Implementing Operational Analytics using Big Data Technologies to Detect and Predict Sensor Anomalies

Joseph Coughlin, Rohit Mital,
Shashi Nittur, Benjamin SanNicolas, Christian Wolf, Rinor Jusufi
Stinger Ghaffarian Technologies Inc.
Colorado Springs, CO
joe.coughlin@sgt-inc.com

ABSTRACT

Operational analytics when combined with Big Data technologies and predictive techniques have been shown to be valuable in detecting mission critical sensor anomalies that might be missed by conventional analytical techniques. Our approach helps analysts and leaders make informed and rapid decisions by analyzing large volumes of complex data in near real-time and presenting it in a manner that facilitates decision making. It provides cost savings by being able to alert and predict when sensor degradations pass a critical threshold and impact mission operations.

Operational analytics, which uses Big Data tools and technologies, can process very large data sets containing a variety of data types to uncover hidden patterns, unknown correlations, and other relevant information. When combined with predictive techniques, it provides a mechanism to monitor and visualize these data sets and provide insight into degradations encountered in large sensor systems such as the space surveillance network.

In this study, data from a notional sensor is simulated and we use big data technologies, predictive algorithms and operational analytics to process the data and predict sensor degradations. This study uses data products that would commonly be analyzed at a site. This study builds on a big data architecture that has previously been proven valuable in detecting anomalies.

This paper outlines our methodology of implementing an operational analytic solution through data discovery, learning and training of data modeling and predictive techniques, and deployment. Through this methodology, we implement a functional architecture focused on exploring available big data sets and determine practical analytic, visualization, and predictive technologies.

APPROACH

This study developed an operational analytics implementation that uses Big Data technologies and machine learning algorithms to determine and predict sensor anomalies. A previous study [1] showed that Big Data Analytics can uncover anomalies that may be missed through conventional analyses. This study enhances that effort and shows a methodology to implement operational analytics that can be applied toward common solutions for data analysis. Our operational analytics implementation relies on continuous learning from historical data to analyze data in the stream of real-time operations. In the previous study, where data was identified that can be used to uncover anomalies, this implementation extends that approach and now identifies trends and correlations that reveal anomalies that can be missed by traditional analytic techniques with limited datasets. This study adopted a three-step methodology to implementing operational analytics – Discovery, Modeling and Operations as shown in Fig. 1.

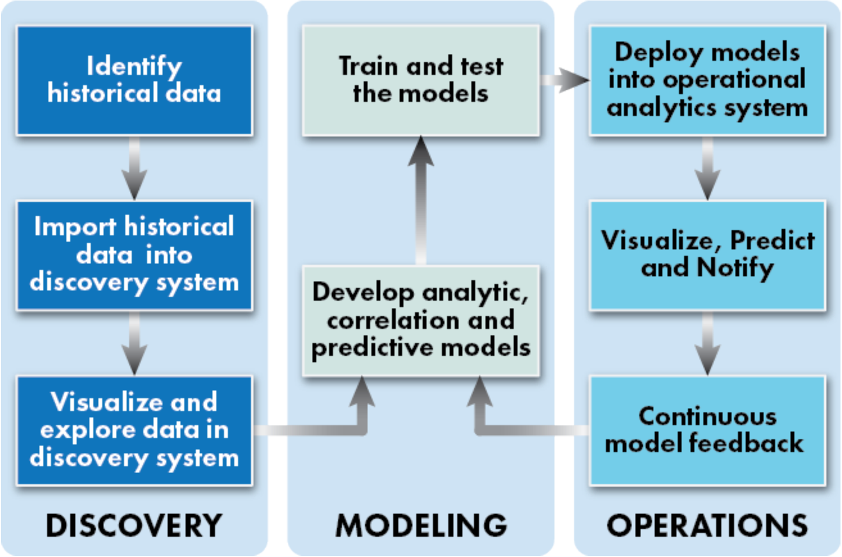

Identify historical data
Import historical data into discovery system
Visualize and explore data in discovery system
Train and test the models
Develop analytic, correlation and predictive models
Deploy models into operational analytics system
Visualize, Predict and Notify
Continuous model feedback
DISCOVERY
MODELING
OPERATIONS

Fig. 1. Operational Implementation Approach

Fig. 1 shows the three steps to implement operational analytics and the continuous feedback between learning and operational deployment. The following sections will elaborate on the methodology employed as applied to a real-world problem of analyzing large datasets such as would be encountered at an operational site.

PHASE 1: DISCOVERY.

The discovery phase uses data exploration and visualization to understand historical behavior of the target datasets. This phase attempts to understand the impacts of one dataset on another so that relationships can be determined and an analysis approach can be implemented.

The study simulated data similar to that captured by a radar site as it tracks satellites. Various anomalies introduced into this simulated data created distinct datasets for discrete insights into the effects of each variable. Although the simulated data includes many types of data that would be recorded at a real site, the discovery phase focused on the subset of the data that was identified in the previous study as being important for uncovering and diagnosing anomalies.

The types of data that were analyzed in detail are:

- Radar range constant (RRC). A calibration constant used to correct for errors in the radar
- Radar cross section (RCS). The area of a satellite as seen from the Earth at a specific time, a value that can change over time depending on the shape and rotation of the satellite
- Elset age. The time from the elset epoch time
- Tasking performance. The percentage of satellites tasked that were actually tracked

The first three fields were chosen based on the possibility of them impacting the performance of the radar, while the fourth field is a direct calculation of the performance of the radar.

PROCESSING ARCHITECTURE

Amazon Web Services (AWS) Cloud Computing Platform was used for this project. The data was generated and stored on AWS Simple Storage Service (S3). AWS provides the ability to easily scale up and scale down to match the velocity and volume of data.

The data was both visualized using a sophisticated big data visualization stack and further processed. The visualization software used consisted of three tools: ElasticSearch, Logstash, and Kibana (ELK) [3]. The tools that comprise ELK are used to search, capture, and visualize the data, respectively. To process the data, Apache Spark [2] was used, which is a fast and general engine for large-scale distributed data processing.

The tools that make up the ELK stack are integrated to enable horizontal scalability, making ELK a great choice for big data analytics. ELK was configured on a cluster setup. Due to ELK's inherent cluster support, this was easy to accomplish. The cluster consisted of three m4.xlarge instances for Elastic Search, one m4.xlarge instance for Log stash, and just a t2.small for Kibana, as shown in Fig. 2.

