Case Studies: Big Data Analytics for System Health Monitoring

Dinkar Mylaraswamy[1], Brian Xu[1], Paul Dietrich[1] and Anandavel Murugan[2]

[1] Honeywell Aerospace, Golden Valley, MN, USA

[2] Honeywell Technology Solutions Limited, Madurai, India

Abstract. *This paper describes a case-study where we built and exercised a cloud computing framework with machine learning (ML) algorithms to improve the accuracy of Auxiliary Power Units (APU) health monitoring. An APU is a small turbo machine that flies on all commercial transport airplanes. The paper describes the objective of our study, sources of available data, the ETL scripts to populate the underlying HBase tables and two examples. In one example machine learning algorithms operating on multiple data sources produce useful insights to increase our ability to predict APU wear from 39% to 56%. In the second example, it increased our ability to predict shutdown events from 19% to 60%. This case-study illustrates the effectiveness of big data analytics and tools to discover additional insights that can further reduce operational interrupts arising from airborne equipment problems.*

Keywords: Big Data Analytics, Machine Learning, System Health, Case Studies, Cloud Computing.

1 Introduction

In recent years, the aviation industry has witnessed a steady increase in more data being collected by Aircraft Condition Monitoring Systems (ACMS). Data volumes ranging from 5 ~ 10 megabytes per flight hour (each aircraft) are routinely collected by onboard recorders and sent directly over airport Wi-Fi and GSM wireless networks without incurring the costs associated with ACARS messaging. Advances in IT and software that allow secure movement of data from airplanes provide an ideal framework for embedding statistical machine learning algorithms that can discover sweet-spots in global operations can feedback into day-to-day actions. This cloud computing based information network created by these connected aircrafts (a part of Industrial Internet) hold the potential of providing valuable knowledge needed to maintain profitability in an economically challenged civil aviation industry.

Technically, one of our study areas that can benefit most from current big data analytics is to reduce maintenance cost of high-value aerospace assets. For example, a recent GE article estimates a $250M savings in engine maintenance cost [1] is possible using insights gained from machine learning (ML), data mining and knowledge discovery. While the actual savings depend on specific aftermarket business policies, such case studies have been widely reported, clearly indicating the potential of data mining methods for discovering useful business insights from big data. In this paper, we describe our approach to data mining using data collected from auxiliary power units (APUs).

Our primary objective is to develop Predictive analytics. Specifically predict events and failures in an APU before they cause an operational interrupt. Our approach is to analyze past historic data available from two distinct sources: data collected while the APU is operational and data when the APU is being repaired. Currently these data sources are separated and in many cases, the design schemas within these databases are poorly documented. Further, the schema is not uniform when dealing with different APU makes – since these products were introduced at different time periods. The repair records are aggregated by three global centers—each of which uses different tools that introduces site-specific biases. The challenge lies in making multiple queries, retrieving the correlated data and developing predictive analytics. Data quality (missing and incorrect entries) makes this problem difficult—using a common APU serial number and timestamp is not sufficient. It is this problem statement we try to address using big data tools.

This paper is organized as follows: Sections 2 introduces the problem along with a brief description of the APU. Section 3 describes our cloud-based ML framework and its key components. Section 4 describes the Parser used for data ETL to populate data into our HBase and HDFS. Analysis and results are presented in Section 5. Conclusions and significance of this work is presented in Section 6.

2 Problem Statement

An auxiliary power unit (APU) is a small gas turbine engine that provides pneumatic and electrical power to the airplane. The main functions of an APU are listed below:

1. High pressure bleed air for starting the main propulsion engines
2. Provide air to the environmental control system and cabin pressurization
3. Drive a generator to provide electric power for the airplane

Though an APU does not provide propulsion, the internals are as complex as large jet engines. A typical APU for commercial transport aircraft is broken up into three main sections – the *power section*, the *load compressor* and the *gearbox*.

Two ongoing programs within Honeywell are relevant to the discussion presented in this paper.

1) Predictive Trend Monitoring and Diagnostic (PTMD) program which reports data from APUs while they are operational
2) Product In-Service Performance (PIPS) program which reports data from APUs after they are removed from the airplane and sent to the repair shop.

Honeywell's Predictive Trend Monitoring and Diagnostic (PTMD) services provide APU usage and sensory data downloaded through the Aircraft Communication and Addressing Reporting System (ACARS). Stated simply, it provides sensory data while the APU is operational and installed on the aircraft.

