Machine Learning Techniques Applied to Flight Test Data Evaluation

Kelton Busby, kbusby@aerotec.com, AeroTEC, Inc (USA), Rebecca Hattery, rhattery@aerotec.com, AeroTEC, Inc (USA)

ABSTRACT

This paper proposes an application of machine-learning methods to the analysis of flight test data. A set of training data is used to develop relationships between measurands and generate predicted behavior. These relationships are forecast onto data from the same aircraft model to identify unpredicted measurand behavior. The application of this method may significantly reduce post-test identification time of problematic measurands. Statistical analysis methods are used to determine quality of identified relationships and reduce instances of confirmation bias. The importance of a carefully selected training data set and development of robust relationships with low collinearity is emphasized. The developed application demonstrates faster instrumentation failures diagnosis than traditional methods. An area of continued research includes application of highly accurate models developed for an aircraft to reduce required instrumentation.

1. INTRODUCTION

Safety, technical integrity, and certification requirements generate the need to monitor and record aircraft test data from existing aircraft bus data and installed measurands. An aircraft development and certification program can require thousands of simultaneously recorded parameters. Analysis of data gathered on test vehicles traditionally requires labor intensive methods for managing sensor health and data integrity. Currently, data is monitored during and after testing by subject matter experts (SMEs). The SME relies on instrumentation engineers to ensure measurands are functioning correctly. If a measurand is determined to be malfunctioning, the SME documents the issue and instrumentation engineers address or correct the issue. In sensor networks that are not monitored for errors, bad data may be recorded for an extended period. Automated analysis can aid in prompt detection of sensor systems that have gone awry. Some potential uses of this information are: Early detection of test or safety critical sensor failures, detection of out of calibration exposure, drift, or unusual response, reduction of required sensor installations, increased redundancy in control systems, and data replication and replacement for faulty parameters. This paper will present an overview of machine learning terminology and theory to familiarize the reader with the proposed techniques. A methodology is described which is employed to analyze sets of flight test data. The data set will be described. The machine learning methods will be used to locate and predict measurand health. A brief discussion of areas of future research and industry potential is in the conclusion of the paper.

2. METHODOLOGY - THEORY

This paper presents a methodology for using machine learning methods to detect and handle unpredicted measurand behavior. There are many critical components to creating a reliable method for detecting such abnormalities. The methods found to be successful in this paper are not meant to be comprehensive or exhaustive but represent successful applications on flight test data as discussed in Section 4. The general step by step procedure for detecting data abnormalities using machine learning is:

- Gathering Data
- Exploration & Cleaning
- Selecting a Model
- Analyzing Results.

Prior to reviewing the steps in detail, there is a brief introduction to relevant terminology used in this paper.

2.1. Terminology

This paper uses terminology present in data science and statistical or machine learning literature. Some terms are less common in test engineering. Those are laid out here for clarity.

Feature – A column in a dataset, Figure 1. Represents a characteristic or property of an observation.

Hyperparameter – A control input to a model that adjusts the model algorithm's optimization function.

Measurand – Recorded data representing some physical or engineering phenomena. Ex. Temperature

Model – A quantitative response as a function of independent variables. Approximates the true response of a dependent variable as a function of the independent variables.

Overfitting – model that applies relationships between the dependent variable and the independent variables that do not actually exist. These perceived relationships are noise and are ideally not be included in the model.

Prediction – The output of a model given values for the independent variables in the model.

Training Data Set – Recorded data used to generate a model.

Test Data Set – Recorded data used to perform final performance checks on the model.

Underfitting – An underfit model misses true relationships between the dependent variable and the independent variables.

Validation Data Set – Recorded data used to analyze model performance and tune the model after model is created / fit / generated from training data

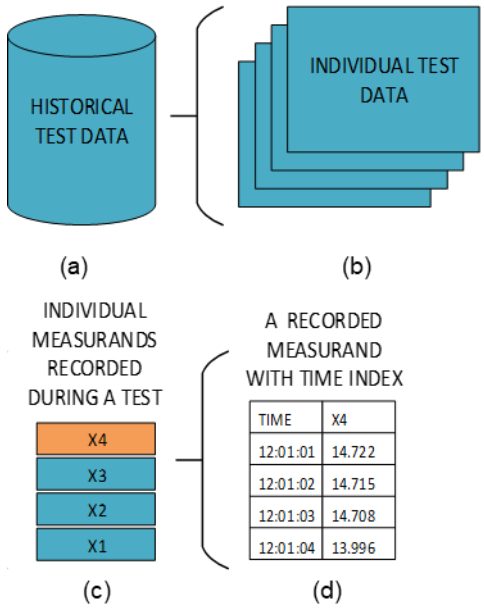

Figure 1: Breaking Down Available Data to Features with Time Index

2.2. The Dataset

A set of historical test data from a test article is necessary to use this paper's methodology to detect unpredicted measurand behavior. The data is used to develop statistical relationships between measurands. The following are important considerations in gathering data to ensure success in implementation: Data formatting, time alignment, and sampling rate. The data format is assumed to be tabular where rows represent samples and columns represent features. An example of properly formatted data is provided in Table 1.