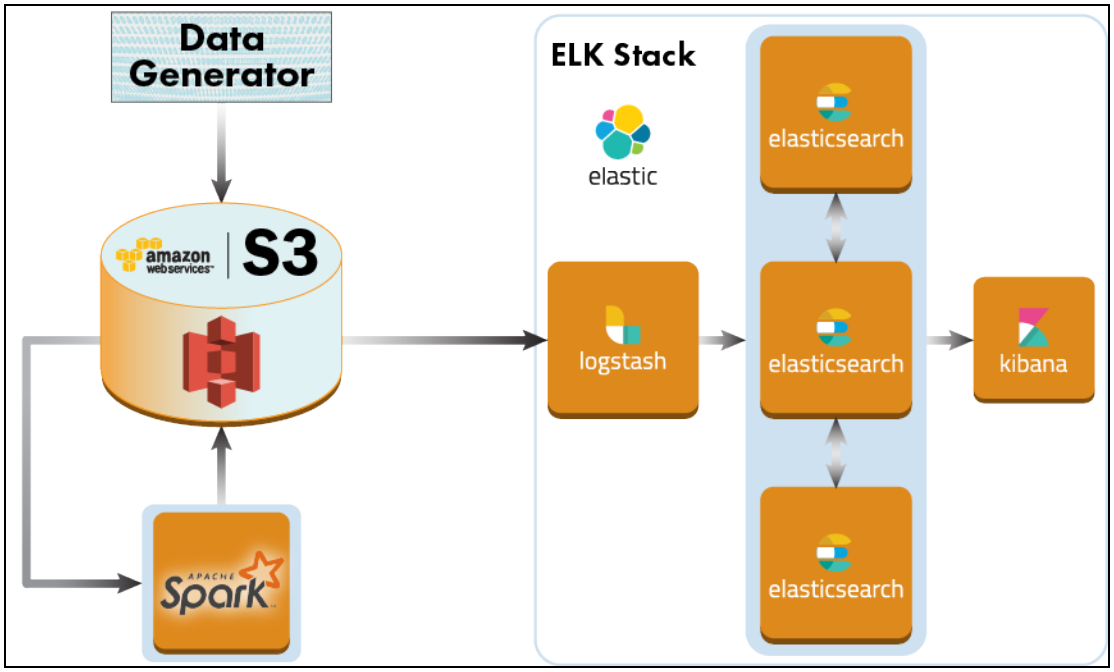


Fig. 2. Data Visualization and Processing Architecture

In the previous study [1], Tableau was used as the main visualization software. We decided to move away from that product in favor of the ELK stack for several reasons. Tableau seems best suited for historical data, which can be helpful in initial development, but the goal of this project is to look at real-time streaming data — something ELK is much better at. ELK also offers a much more complete package than Tableau. Tableau's functionality is comparable to that of Kibana, but ELK also includes ElasticSearch and Logstash. Logstash is used to ingest data and ElasticSearch stores this data, but when using Tableau other services must be integrated for similar capabilities.

The performance of the radar, defined by tasking performance, was the focus of the analysis. The goal was to be able to tell which of the variables affect the performance and in what ways. To do this, the shape of the data needs to be analyzed. To get an initial idea of what the data looked like, the data were visualized using Kibana on both the raw and processed data. Correlations in the data could then be determined and impacts on the tasking performance could be seen. Fig. 3 shows the tasking performance as a function of time for four tasking categories, which are the priorities at which the satellites are collected.

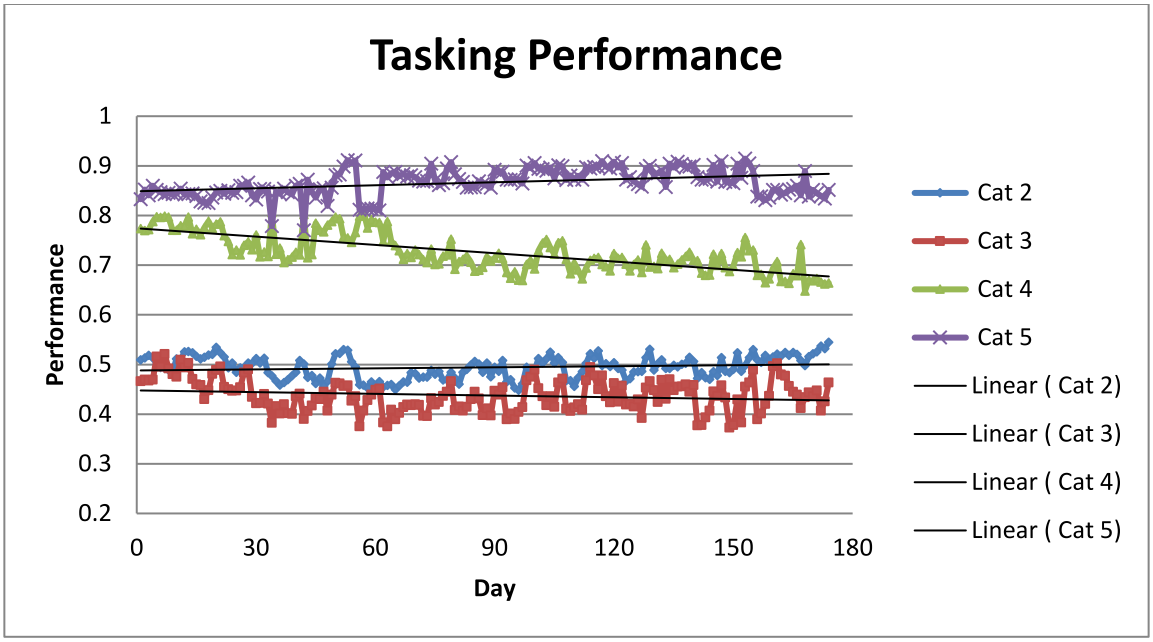


Fig. 3. Tasking Performance

The tasking category 4 satellites show a decrease in the tasking performance in the simulation. There is also a slight upward trend in the category 5 satellites while category 2 and 3 are relatively flat. Much of the remaining analysis is designed to understand the reasons for the trends and anomalies seen in these data. A cursory inspection of the tasking performance shows trending and variability that must be understood in order to assess if there is a true impact to the radar performance or if the observed data is readily explained. This study will determine if there is a true degradation and if we can predict how and why the system is not performing as expected.

During the discovery phase, the objective is to examine the variables that impact the data. One such variable is the RRC, as shown in Fig. 4. This variable impacts the sensitivity of the radar and influences the probability of detection of satellites.