The general description of PTMD is given in [7]. However it suffices to mention the following three key outputs from the PTMD system:

A. **AHRS**: This is called the APU usage hours or AHRS. This is similar to the odometer reading in a car, albeit it measures the cumulative "time duration" rather than distance.
B. **EGT margin**. Within the APU fuel is converted to mechanical energy. The exhaust gas from the APU is the energy that is not converted to useful work. EGT margin measured in degree Celsius is a measure of the thermodynamic efficiency of the APU.
C. **PB margin**. In order to start the main propulsion engine, bleed air provided by the APU must meet certain pressure and flow conditions. The bleed margin (abbreviated as PB margin) measures the APU's ability to meet these constraints.

As the APU ages, both the EGT and PB margin decreases steadily with AHRS and at some point, the APU operation is no longer economically viable and hence it is removed from the aircraft and sent to the repair shop for repairs.

When the APU is received at the repair shop, Honeywell's Product In-Service Performance System (PIPS) captures actions performed at the repair shop. The general description of PIPS is outside the scope of this paper. However it suffices to mention the following two three annotations from the PIPS:

a) **Symptom**: This is an enumerated text describing the reason for removing an APU, typically provided by the airline operator. Examples include: no-start (APU is not able to start), auto-shutdown (the APU is shutting down due to some internal problem), and high-wear (the APU efficiency has decreased beyond its economic threshold.
b) **Description**. This is a free-form text provided as a summary of the repairs performed on the APU. It includes parts replaced, damage observed, and possible primary cause and secondary effects.

Our primary objective is to develop analytics aimed at predicting (1) severe wear, and (2) erratic behavior arising from auto-shutdowns (the APU switches itself off to protect internal damage). Figure 1 summarizes our problem statement.

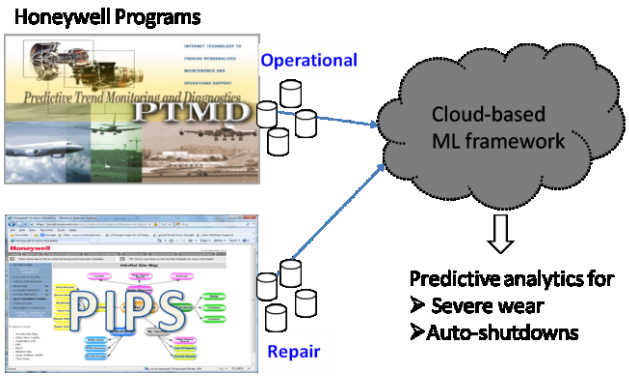

Figure 1: Summary of data available for applying ML methods and mining

Next we briefly describe the cloud-based ML framework used to develop these predictive analytics.

3 Cloud-based ML Framework

Details of the cloud-based ML framework are described in [2]. Here we present a short description. The cloud is a Hadoop cluster of Linux machines. Salient features shown in Figure 2 are described below. Additional details are provided in [2].

1. Our data storage strategy uses HDFS—distributed file system built on top of Hadoop
2. Mahout provides machine learning algorithms for clustering, classification tasks. The list also

includes algorithms classifying textual repair descriptions.

3. The HBase has two technical components: (a) Convenient base classes that support Hadoop MapReduce jobs and functions with HBase tables; and (b) Query predicate pushes down via server side scan and gets filters that will select related data for track management systems.
4. A user can plug-in algorithms based on domain knowledge. Our initial focus is on algorithms developed in Mathwork's Matlab and the R language.

Information from the two sources (PTMD and PIPS) arrives as *reports*. We considered two aspects while arriving at the storage strategy needed for analytics: (1) Reports need to be logically grouped, based on typical access pattern, (2) File should also be organized in a way to support efficient map reduce jobs. When we initially stored our reports in native report format, map reduce jobs took more time than the desktop execution. The reason for this behavior is as follows: map-reduce jobs are efficient in handling files smaller than the HDFS block size. We found typical report size (from both PTMD and PIPS) was less than few MBs while the HDFS block size is 64MB. This not only slows down map reduce jobs, it is also not good for HDFS.