2.2.1. Time Alignment

Time series data, data capture synchronously and indexed in time order at a common time spacing, is a key component for developing strong statistical relationships. Synchronously captured time series data is the most common data type from flight test. Data captured asynchronously cannot be relied on to create a consistent usable relationship that generalizes well to future data. The more complex the data system or the larger the number of measurements, the more critical alignment becomes. Data manipulation may be required for out of alignment data. Methods exist for dealing with unevenly spaced time series, such as linear interpolation, but the quantification of introduced biases is difficult. Table 1 presents a set of time-aligned synchronous data is below. Each feature is sampled every second on the second.

Table 1: Time Aligned Sample Data

Time	Exhaust Temp	Inlet Temp	RPM	Manif. Press.
12:33:57	1320	76	2200	12.3
12:33:58	1320	76	2210	12.3
12:33:59	1314	77	2210	12.4
12:34:00	1320	76	2215	12.4
12”34:01	1317	75	2230	12.4

2.2.2. Sample Rate

Sample rate is the frequency at which a measurement is taken. The sample rate may vary between features; this variance may also cause time asynchronization. Data sample rates and variation should be considered before using data as part of a machine learning application. If data is not all taken at the same sample rate, imputation, interpolation, up/down sampling, or data removal needs performed. If the slower sample rate data contains the dependent variable, simple down-sampling of the independent variables with higher sample rates is enough. If the higher sample rate data contains the dependent variable, then the independent variables with lower sample rates will need to be up-sampled.

2.3. Exploration & Cleaning

Data Exploration is a data analysis technique that provides insights into the data. For numeric data, data exploration includes plotting feature distributions, creating correlation matrices, and reviewing statistical summaries to better comprehend how the data looks, what kind of variability exists, and the presence of missing values in features. The result of is an understanding of the types of data, the shape of the data distributions, significant statistics about the data, and what relationships exist in the data set.

In flight test, data is usually numeric time series data. This paper assumes numeric data which is either numeric by default (temperature measurement) or by feature engineering (discretization).

Data visualization is the use of plots of data statistics and relationships to understand more about the data set including the data distribution shapes, range, outliers, patterns, and multicollinearity. Different models used in this methodology make a variety of assumptions about the data which need to be understood and handled. If many linear relationships exist in the data, it may be wise to use primarily linear models like multiple linear regression. Many linear relationships are a sign to check for multicollinearity, which can occur when independent variables are highly correlated with one another. Multicollinearity is a common issue with test data and should be handled appropriately as its impact on models can be significant.

An easy way to detect multicollinearity in the data is to check the Pearson correlation coefficient (PCC) between each feature in the data. The PCC is often visualized in a correlation matrix. The PCC is distributed in the following manner: -1 is perfectly negatively linearly correlated, 0 is not linearly correlated, and 1 is perfectly positively linearly correlated. A correlation matrix shows the PCC between every feature in the dataset in a table. A PCC matrix for some sample data is presented in

Table 2.

Table 2: Correlation Matrix Showing PCCs

	Exhaust Temp	Inlet Temp	RPM	Manif. Press.
Exhaust Temp	1.000	-0.395	-0.281	-0.612
Inlet Temp	-0.35	1.000	-0.645	0.000
RPM	-0.281	-0.645	1.000	0.667
Manif. Press.	-0.612	0.000	0.667	1.000

2.3.1. Cleaning

Clean data makes for better model performance, reduces noise, and ensures that data used by the model are valid, accurate, complete, and consistent. Cleaning includes building an automated process for ensuring datatypes are correct, missing data is handled, synchronously sampled data is aligned, and dates are formatted and recognized. Clean data is significantly easier to work with, easier to maintain, and can be pushed through machine learning algorithms without errors.

Cleaning is usually handed by building a program, script, or algorithm in a programming language like Python. Cleaned data should be consistent in format for input into a machine learning algorithm. Figure 2 compares data before and after cleaning.

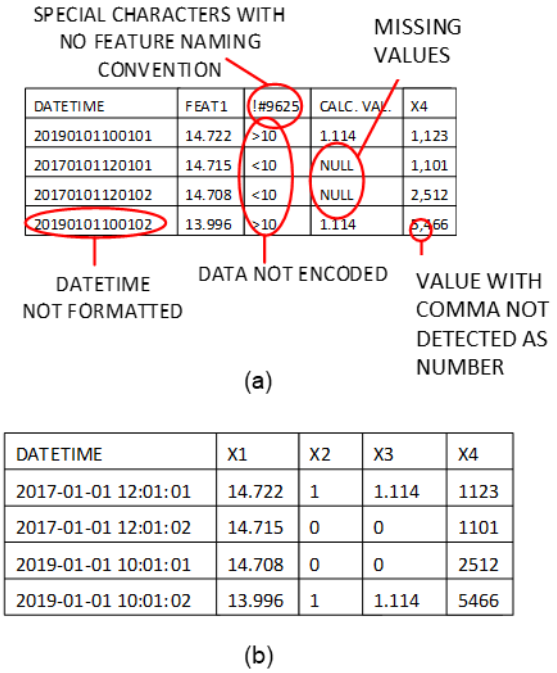

Figure 2: Data Before (a) and After (b) Cleaning

2.4. Selecting a Model

After Data Exploration it is possible to make a model selection. An interesting aspect of identifying unpredicted measurand behavior with machine learning is that almost every feature in the dataset is eventually a dependent variable to be modeled using the rest of the dataset, as shown in Figure 3. This is unlike traditional machine learning problems which have a single dependent variable and many independent variables.