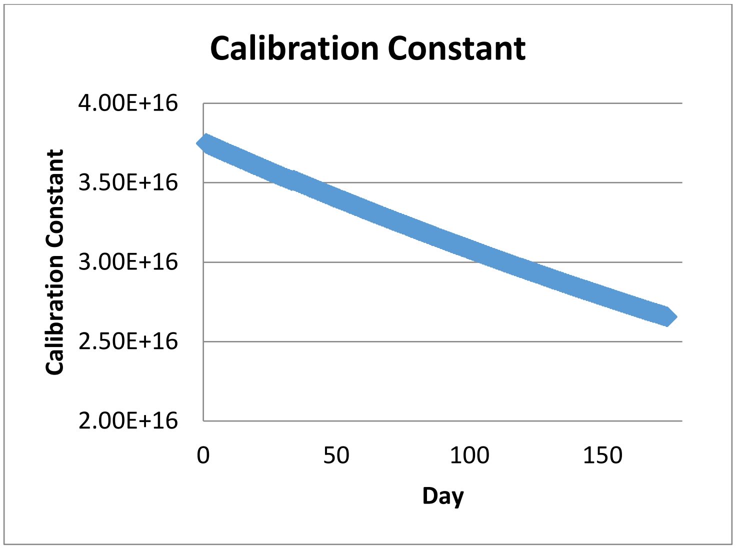


Fig. 4. Radar Calibration Constant (notional)

As it decreases, the radar is less able to detect smaller cross section objects. In this simulation, the RRC is decreased to understand the impact and to reproduce a degradation in the system to understand how to visualize similar radar degradations. Also, there is a need to understand long term trends and short term anomalies in the data and model them appropriately.

Since tasking performance is a gauge of how well the system is performing, by correlating tasking with other fields such as RRC, RCS, and elset age, we can attempt to explain the changes in tasking performance. By looking at the tasking performance for tasking category 4 against RRC, as shown in Fig. 5, as the RRC decreases, so does tasking performance. The correlation between the two parameters is relatively high at about 0.6. With this correlation it can be deduced that the decrease in RRC caused the radar's tasking performance to decrease for the category 4 satellites, as seen in Fig. 3.

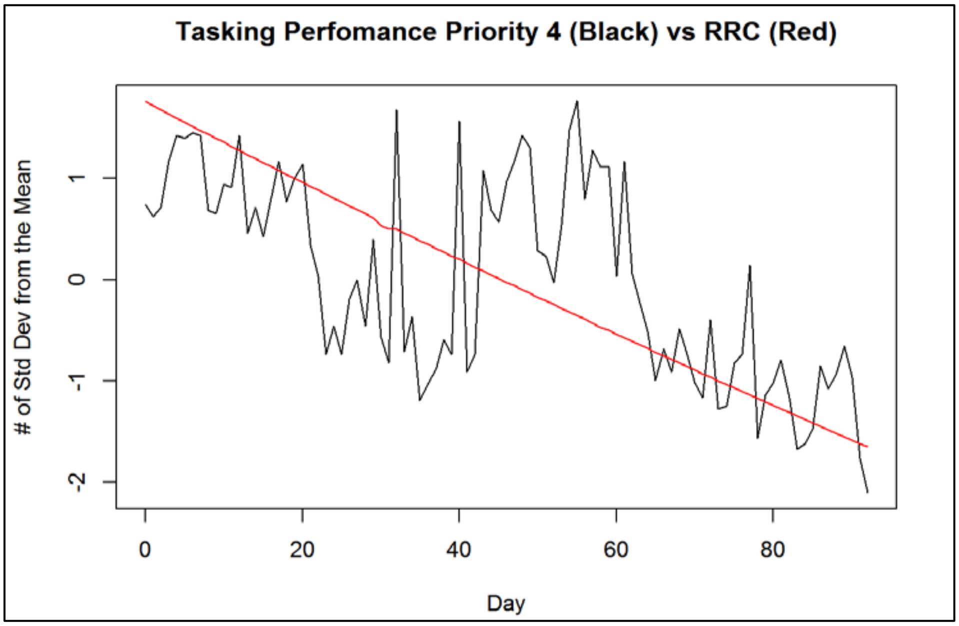


Fig. 5. Tasking vs RRC correlation

Fig. 6 shows tasking performance against RRC as a scatter plot for a different view of correlation between the two variables. There is a strong positive correlation between the fields. Therefore, an increase an RRC typically means an increase in tasking performance.

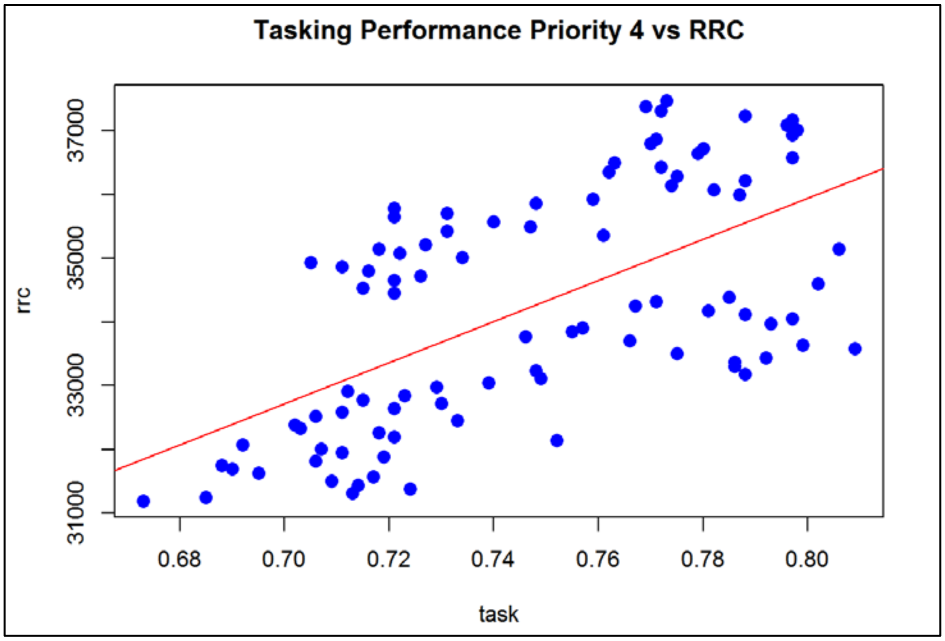


Fig. 6. Tasking Performance vs RRC

Fig. 7 shows the correlations between the three variables identified as having an impact on the tasking performance against other variables. As can be seen, the highest correlation is between the RRC and the Tasking Performance.

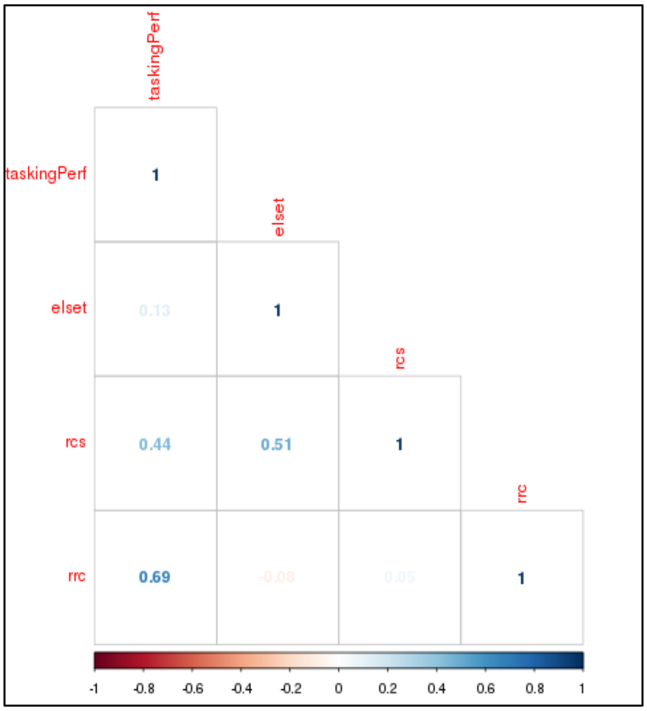


Fig. 7. Correlation matrix for the 3 primary variables against tasking performance

By comparing plots of tasking performance against RCS, RRC, and Elset age, as shown in Fig. 8, along with the correlations of tasking performance and these factors, it can be concluded that RRC has the most influence in this case and is the primary cause of decreased tasking performance. It can be seen that RCS (average RCS over the tasking category) also influences the drop in tasking while elset age had little or no influence. These results confirm the results in the previous study.

