1 timetable}	{44×1 timetable}	{1
	3	18-Dec-2019 04:55:09	18-Dec-2019 04:56:44	94.898	{1680×1 timetable}	{71×1 timetable}	{1
	4	04-Jan-2020 01:55:28	04-Jan-2020 01:56:08	39.841	{ 554×1 timetable}	{32×1 timetable}	{
	5	04-Jan-2020 02:09:00	04-Jan-2020 02:10:11	71.53	{1220×1 timetable}	{56×1 timetable}	{1
	6	04-Jan-2020 05:30:51	04-Jan-2020 05:32:26	94.898	{1329×1 timetable}	{73×1 timetable}	{1
	7	22-Jan-2020 01:41:14	22-Jan-2020 01:41:56	41.122	{ 696×1 timetable}	{33×1 timetable}	{

This is "EventsData" we found in the Detect step

This is signal data that matches the time ranges to the left

The workflow shown in the previous pages was used to analyze several ADAS features.

Next few pages show an early example application of this workflow for Active Park Assist

Example Use Case – Active Park Assist Plus

- ***Active Park Assist Plus*** was introduced on the 2020 Lincoln Corsair, Aviator and Ford Explorer, Escape
- This is a significant improvement over ***Enhanced Park Assist*** of previous model years
- Need to compare and contrast usage of the new system was identified
- A Big-Data-Drive analysis was initiated
- A total of **48** CAN signals spread over **13** messages were selected for analysis

Example Use Case – Active Park Assist Plus

- Event Detectors based on multiple parking signals were designed
- Offline Sensor Fusion using GPS, Wheel Counts, Inertial Measurement unit was used to accurately model the vehicle and surroundings.
- An event data dictionary was developed and the success and failure of every park assist event was determined.
- Data from the study was fed to Ford's Simulator and also for ADAS Scenario Generation
- Findings from this study have helped guide the future development of Active Parking Features

All the CAN message decoding, data processing, event detection, Sensor Fusion was done using MATLAB and a multitude of toolboxes (such as a customized version of the VNT – Vehicle Network Toolbox)

Application for other features

The Toolchain developed in Parking has since been extended to analyze other ADAS features such as

- Trailer Backup Assist
- Adaptive Cruise Control
- Lane Centering
- Reverse Brake Assist

In addition it has been used to study ADAS related phenomenon such as

- Hands off Analysis during Lane Centering
- Vehicle Wander Analysis during Lane Centering
- Occurrence of Forward Collision Warnings and many other ADAS related issues

Example of Results from the Analysis

This shows the results of Lane Centering Wander Analysis

In Conclusion

- An efficient workflow to analyze Big-Data-Drive vehicle CAN Traffic has been developed
- It uses a variety of MATLAB's strong Big-Data capabilities and Toolboxes in conjunction with Apache-Spark resources
- The workflow has been used successfully to provide rich insights into a variety of ADAS features and phenomenon
- Many Connected Vehicle Big Data efforts are limited to basic summary operations such as # of events, average quantities. They also analyze only a limited # of signals. This study is a comprehensive look at the entire "User Experience" and creates Scenarios and Experiences and not just Statistics
- Ford and MathWorks will continue to look for more opportunities to extend this fruitful collaboration.