Using Big Data Technologies and Analytics to Predict Sensor Anomalies

Joseph Coughlin
Rohit Mital
Will Fu
Stinger Ghaffarian Technologies Inc.
Colorado Springs, CO
joe.coughlin@sgt-inc.com

ABSTRACT

A goal of big data analytics is to help leaders make informed and rapid decisions by analyzing large volumes of complex data, as well as other forms of data that may be untapped by conventional analyses, and presenting it in a form that facilitates decision making. Big data analytics is the process of examining large data sets containing a variety of data types to uncover hidden patterns, unknown correlations, and other useful information.

Sensors typically record significant amounts of data but it is often not exploited except in special cases and after historically large amounts of analysis time. Big data analytics provides a mechanism to routinely monitor these data sets while also providing insight into anomalous events, such as are encountered in large sensor systems such as those in the space surveillance network.

In this study, we simulate recorded data from a notional radar or optical sensor and use big data technologies and the analytics to process the data to analyze and predict sensor performance. This study focuses on data products that would commonly be analyzed at a site and how big data technologies can be used to detect anomalies. This study shows how the ability to rapidly drill down into the data enables an analyst or decision maker to assess potential system anomalies.

This study shows how current technologies and predictive analytical techniques can be used to view the data, detect and explain anomalies, and predict preventative maintenance actions in a timely and automated manner.

1. APPROACH

This study is an attempt to determine how big data processing and predictive analytics (BDPA) can be used to determine and predict sensor anomalies. BDPA analyzes large datasets to identify trends and correlations that may be missed by traditional analytic techniques with limited datasets. A common approach in BDPA is to build a “dashboard” showing high level data, or summaries of data, that can be monitored and an operator alerted when anomalies are detected. The power of BDPA is the ability to “drill-down” and extract meaningful information from subsequent levels of detail data. Since these techniques rely heavily on adequate long term data, it is important that site data are maintained.

For this study, we simulate recorded data from a notional sensor and:

1) Introduce data anomalies, such as aberrant noise or other fault conditions, into the recorded data.
2) Calculate relevant parameters from the recorded data.
3) Determine which sensor anomalies can be derived from the calculated parameters.
4) Examine big data analytical methodologies that could be used to analyze sensor anomalies
5) Present a fully interactive “dashboard” and subsequent drill-down analytics that show successively more meaningful data.
6) Evaluate methodologies that could be used to predict sensor anomalies from the recorded data, such as regression, pattern recognition, or machine learning.

The simulation of the recorded data mirrors common data recorded at sites. Typical examples are track and observation data and associated parameters that deal with how well as system is performing. These data are generated by propagating the satellite catalog through the field of view of the sensors and computing tracks or observations consistent with what might be expected. We also generate a set of “truth” data for comparison. In order to examine BDPA methodologies we generate the data for long time periods. In this study, we have a baseline of 150 days of truth data. The simulated data contains Gaussian noise that reflects what should be normal amounts of noise in the data. Therefore, any anomalies must override the natural noise levels encountered in the data to be detectable. Using larger datasets, which is made possible through big data techniques, also allows a greater chance of being able to dig valuable information out of the data that might be lost otherwise.

The processing flow and architecture of our big data processing is shown in Figure 1. To handle large datasets, automated processing that efficiently inserts data into a database while running parallel big data algorithms on the simulated data is critical. Data is input to the system using Apache Kafka [1], a publish-subscribe messaging system to handle large data sets. Apache Storm (Storm) [2] is a distributed soft real-time data processing platform that we are using to run algorithms in parallel. Computations on Storm, are performed using a "topology", where a topology is a graph of computational nodes. Each node in a topology contains processing logic, and links between nodes indicate how data should be passed around between nodes. The topology is submitted to the Storm Cluster, a cluster of virtualized servers that dedicates supervisors and workers to complete the steps outlined in the code. In our case we have two topologies, a storage topology for storing data, and a calculation topology which is responsible for performing data calculations. A Storm topology processes messages until it is stopped. For a sensor application, there would be a constant stream of data that is processed by the Storm topology.

MongoDB [3] is the NoSQL database of choice, which stores data points in key-value JavaScript Object Notation (JSON) type objects and is scalable, a huge advantage when considering the demands of big data. The raw sensor data is stored in the MongoDB database, using the storage topology. The raw data are extracted and processed by the server running the Apache Storm calculation topology which contains the core processing algorithms. In our MongoDB implementation, the simulated data is stored in one set of collections, while the results of the computation are stored in a separate collection. No user-interaction is required and the separation of collections allows the visualization to be performed in parallel with computational activities.

Data are extracted from the database using the Tableau software package which is a client-server architecture. Some of the figures below reflect the data visualization capabilities of Tableau. Using Tableau we built a series of dashboards to provide operator interaction and dynamic drill-down into both the results and raw data stored on the MongoDB database.