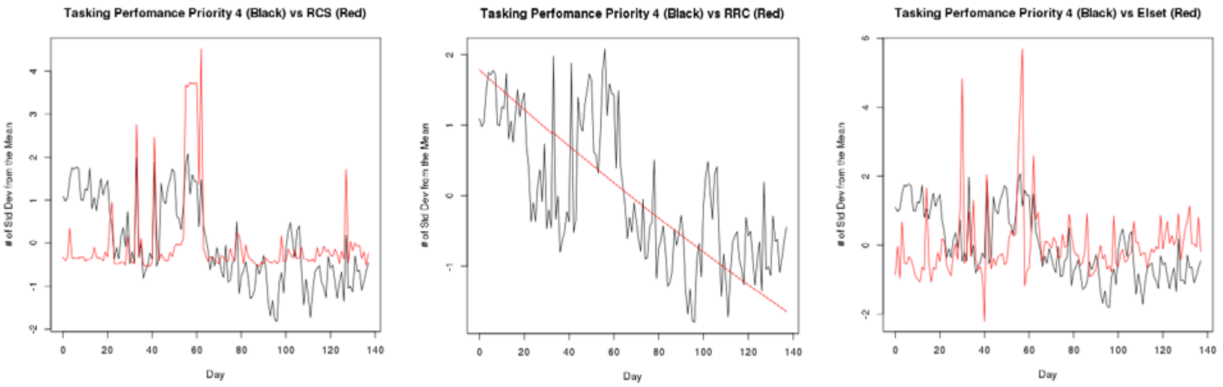


Fig. 8. Tasking Performance vs RCS, RRC, and Elset age

It is important to remember that these correlations are based on the long term trending of the data and do not reflect short term variations. Short term variations or anomalies must be explained through other mechanisms besides such simple correlations.

During discovery, the reasons for short-term anomalies must also be explored. Around day 60, in Fig. 3, there is a deviation in the tasking performance. Analyzing the age of the elsets, as shown in Fig. 9, provides the insight that old elsets which may be more difficult to track could be a cause of this anomaly and cause changes in the tasking performance in the simulation.

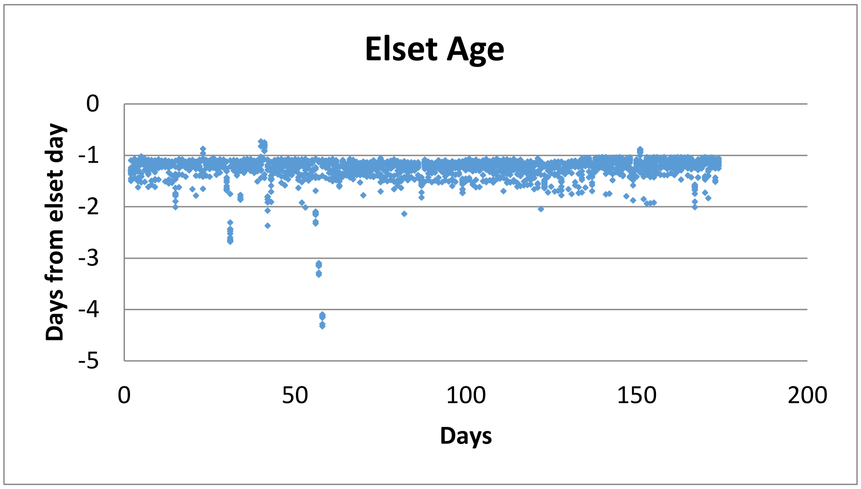


Fig. 9. Elset age

Another source of anomalies or changes in the tasking performance can be the RCS of the satellites that are being tasked for collection. Fig. 10 shows the distribution of RCS and how it changes as a function of time. The shift on day 58 to larger objects could account for the changes in the tasking performance since the larger objects would have a greater probability of detection.

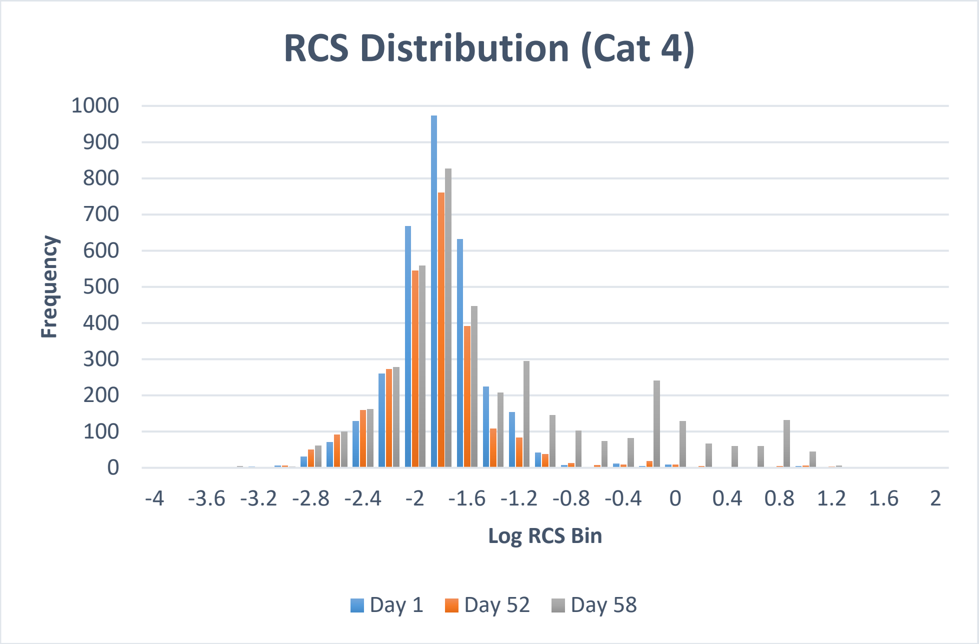


Fig. 10. RCS distribution for tasking category 4 satellites

Fig. 11 shows the RCS distribution and the changes as a function of time for two selected days of interest. As for the Category 4 satellites, changes in what is tasked for collection will influence the tasking performance.