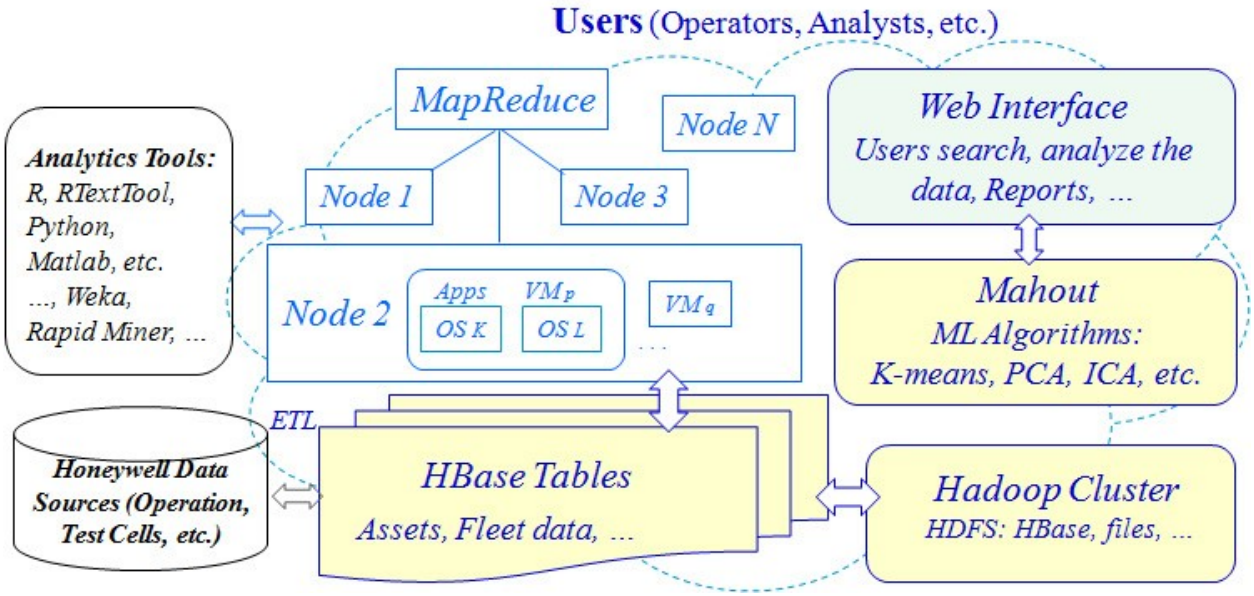

Figure 2:
Architecture and Components of the Cloud-based ML Framework

Honeywell has been supplying APU for the commercial aerospace industry for several decades. As a result a large amount of legacy data was available for predictive analytics development. Organizing this legacy data was an important step in our process. The legacy data ETL is described in the next section.

4 Data ETL to HBase and HDFS Using Parser

An important first step in our process was to import legacy APU reports. The format of these reports range from comma-separated-values ASCII text files to binary files with specific encoding. The import process into HBase and HDFS is carried out using our Python package (*importer*.py). The importer module is a script with a number of different options to control the source of input data (SQL database, CSV files, etc.), its specific format and the destination for the data. This information is supplied in a configuration file. The HappyBase library [3] is used for access to HBase (via the Python Thrift gateway) and the Pydoop package [4] is used for access to HDFS. Our importer script has a valuable debugging feature that allows us to see exactly what would be added to any HBase table without committing any changes. Figure 3 shows the screen shot illustrating the usage of this script for importing legacy data from the PIPS source.

The destination for the imported data extracted from legacy reports are a series of HBase tables. The HBase table design is described next.

```
·(thrifty)tt-a664-4 [1250]%
·(thrifty)tt-a664-4 [1250]%
·(thrifty)tt-a664-4 [1250]%
·(thrifty)tt-a664-4 [1250]% ./importer/importer.py --index-file=./PIPS/PIPS.APU13
·3.2013_06_17.csv --field-separator=',' --dry-run --leave-source PIPS
```

Figure 3:
A Screenshot of importing PIPS data into our HBase and HDFS

An important consideration in HBase table design is row key design. Each row in HBase table is identified by a random unique key. Best practice is to make this key a composite key matching the querying pattern. A composite key is created by adding multiple attributes of data stored in the row. In our case, users would search reports based on the asset model, serial number and operator. They would then filter the reports based on a time range. So row key we chose had the following format:

<model>:<serial>:<operator>:<timestamp>

To keep the row key unique, random characters are appended to the row key. With this row key format, a scan can be done with a row key filter. Row key filter based scan just looks up the row keys and hence it is faster. Random characters appended to the key also help in avoiding region server hotspotting. Region server hotspotting is a case when more data is accumulated in one region server when the row keys are logographically closer. This overloads one region server when the other region server is underutilized.

The other aspect of HBase table design is the column and column family design. In our case, for all the report tables, we have two column families. One stores the report data and other stores the asset details. The column name for these families is listed in Figure 4.

```
'ast:mod',      # APU Model
'ast:ser',      # APU Serial Number
'ast:ow',       # APU Owner
'ast:OC',       # APU Ownercode
'rpt:mis',      # Report Mission
'rpt:pth',      # Report Path
'rpt:tm',       # Report Date
```

Figure 4: A Screenshot of the column families

The next section describes the analysis and the results obtained by applying ML methods.