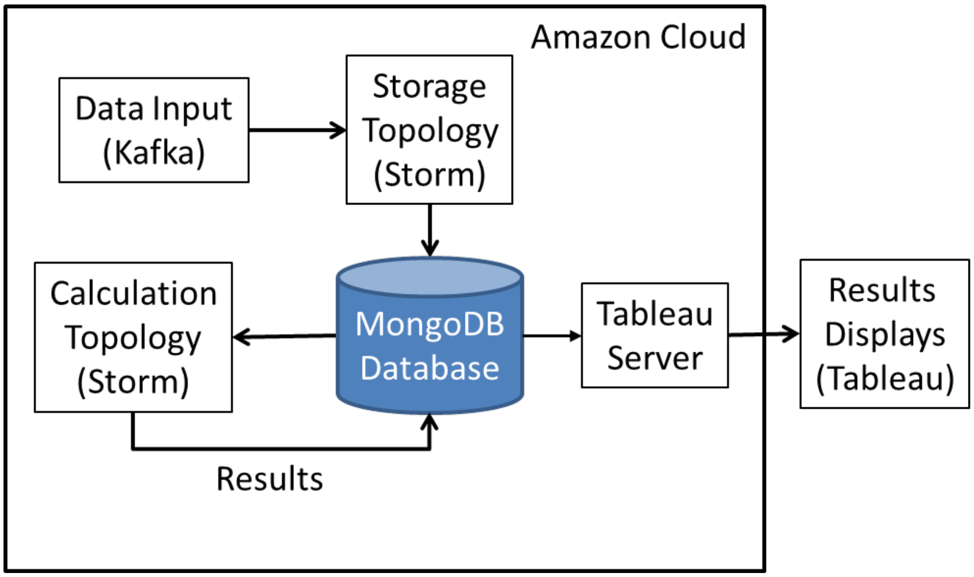


Figure 1. Processing Flow and Architecture

In order to investigate how well our algorithms can detect anomalies in a big data setting, we introduce a number of different error conditions or anomalous sensor behaviors into the simulated sensor recorded data. These anomalies can include noise, irregular data, or other behaviors that have been encountered by real-world sensors. As will be shown, low level anomalies may or may not be reflected in the roll-up dashboard views. Determining meaningful metrics that can capture sensor anomalies, especially from a big data perspective, is a key part of this study.

As a gauge of how well a sensor is performing, the observed data can be compared to high resolution data, such as would be generated by laser-ranging sites. The comparison between the observed data and the "truth" data yields performance metrics, or Essential System Parameters (ESPs). For the purpose of this study, these ESPs are Range, Azimuth, and Elevation variances from the "truth". When the ESPs are computed, the results are limited to known calibration satellites to reflect the stability and known attributes of these objects.

Figure 2 shows a test case for the ESP generation where additional noise and a data bias is added to the day 3 observation data. The figure shows the results using the range variance ESP calculation. The other days show variances based on the Gaussian noise that has been added. As can be seen, even a small amount of additional noise can show marked results and would be readily observed.

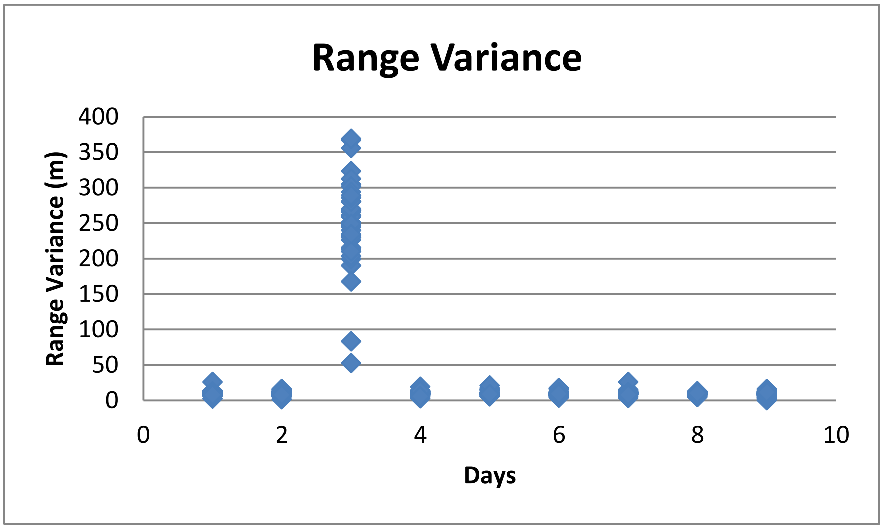


Figure 2. Range Variance ESP

Another top-level view of the data that can be used for diagnosis is how well the sensor is able to collect the data it is tasked to collect. Figure 3 shows an example of the tasking performance for notional tasking categories. For this test case, the RCS of all the satellites was artificially decreased by a factor of 2 between day 5 and day 9 and the number collected/number tasked for each category was calculated.