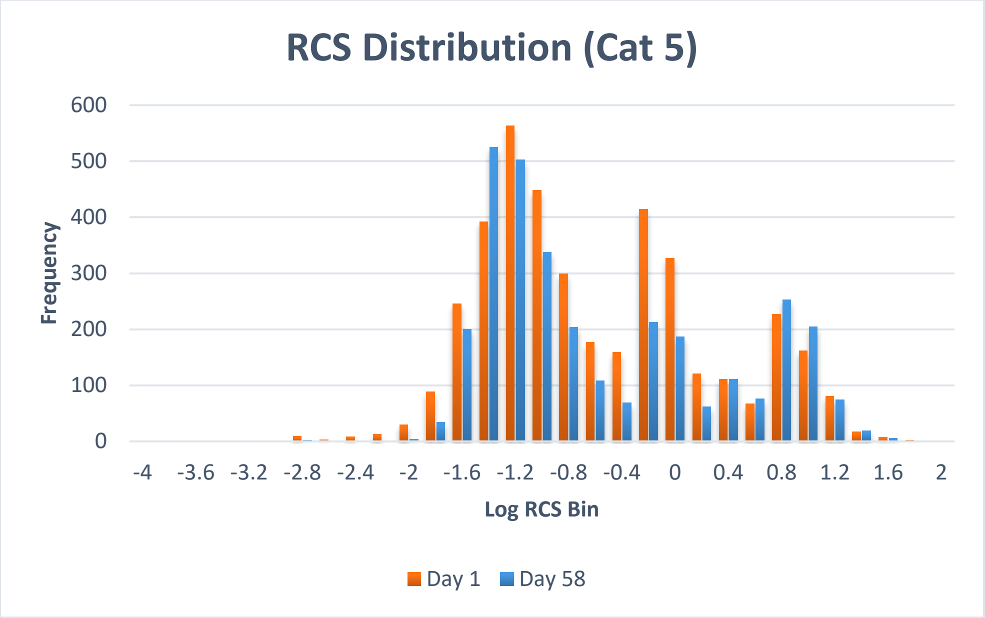


Fig. 11. RCS Distribution for tasking category 5

Both Fig. 10 and Fig. 11 show normal variations in the satellites selected for tasking. This normal variation must be accounted for in determining if the sensor is experiencing anomalies or if the deviations are due to expected changes. The change in distribution could also impact long term trending if the change is more gradual in time. The shift in RCS distribution toward larger satellites could explain the increase in tasking performance as the larger satellites would be easier to track.

PHASE 2: MODELING

The modeling phase extracted and expanded upon key insights observed in the discovery phase. Modeling for operational analytics is not a simulation seeking to recreate an external process, but rather a formalized set of insights gleaned from existing data and used to make accurate predictions about future data. These insights were used to automate and formalize the laborious process of data analytics. The culmination of this phase is the development of a model that can be used by operations to perform root cause analyses and perform predictive analytics.

The two mathematical constructions used to build the model were the Auto Regressive Integrated Moving Average (ARIMA) model along with correlations and regressions.

The ARIMA model is specific to time-series data and produces a forecast for the future values of a given input series. It predicates itself on the idea that future data will resemble past data. Working from this assumption, the ARIMA model extracts the overall trend, as well as periodic behavior or cyclicality, from the data. The ARIMA model is composed of an autoregressive model and a moving average model, along with a differencing parameter to establish stationarity. Every ARIMA model therefore has three parameters, commonly referred to as p, d, and q, which define the behavior of the time series. The p parameter is the order of the autoregressive model, or the number of time steps back from a given time step such that the linear correlation between the p previous time steps and the given time step is maximized across the entire time series. The q parameter defines the order of the moving average model, or the number of time steps back from a given time step such that the correlation between the average of the q previous time steps and the given time step is maximized across the entire series. Finally the d parameter indicates the order of differencing necessary to achieve a stationary mean, which is required for the remaining ARMA components to work correctly. Based on the manual analytics done in the discovery phase, which revealed positive correlations between tasking performance and each of radar range constant, radar cross section, and elset age, we replicated these correlations at each time step in our data. Based on the strength of each correlation, the model determines the mostly likely cause of day-to-day variation in tasking performance.

As the model explains current tasking performance, it also aims to predict future performance. By using an ARIMA model, the model generates a forecast at 80% and 95% confidence intervals. As new data enters the model’s pipeline, it continually recalculates, updating its forecast to account for the additional information. The model uses R’s built-in auto-ARIMA function, which determines the best possible ARIMA parameters. This function calculates the Bayesian Information Criterion and the Akaike Information Criterion for different parameter values, and then outputs the parameter combination that extracts the most “information” from patterns in current data for use in future forecasting.

The model was created using R and SparkR as shown in the architecture diagram in Fig. 12. Data from S3 is continuously pulled into R for three modeling computations.

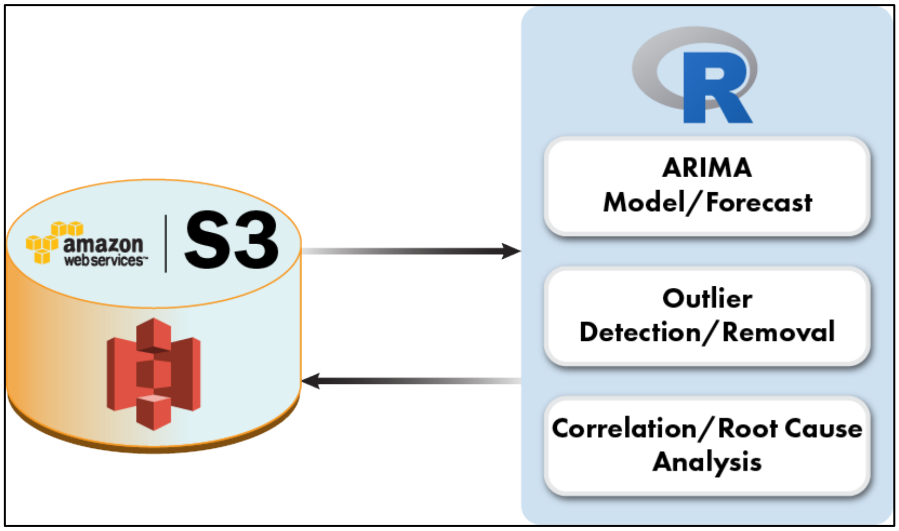


Fig. 12. Modeling Architecture

To test the model, the historical data acquired during the discovery phase was broken up into training data and testing data. 60% of the historical data made up the training data, the remaining 40% of historical data was testing data. We used the training data to configure or train the model. The testing data was then used to test the validity of the models.