5 Analysis and Results

As stated earlier our objective was to explore ML methods that could discover (1) factors that indicated severe wear on the APU, and (2) factors that have led to an auto-shutdown event. The end goal of this exercise was to encode the resulting understanding as a predictive analytic algorithm for continuous monitoring of APU health. The analysis performed on the large volume and variety of data being collected by the PTMD and the PIPS program. Our intent in this section is to summarize the effectiveness of the ML methods.

A fundamental concept in our analysis is called a lifecycle point (LCP). LCP for an APU is the time interval between its n^{th} repair shop visit and the $(n+1)^{th}$ repair shop visit. $n = 0$ is a special case when the APU leaves the manufacturing shop and enters the operating fleet for the first time. The LCP is explained pictorially in Figure 5.

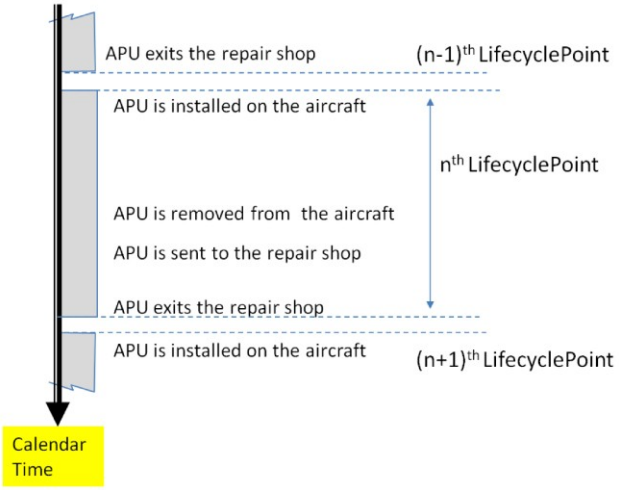

Figure 5: LifeCyclePoint is the 'atomic unit' for applying ML methods

The primary indicators available for ML analysis while the APU is in the repair shop are the symptoms and free-form text description of maintainer actions and observations. The free-form text was processed using a combination of Bayesian and Regression text-mining methods to produce target categories/classes that summarized airline operator describing the reason for removing the APU and the repair performed. Two specific categories of interest in this study were those related to severe wear and auto-shutdown.

In section 2 we described the three primary indicators available for ML analysis while the APU is operational. These are: EGT margin, PB Margin and

Usage Hours (AHRS). As the APU ages, the both the EGT and PB margin decreases steadily with AHRS. Typical trend of EGT Margin within a lifecycle point is shown in Figure 6.

It is well known in the industry that the EGT Margin is a good indicator of APU general wear [5]. The margin becomes smaller (approaches zero) as the APU wears and becomes negative as the APU wears severely. Further, an APU that has been operating longer is more likely to develop internal faults and hence more likely to exhibit auto-shutdowns. In addition to the numeric values of EGT and PB margin at specific AHRS values, we pre-processed the trend lines to retain the following features.

1. Slope – this measured the rate at which the EGT and PB margins changed with respect to AHRS
2. Jumps – this measured large changes in the EGT and PB margins between two consecutive flights
3. Slope changes – this measured inflexion points in the EGT and PB trend line where the slope changed values.

Figure 6: EGT margin as a function of APU usage.

A total of $3 + 2 \times 3 = 9$ features were available for the clustering and classification ML algorithms.

Both the PTMD and the PIPS programs have been tracking APU lifecycles for several years. They span three APU models used by more than 100 global airline operators. Lifecycle points available for ML are summarized in Table 1.

Table 1 : Scope of ML analysis

APU	# of Lifecycle points	Features per lifecycle point
Model A	1001	9
Model B	764	9
Model C	629	9

We applied three ML classification methods—Random Forest (decision trees), Support Vector Machine and Naïve Bayes. We assume the reader is familiar with three standard ML algorithms. The choice of these methods was motivated by the fact that they represent non-overlapping approaches for capturing the decision boundaries. We used a uniform weighted averaging (weight = 1/3) to fuse the output generated from each of the method. In addition to quantifying the accuracy of detection, we were also interested in understanding how much each of the nine features contributed to our ability to predict severe wear and auto-shutdown events.