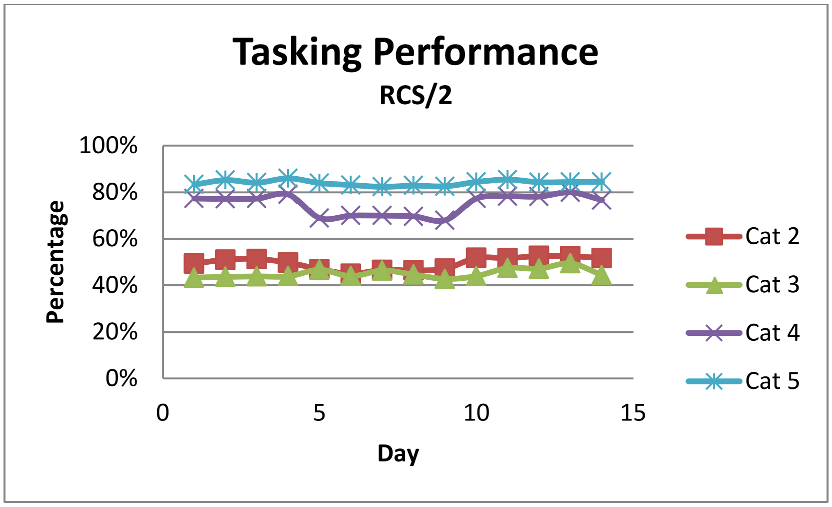
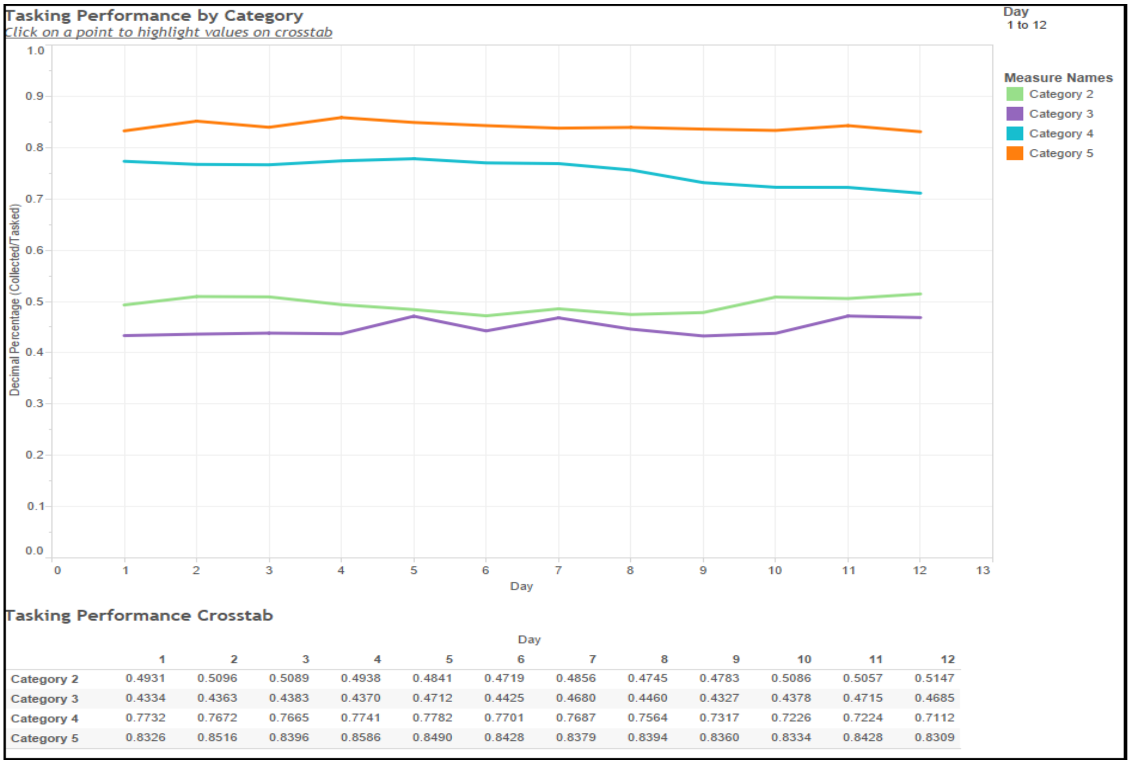


Figure 3. Tasking Performance

As before, it is seen that underlying anomalies in the system can be reflected in high level visualizations. Of note, the effect of creating an anomaly is only readily visible for some tasking categories, in this case Category 4 and to a lesser extent for Category 2 satellites. This high-level view naturally would lead an analyst to want to perform a more detailed analysis of the data. The ability to drill-down to subsequent levels of data detail is a key feature of BDPA. In our Tableau dashboard, as shown in Figure 4, analysts can narrow the day range to inspect deeper trends and highlight points to obtain exact values. This view of the dashboard consists of a set range of 12 days.


Tasking Performance by Category
Click on a point to highlight values on crosstab
Day
1 to 12
Measure Names
🟩 Category 2
🟪 Category 3
🟦 Category 4
🟧 Category 5
Decimal Percentage (Collected/Tasked)
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
3
4
5
6
7
8
9
10
11
12
13
Day
Tasking Performance Crosstab

	Day											
	1	2	3	4	5	6	7	8	9	10	11	12
Category 2	0.4931	0.5096	0.5089	0.4938	0.4841	0.4719	0.4856	0.4745	0.4783	0.5086	0.5057	0.5147
Category 3	0.4334	0.4363	0.4383	0.4370	0.4712	0.4425	0.4680	0.4460	0.4327	0.4378	0.4715	0.4685
Category 4	0.7732	0.7672	0.7665	0.7741	0.7782	0.7701	0.7687	0.7564	0.7317	0.7226	0.7224	0.7112
Category 5	0.8326	0.8516	0.8396	0.8586	0.8490	0.8428	0.8379	0.8394	0.8360	0.8334	0.8428	0.8309

Figure 4. Tableau Dashboard of Tasking Performance

The use of BDPA approaches allows the extraction of trends from sensor data. As a first example scenario, we examine what can be detected if the calibration constant is allowed to vary with time. We shall see that some calculations or data analytic approaches are more conducive to detecting sensor anomalies than others. The calibration constant is the value used to correct the detected radar or optical cross section (RCS/OCS) to a known value based on calibration satellites. It reflects the system performance and any noise or systematic errors within the sensor. As such, it is a good indicator of potential problems in a sensor. As the calibration constant decreases, the probability of detection also decreases. For this scenario, the calibration constant is slowly varied over a 50 day period.

Figure 5 shows the Tasking Performance over the time period. A cursory glance suggests that something is going on based on the decreased tasking performance for the Cat 4 and Cat 5 objects. Understanding the reason for this decrease becomes the next step in determining what the data is revealing.