Using ARIMA models, the system predicts future tasking performance over the next 14 days. To get the most accurate ARIMA models, outliers had to be removed due to the negative impact of rapid variations of the data. The "tsoutliers" packages in R which is able to detect additive, level shift, and temporary outliers was used. An additive outlier is a very large/small value that occurs for one observation and it does not affect subsequent observations. Level shift outliers shift the level of all the data up or down for subsequent observations. Temporary outliers also shift the level of the data up or down but they slowly decrease in influence until the data comes back to its normal form.

Fig. 13 and Fig. 14 show ARIMA predictions for 14 days for the tasking performance for tasking category 4, with and without outliers. As can be seen in this case, the outliers strongly influence the projection. The projection with the outliers removed shows the trend that would be expected for the decreasing tasking performance.

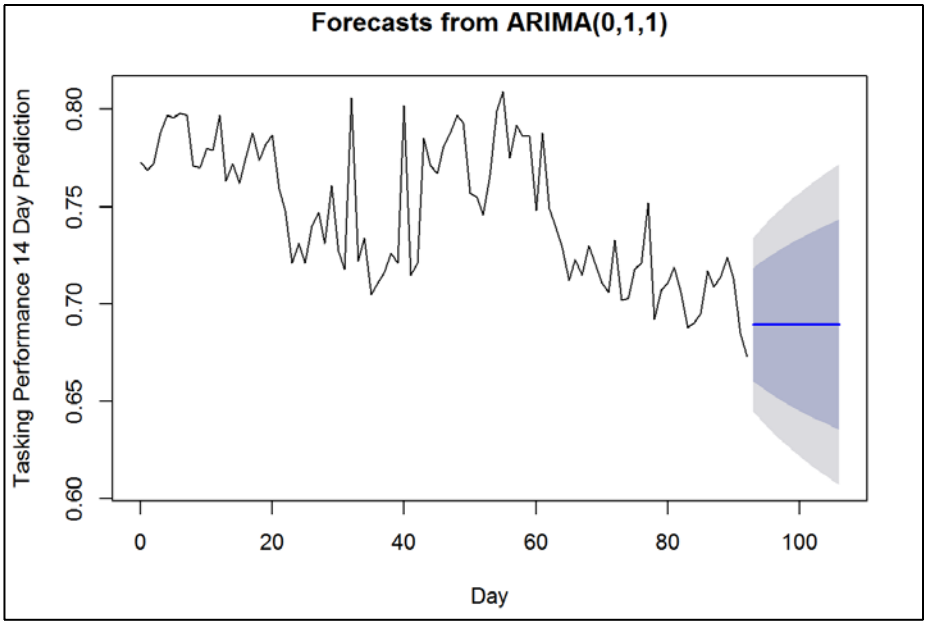


Fig. 13. Tasking Performance ARIMA forecast for Category 4 (outliers included).

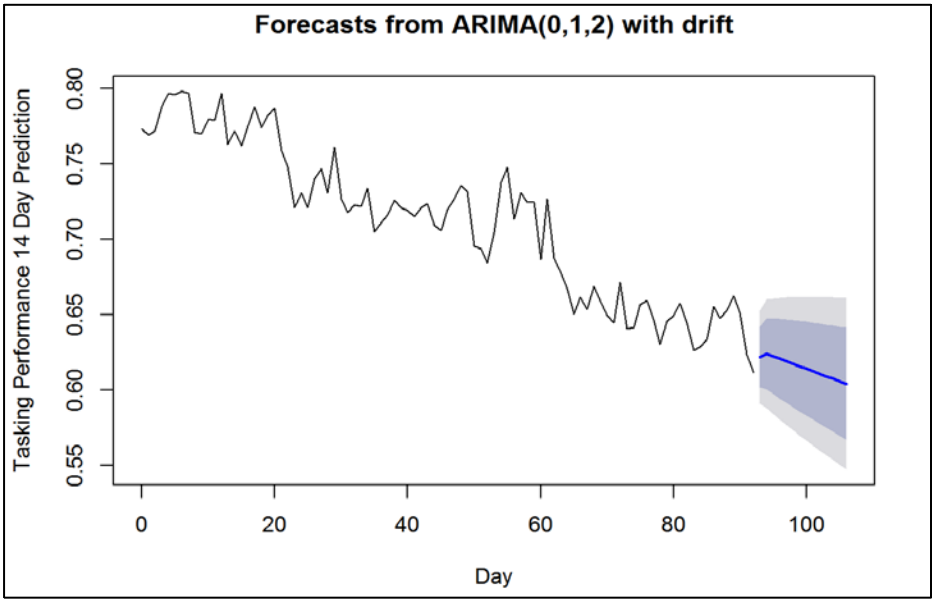


Fig. 14. Tasking Performance ARIMA forecast for Category 4 (outliers removed).

Fig. 15 and Fig. 14 show ARIMA predictions for the tasking performance for tasking category 5, with and without outliers. Since this is a relatively flat graph with minimal trending, the impact of the outliers is not as significant. The projections do not show the slight upward trending in the Category 5 data as could be inferred from the graph in Fig. 3.

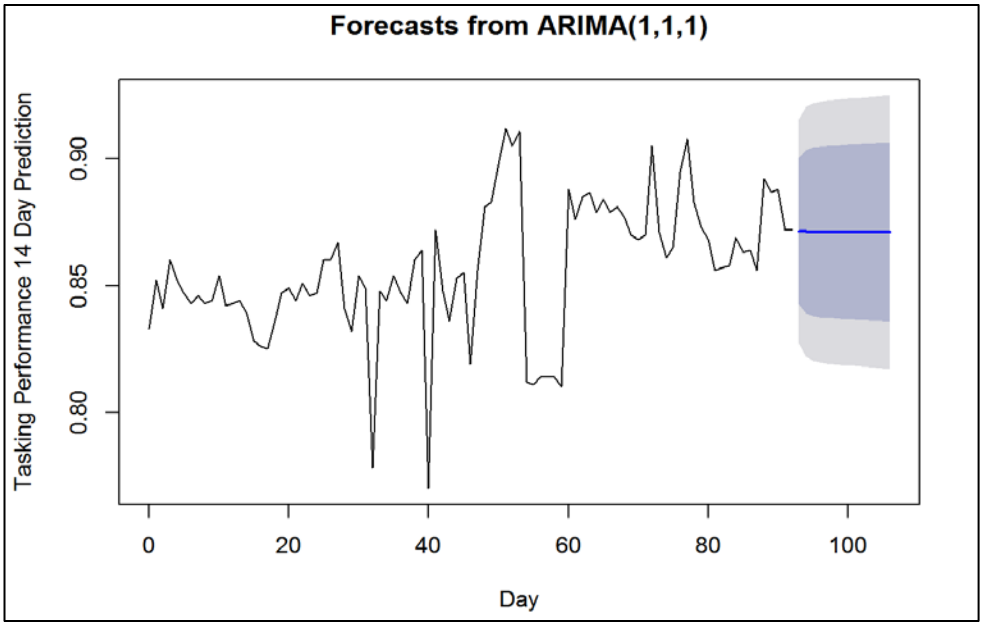


Fig. 15. Tasking Performance ARIMA forecast for Category 5 (outliers included).