Figure 7 shows the results for the severe wear category. It is well known in the industry that the EGT Margin is a good indicator overall APU wear [5]. However, as shown in Figure 7, it can only predict he severe wear event only 39-out-100 times. The ML methods discovered three additional features that can increase the effectiveness in predicting APU severe wear from 0.39 to 0.56. These additional features were: (1) the slope of the EGT margin trend, (2) the slope of the PB margin, and (3) jumps in the EGT trend line.

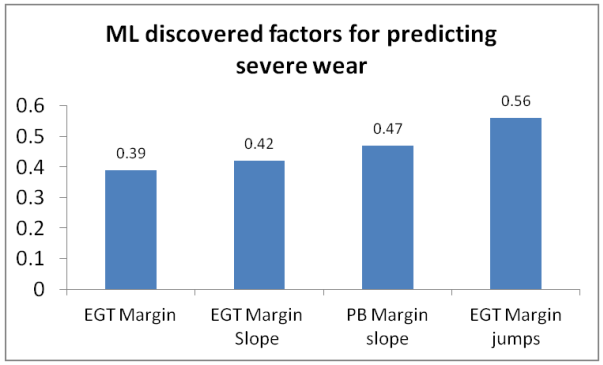

Figure 7: New features discovered by the Cloud-computing ML framework for predicting APU wear.

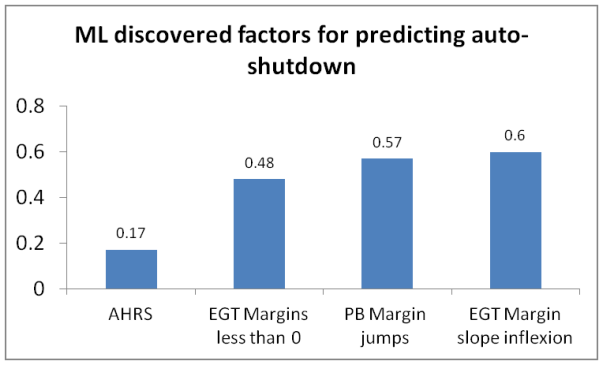

Figure 8: Features discovered by the Cloud ML framework for predicting APU auto-shutdown.

Figure 8 shows the results for the second example—predicting APU auto-shutdown events. While the APU age characterized by the AHRS can only

provide 17% prediction accuracy, the framework discovered three new features that increased the accuracy to 60%. As shown in Figure 8 these features were: (1) EGT margins becoming negative, (2) jumps in the PB trend line, and (3) inflexions in the EGT margin trend line.

6 Conclusions

In this paper, we described the analysis and the results we obtained using our cloud-based ML framework. Specifically we described the use-case where we used the framework to gain additional insights for improving the accuracy of Auxiliary Power Units (APUs) health monitoring. The paper described the Honeywell's' data source we used, and the customization of our previously developed cloud-computing framework [2] with respect to data ETL and the HBase table design. Among various ML algorithms, we selected three algorithms to investigate how big data analytics can be scaled and applied to solve real aviation industrial problems.

We illustrated the effectiveness of the cloud computing ML framework using two examples aimed at predicting (1) severe wear, and (2) erratic behavior arising from auto-shutdowns. In the wear example, the ML framework discovered three new insights that can increase the accuracy from 30% to 56%. In the auto-shutdown case, the ML framework discovered three insights that improved the accuracy from 19% to 60%.

Both these examples clearly illustrate the value-provided cloud computing tools and big data analytics can provide to the aerospace industry. For example, insights gained by analyzing decades of historical data can be converted into predictive analytics for system health monitoring. The resulting service would provide alerts to an airline maintainer and minimize operational interrupt. This analysis can be done in real-time using streaming data from more connected airplanes.

Our future work will focus on expanding our big data analytics for developing data-driven predictive analytics for other aerospace assets.

7 References

1. Evans, P. C., & Annunziata, M. (2013). Industrial Internet: Pushing the Boundaries of Minds and Machines. http://files.gereports.com/wp-content/uploads/2012/11/ge-industrial-internet-vision-paper.pdf
2. B. Xu, D. Mylaraswamy, and P. Dietrich. A Cloud Computing Framework with Machine Learning Algorithms for Industrial Applications. WorldCom ICAI, July 2013, Las Vegas, USA.
3. HappyBase, http://happybase.readthedocs.org
4. Pydoop, http://pydoop.sourceforge.net/docs/
5. Honeywell PTMD: https://commerce.honeywell.com/webapp/wcs/stores/servlet/eSystemDisplay?catalogId=10201&storeId=10651&categoryId=42999&langId=-2