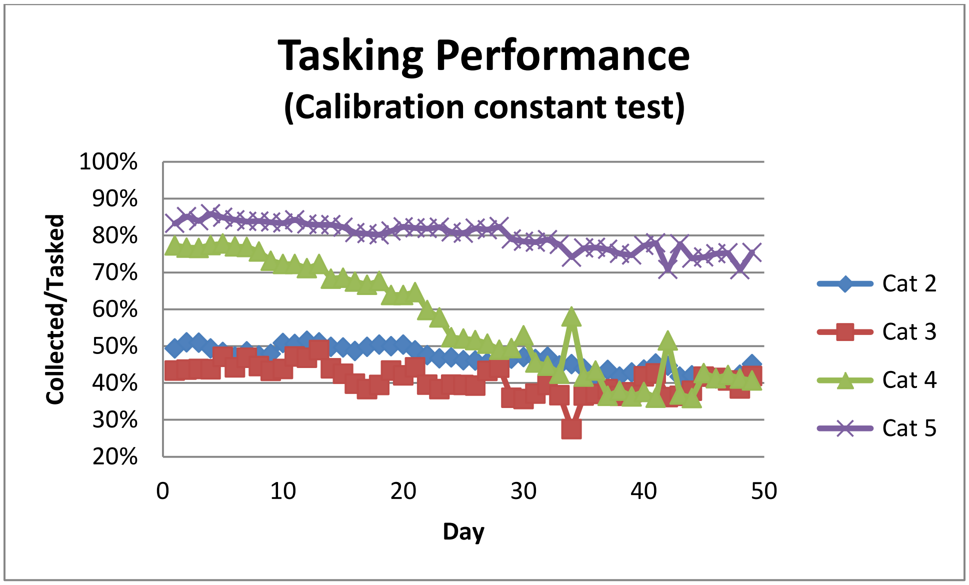


Figure 5. Tasking Performance

Figure 6 shows the range variance ESP calculations. From this view, all the data fall within the bounds of the expected errors. There is little indication that there may be anomalies within the sensor.

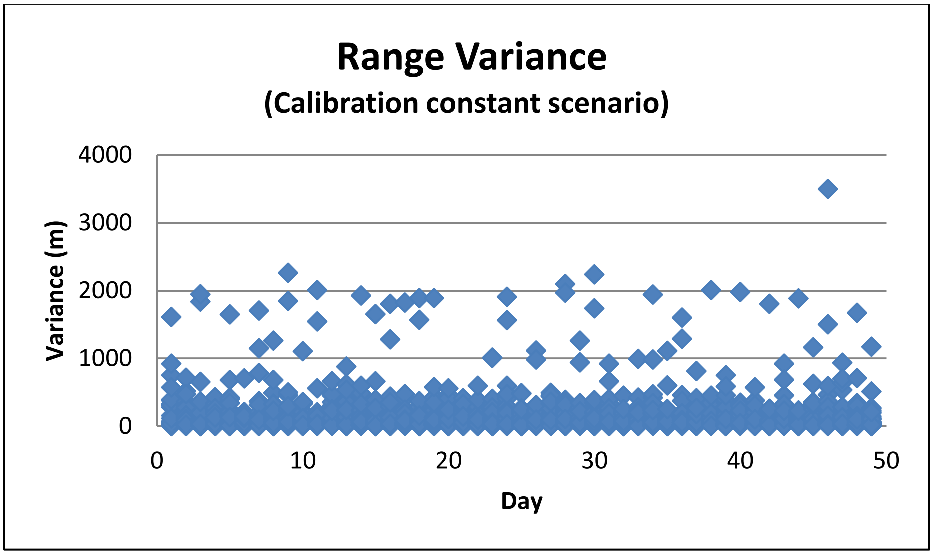


Figure **6**. Range variance for varying calibration constant

To further attempt to understand the decreases in the Tasking Performance shown in Figure 5 we examine the tasking performance in more detail. Figure 7 shows the number of objects collected for each of the tasking categories. The Cat 4 tasking does show a decrease over time, however, from Figure 8 it is seen that there is also a decrease in the number tasked at about day 25.

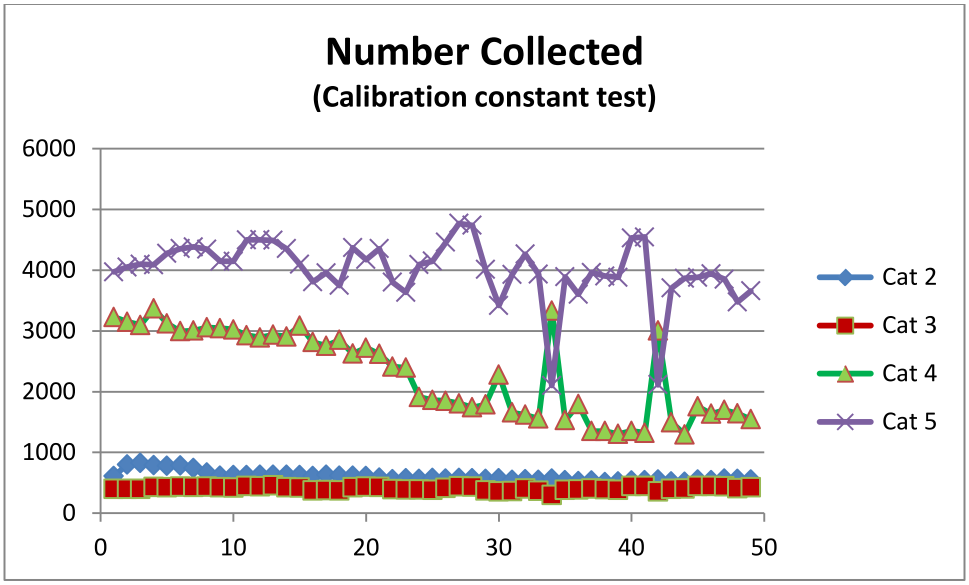


Figure 7. Number collected

Fortunately, the numbers collected versus numbers tasked are consistent over this time period.