The projection in Fig. 16 suggests a short decrease followed by a longer flat period. As can be seen, caution must be used if the operator relies on the short term prediction as it is influenced by the removal of the outliers.

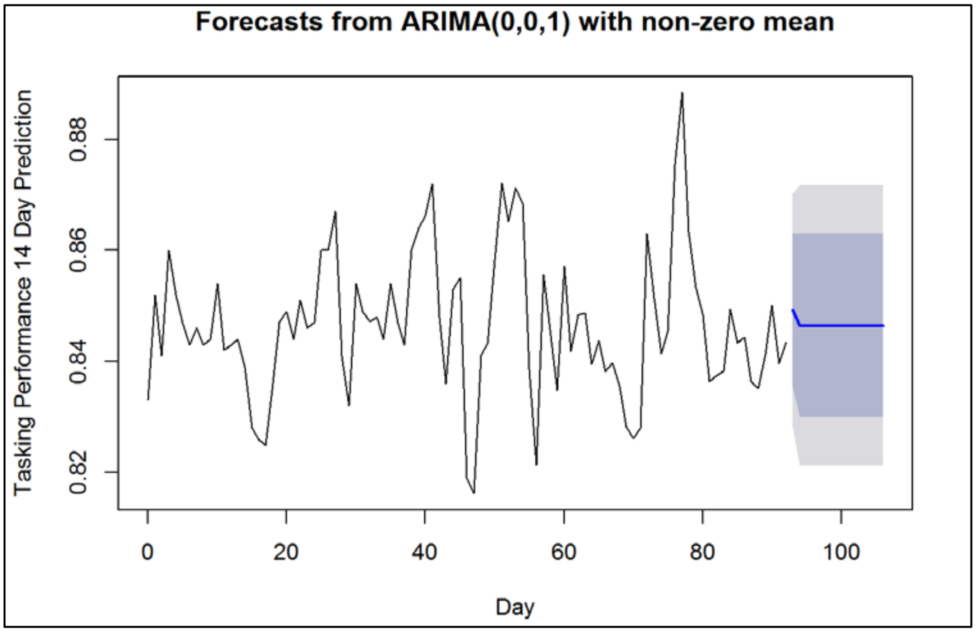


Fig. 16. Tasking Performance ARIMA forecast for Category 5 (outliers removed).

In order to make accurate predictions on the trending, in general it appears that outliers should be removed. Understanding the outliers and their impact on the modeling is a future goal. In the discovery phase it was observed that many of the outliers are due to short term problems with elset age.

Since the tasking can change over the course of time as different satellites gain priority in tasking, the RCS distribution can also change. Although this is expected behavior, any modeling effort must accommodate this factor as it can also impact tasking performance.

PHASE 3: OPERATIONS

A key element of building an operational system is to determine what data should be calculated and presented to the operators/analysts for evaluation. To simulate the desired visualization and analysis, an operational architecture was developed that includes an analysis engine and a dashboard for visualization. For the operational architecture, as shown in Fig. 17, pre-computed data were simulated to run in real time by a Python script that synchronized data indexing in the ELK stack and data analysis in R. In this case R was capable of handling the analysis required but we have also shown that for larger data sets and more analyses, SparkR should be used to analyze the data. Once indexed and analyzed, the data is displayed on an Apache Web Server through Kibana visualizations and R graphs.

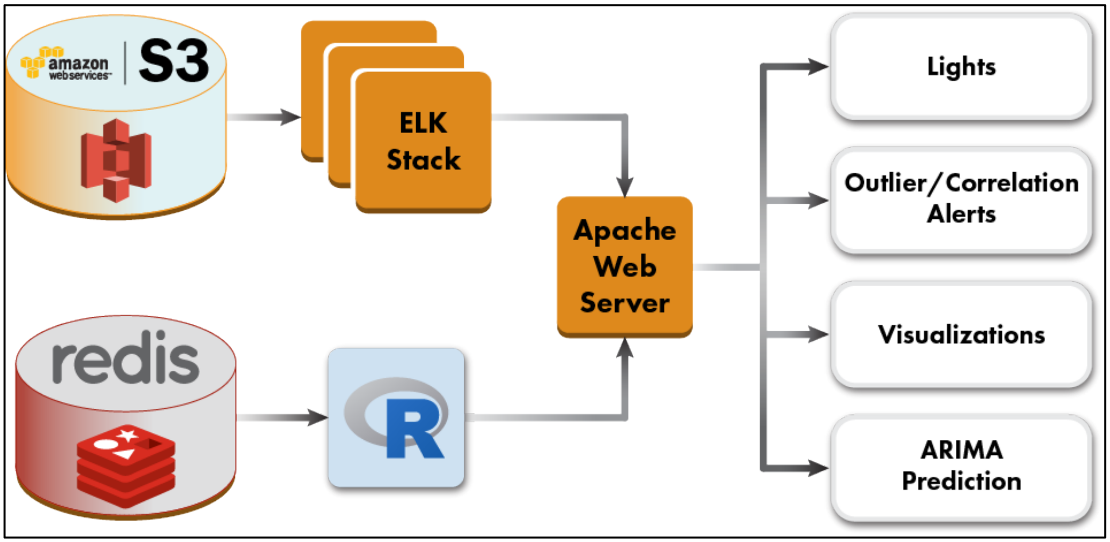
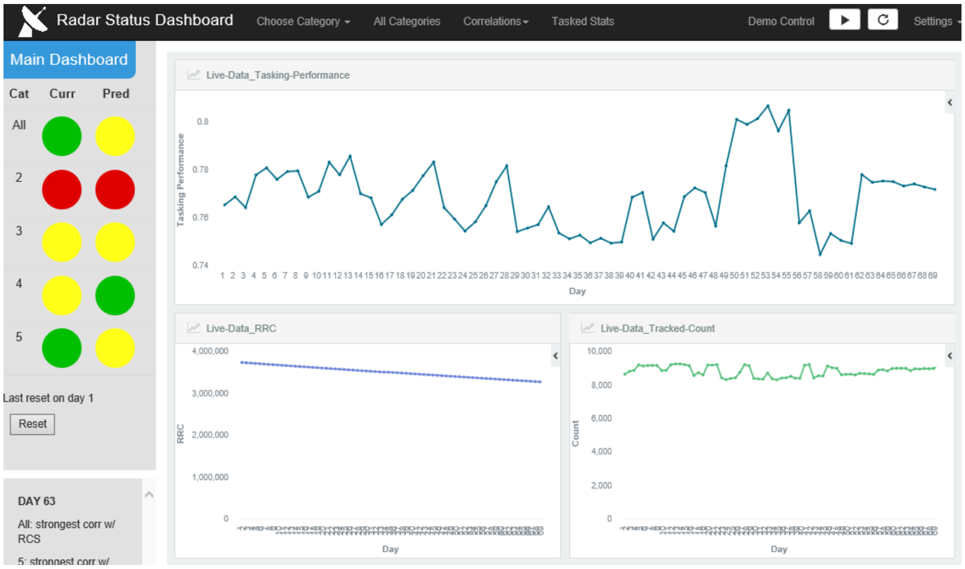