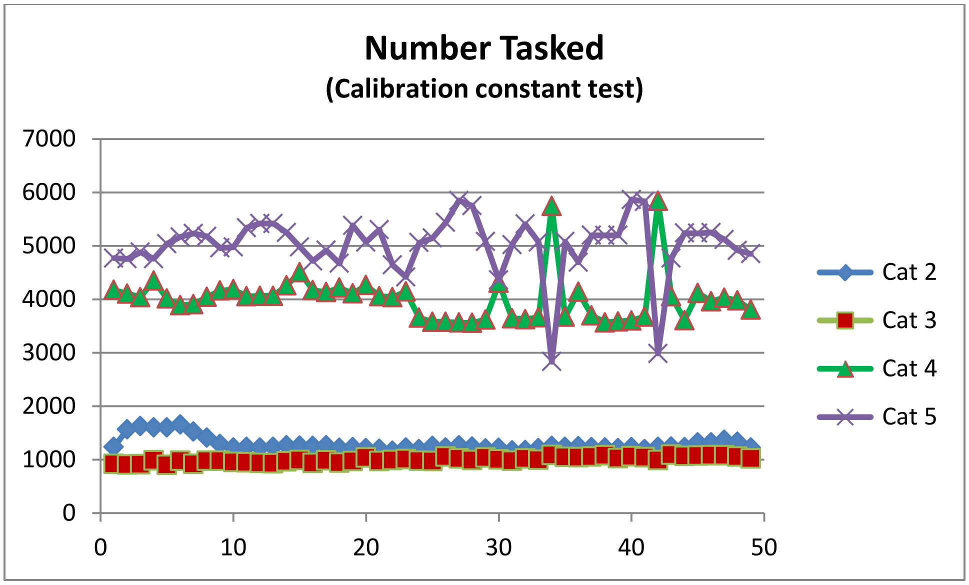
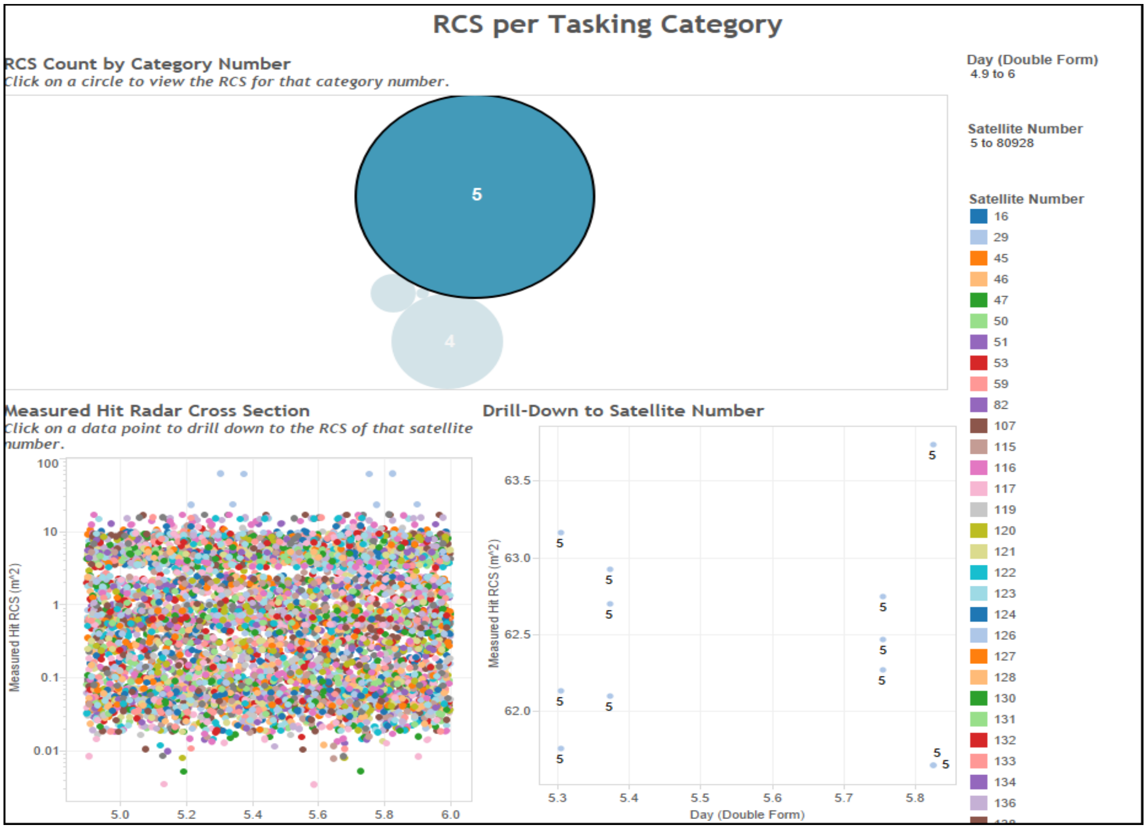


Figure 8. Number tasked

We can also examine the radar cross section for a specific tasking category if needed, such as category 5. The measured RCS values after a very short amount of time can be visualized in Tableau, as shown in Figure 9 . Noticeable outlying values can be attributed to satellite name and number, and drilled-down to view the entire cross section; this is useful once an abnormality is suspected for a particular satellite number.


RCS per Tasking Category

RCS Count by Category Number

Click on a circle to view the RCS for that category number.

Measured Hit Radar Cross Section

Click on a data point to drill down to the RCS of that satellite number.

Drill-Down to Satellite Number

Day (Double Form)
4.9 to 6

Satellite Number
5 to 80928

Figure 9. Tableau RCS Plot

To further understand the reasons for the decrease in the tasking performance, and even to understand the reasons for the spikes in the data, we look at other recorded data. In Figure 10 we look at the elset age (elset epoch time - epoch time for the beginning of the day) for the calibration satellites used in the ESP calculations. From this graph, it is apparent that the spikes in tasking performance are related to cases where the elsets become older (at least in our simulation). This would result in decreased tasking performance. But since it only occurs at specific days, it does not explain the long term trends in the decreased tasking performance.

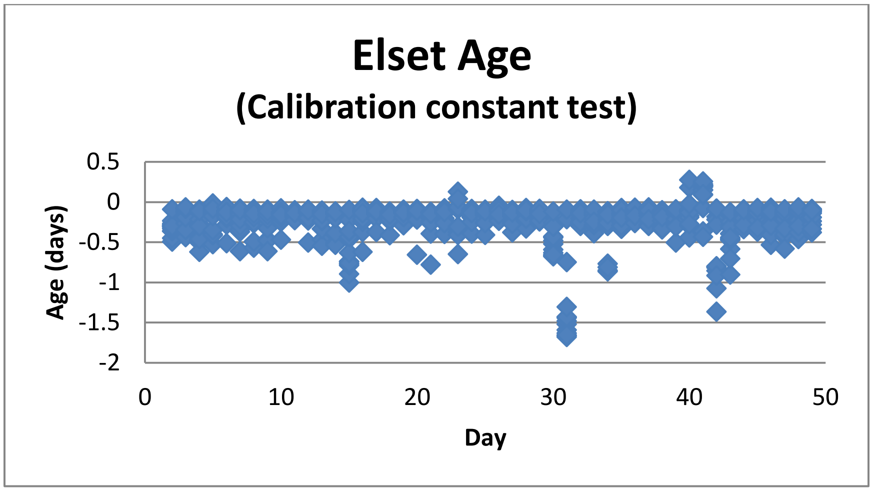


Figure 10. Elset Age

We can also filter the data by satellite number, which can assist analysts by eliminating extraneous variables when coming to conclusions about the root cause of these reasons. In this drill-down example, the elset ages and calibration constants of satellites of interest are displayed for a selected day value, day five.

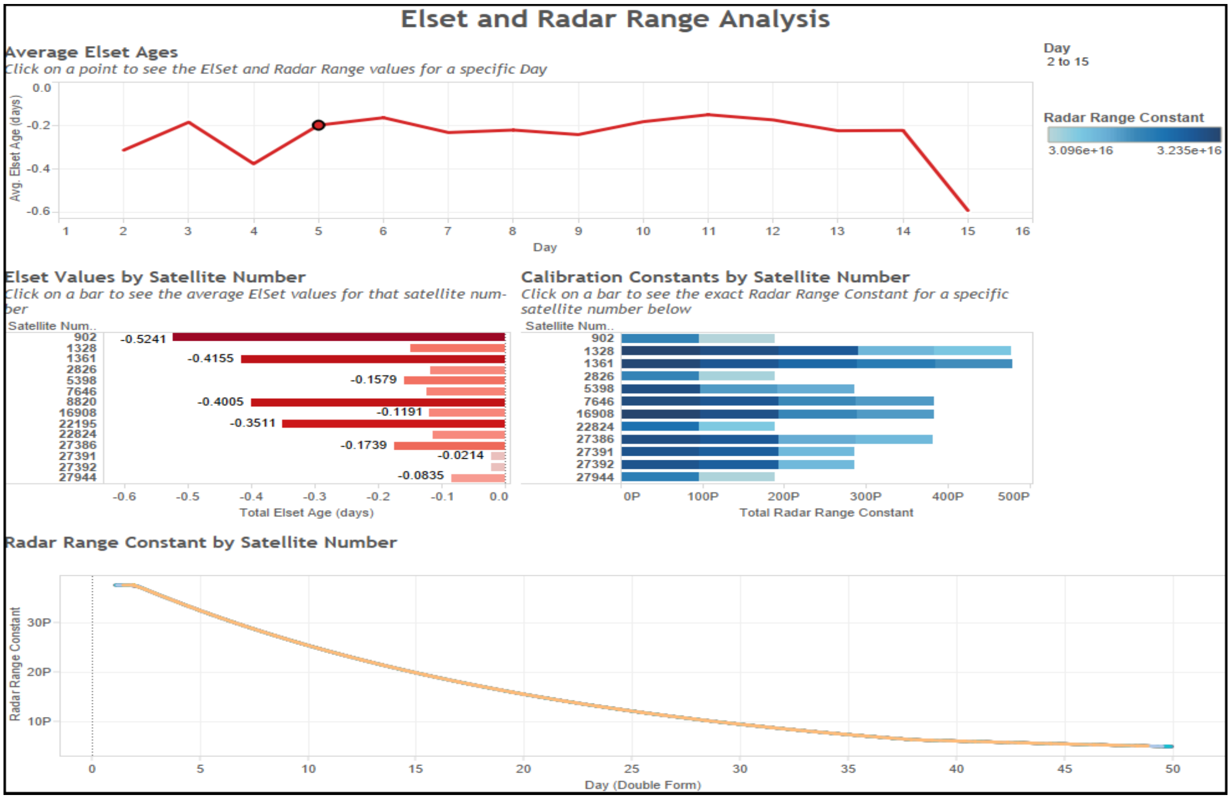


Figure 11. Elset Age Drill-down

Another recorded parameter that may be of interest in evaluating the impact on the tasking performance is the calibration constant. Since there is a direct correlation between the calibration constant and probability of detection analyzing it over time may yield additional insight into the decrease in tasking performance for some satellites. As shown in Figure 12, this does show a decrease over time which corresponds to the decrease in the tasking performance.