Fig. 17. Operational architecture

A dashboard containing key performance indicators and statistics, along with predicted outcomes is a key part of making the operational analytics relevant for use. A conceptual dashboard, shown in Fig. 18, provides a number of services such as the following:

- Tasking performance lights which alert the operator to drops in the performance
- Outlier and high correlation alerts inform the operator of data requiring review
- Predictions of tasking performance show the operator possible future performance trends
- Visualizations with drill down capability provide the ability to drill down into the data to facilitate further analyses.



Cat	Curr	Pred
All	🟢	🟡
2	🔴	🔴
3	🟡	🟡
4	🟡	🟢
5	🟢	🟡

Last reset on day 1

DAY 63

All: strongest corr w/ RCS

5: strongest corr w/

Fig. 18. Conceptual dashboard provides high level views with ability to drill down into specific data elements

This operational dashboard updates as the real-time data are fed into the system. Based on the ARIMA predictions, the lights on the left hand side will display the projected status of the system. For this case, the lights represent projected decreases in tasking performance for a specified number of days, thus alerting the operator to possible degradations in the system. The graphs provide continuous feedback on the status of the system and trending of key performance parameters.

Fig. 19 shows an overview of the architecture and the flow of data through the system to populate the dashboard.

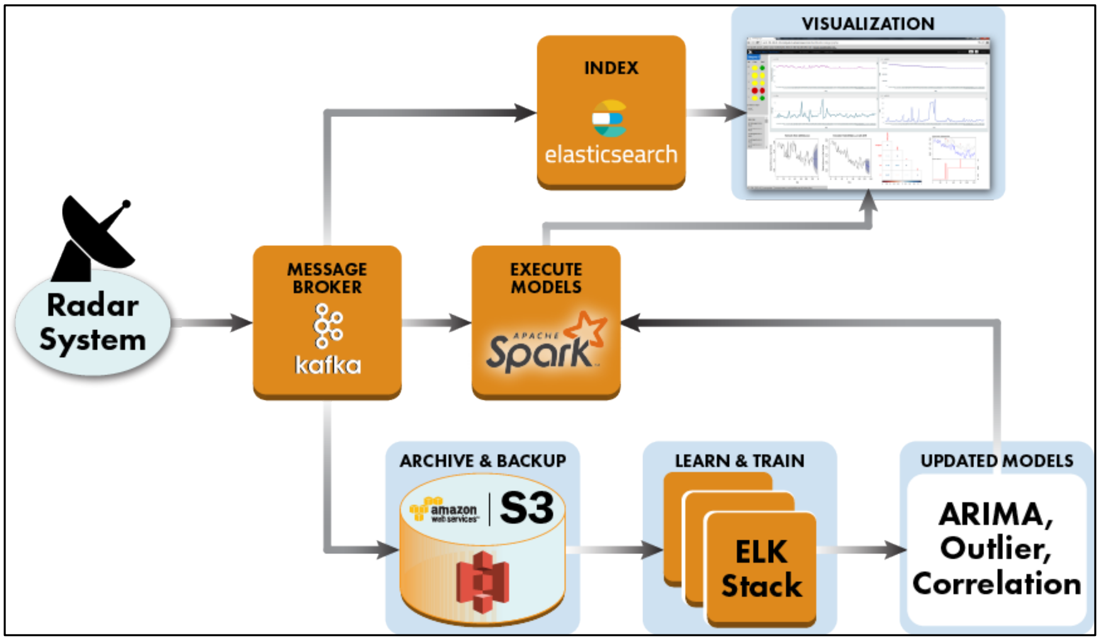


Fig. 19. Notional deployment architecture when using real-time data

For all of these features to work, the system has to continuously ingest and analyze data with the ELK stack and R for the most up-to-date visualizations and models of the data. By doing this in real time, the simulation proves that these models can be run fast enough that the operators and analysts can make conclusions from the data and act on them before the next data set arrives.

CONCLUSIONS

This paper presents a methodology toward developing operational analytics to solve analysis problems with large datasets such as those that may be encountered at operational sites. The three step approach of discovery, modeling and operations enables sites to understand the data and correlations between data sets, model the data interactions, and make predictions based on the data. As shown in this study, the data can be very complex and understanding the interactions and reasons for the data performing the way it does requires a thorough understanding of the system.

This study shows that using operational analytics integrated with big data technologies and predictive techniques it is possible to analyze, diagnose, and predict possible sensor anomalies. The simulations show the importance of being able to drill down into the data to extract the root cause of anomalies both as part of discovery and for later analysis. From this analysis it is clear that multiple approaches and multiple datasets and correlations must be examined to assess sensor anomalies and discriminate anomalies from normal operational changes.

Projections calculated by this study allow analysts to see problems in advance and take action before action is required. This allows for far smoother, more continuous operation by minimizing downtime. In the past, analysts may not be able to see a trend until several days of poor performance. However, by using the methods developed in this study, it is possible to observe trends before they happen and potentially avoid drops in performance by taking preventative actions. Our model also detects outliers in tasking performance data, which provides an easy way to differentiate random fluctuations from more fundamental problems.

Future work should focus on specific areas of understanding the data and modeling different data phenomenologies. Also key is improving the modeling process to incorporate short term anomalies and their impact on the system. Understanding the removal of outliers in the projection of the data would lead to better estimates and confidence in the projections.

The operational analytics and big data technologies methodology and architectures presented here should be considered for current and future sensors or systems where the volume of data is likely to overwhelm an analyst. Due to the complexities of the data, it is very possible to draw the wrong conclusions if only a limited dataset is used for analysis. Drawing the right conclusions from the data requires an organized approach toward data analysis and exploitation such as is presented here. Drawing the correct conclusions will lead to cost savings and ability to maintain key space surveillance assets.

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

REFERENCES

Technical references

1. Coughlin, Joseph, R. Mital, W. Fu, Using Big Data Technologies and Analytics to Predict Sensor Anomalies, Advanced Maui Optical and Space Surveillance Technologies Conference, 2015.

The following references are for the software used:

2. Apache Spark [Computer Software] http://spark.apache.org/
3. The ELK Stack [Computer Software] https://www.elastic.co/
4. R Studio [Computer Software] https://www.rstudio.com/