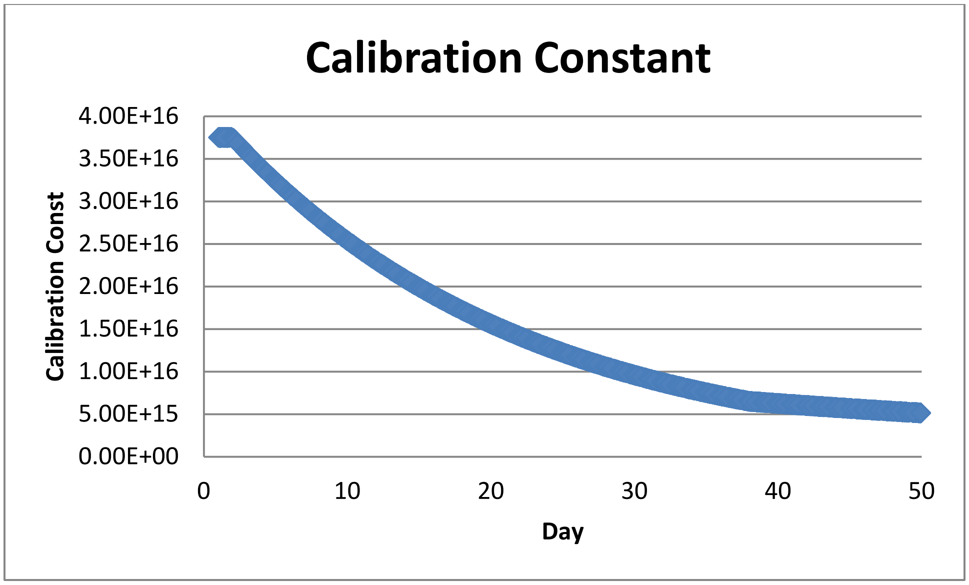


Figure 12. Calibration Constant

From examining the calibration constant over time, we have now come up with the rationale for the decrease in the tasking performance and have identified a preliminary indicator of the cause of the anomalous behavior.

This scenario has shown the importance of being able to drill down into the data to extract the root cause of anomalies. It is also interesting to note that not all indicators, such as the range ESP, provide indicators of anomalies. In this scenario, multiple issues were identified, such as elset age and decreases in the calibration constant, that need to be analyzed to determine the root cause of the problem. From this simple scenario, it is clear that a single solution may not always be the cause of the anomaly. This adds significant complexity in trying to predict the reasons for anomalies.

As the second example scenario, if elsets are not updated, then we use BDPA to determine if this error condition can be readily detected. For this scenario, anomalous data is generated for a 21 day period and the ESPs are calculated.

Figure 13 shows the range variance ESP graph that could be monitored for anomalies. As can be seen, on days 9 and 10 anomalies in the Range ESP can be seen.

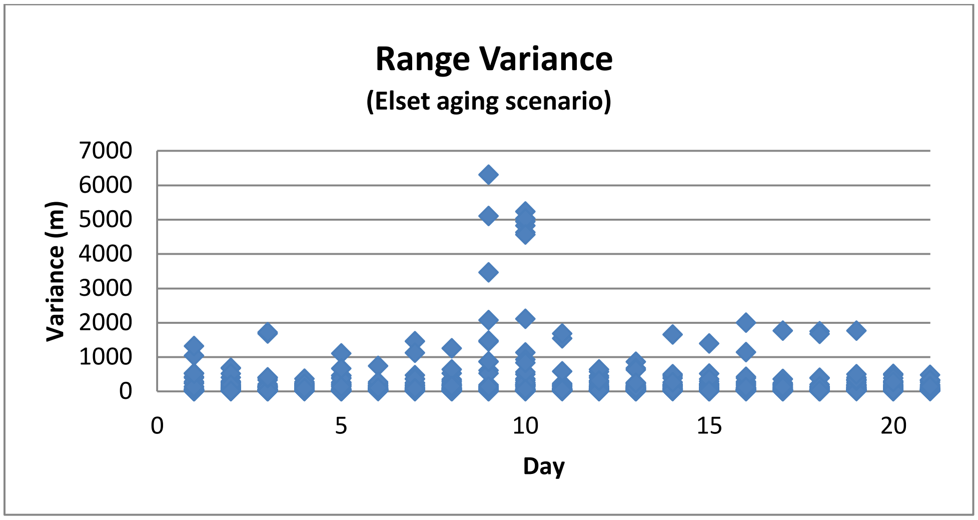


Figure 13. Range Variance for elset aging scenario

Figure 14 shows the elevation variance, such as would be seen from a notional optical sensor. From this figure, anomalies can be seen around day 9, but it is not as apparent as for the range variance. Some ESPs may be stronger indicators of anomalous conditions than others.

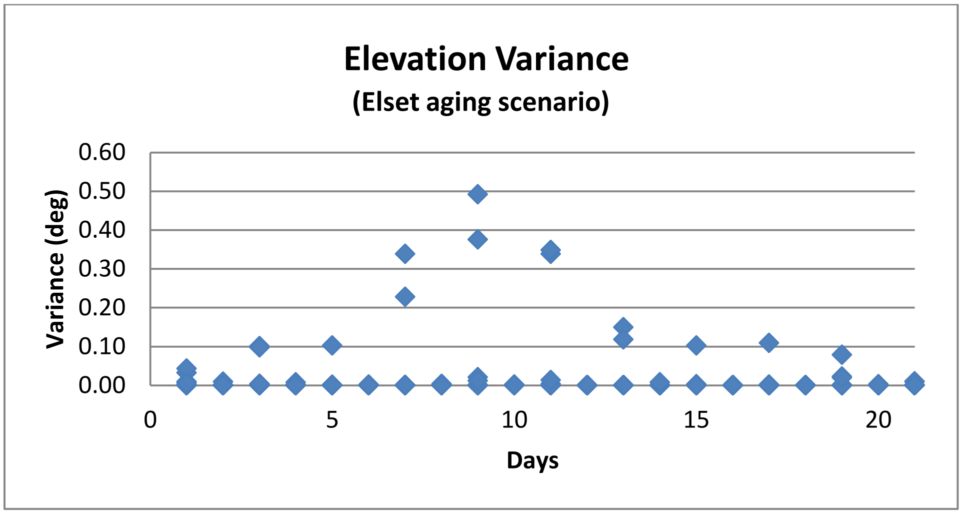


Figure 14. Elevation Variance for Elset aging scenario

Closer examination of some of the anomalies seen in the elevation variance graph suggests that satellites with larger mean motions are responsible for the larger variances.

Figure 15 shows the tasking performance for this scenario. Surprisingly, for Cat 2 and 3 objects for days 9 and 10, the tasking performance appears to increase for the days when the elsets are older.

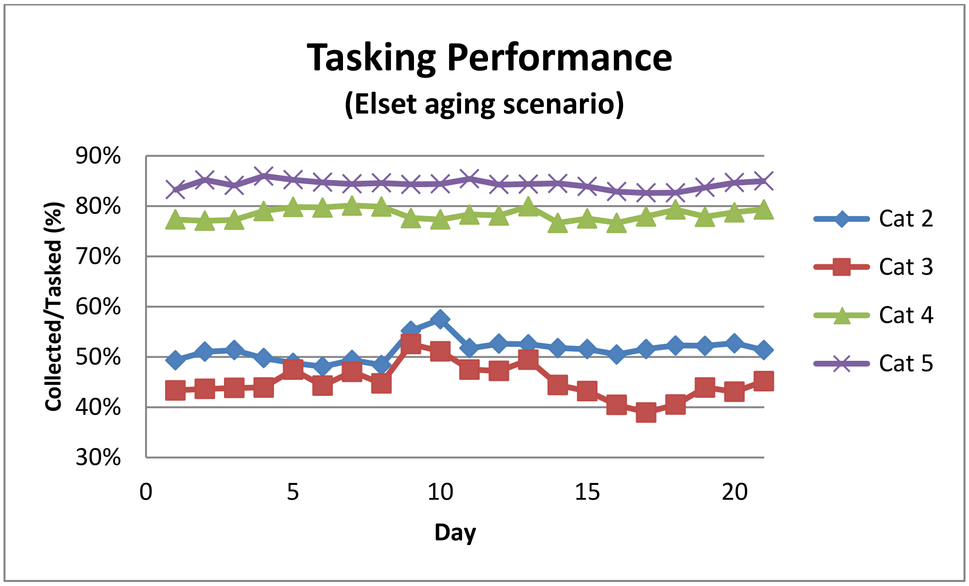


Figure 15. Tasking Performance for elset aging scenario

Although slight anomalies can be seen, the range variance is the only strong indicator of a potential problem in the data. It is necessary to drill-down into the data further to understand the root cause of the anomalies.

Figure 16 shows a computation of the elset age (elset epoch time - epoch time for the beginning of the day) for the calibration satellites used in the ESP calculations. It is clear from this graph that there are problems with the age of the elsets on days 9 and 10.

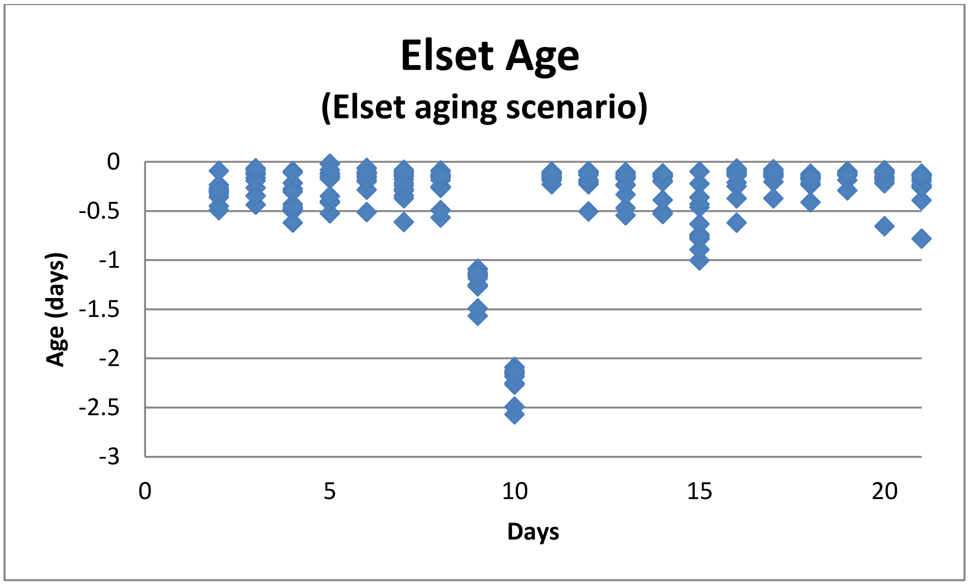


Figure 16. Elset Age

By drilling down into the data, we see that the elset age gives the clue as to the source of the anomaly in the range variance. It is also clear from this scenario that not all calculated parameters provide insight into the sensor anomalies. Some results appear to mask the root cause of the problem. And it is apparent that some results are counter-intuitive and need to be analyzed further to determine the true root cause of sensor anomalies.

3. CONCLUSIONS

This study has shown that there is a utility in using big data analytics to analyze and diagnose possible sensor anomalies. The scenarios analyzed have also shown the importance of being able to drill down into the data to extract the root cause of anomalies. From this analysis it is clear that multiple approaches and multiple indicators need to be examined to assess sensor anomalies.

This study yielded some surprising results. Problems within a sensor can be subtly masked and may not appear in the ESPs or other diagnostics as expected. Examination of the data through multiple mechanisms is therefore critical to a successful data analysis approach.

The next phase of this study will analyze predictive analytics and machine learning approaches toward automation in determining anomalies.

The authors wish to thank SGT for sponsoring this project through their Innovation Technology Center.

4. REFERENCES

The following references are for the software used in our BDPA system:

1. Apache Kafka [Computer Software] http://kafka.apache.org/
2. Apache Storm [Computer Software]. (2011). https://storm.apache.org/
3. MongoDB [Computer Software]. (2007). https://www.mongodb.org/