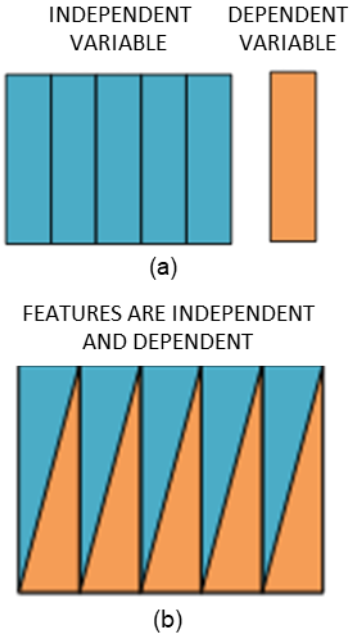

Figure 3: Difference between Traditional (a) and Detecting Abnormalities (b) Machine Learning Problems

The work presented here focuses on a problem type in which the dependent variables are numeric and have relationships with other numeric data. This paper focuses on this type of problem. In algorithms or machine learning this is called a supervised regression problem. Supervised means that there is a training set available to feed through the algorithm with known independent and dependent values. Regression refers to the fact that the result of a prediction is a value for the dependent variable.

There are many algorithms compatible with solving supervised regression problems, including simple linear regression, multiple linear regression, polynomial regression, multilayer perceptron (Neural Network), and Regression Tree / Random Forest. This paper will focus on multiple linear regression. Model requirements can drive model selection as much as model performance. An example requirement is that an engineer must be able to interpret how a prediction was made.

A parametric model, which takes on a specific function like linear regression, is interpretable. Non-parametric models have the flexibility associated with not making any explicit assumptions about the form of a function, but as a result do not provide interpretable results. The result is simply the result after passing independent variable values through the model.

Training multiple models is a productive way to get a baseline set of performance metrics to compare against.

In this paper, accuracy, as the proximity of the predicted value to the actual value, is the most critical performance metric. Further details on understanding model

performance follows in Section 2.6.

Models can be applied by loading the data into any analysis tool with machine learning libraries or the functionality to implement learning algorithms. It is important to select a software tool or suite of tools capable of loading data, training a model, and reviewing results. Python 3, R, and MATLAB are all suitable tools for this type of analysis. The research in this paper was performed using Python 3.

With cleaned and prepared data, it is possible to model a feature of the dataset. A model in this context makes some prediction about the value of one feature (parameter, column, or measurement) given the values of multiple other features in the dataset. The prediction is a function of the independent variables.

The model built approximates the true function of the dependent variable given the information in the dataset. The predicted result given the values of relevant independent variables is the result of interest.

As an example: A multiple linear regression model takes on the following form:

$$\hat{y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n$$

Where:

- $\hat{y}$ represents a prediction
- β represents coefficients determined by the algorithm
- X represents the independent variables used in the model

With this function, it is possible to make predictions when provided with values for all X. β values are determined by the model by minimizing the error in the prediction on the validation set.

2.4.1. Applying a Model to the Data

To properly evaluate the results of a model, the historical data must be split into training, validation, and test sets, Figure 4. A common way to split up the available historical data is to split it 80/20 twice. The first split gives training and testing data. Then the training data is split again 80/20 into training and validation. The test data is set aside until there are final model evaluations to be done.

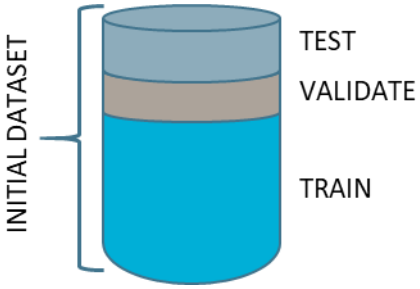

Figure 4: Train, Validation, and Test Split

Fitting the model programmatically is often as simple as calling a fitting function in a programming language like Python on the independent variables with the dependent result. When this is complete, the model can be evaluated.

2.5. Feature Selection & Regularization

With the data split into train/validation/test sets, a model can be built with every feature from a dataset. In general, it is better to reduce the independent features in each model to only relevant features. There are multiple reasons why this is the case, including requiring less data, reducing dependencies, improving interpretability, and improving model performance.

While this paper will not go into detail of feature selection algorithms, the presented methodology does rely on feature selection. Some options for multiple linear regression are forward, backward, or mixed stepwise selection, LASSO regression, and elastic-net. The result of a properly implemented feature selection algorithm is a reduction of independent variables to a smaller set of features that provide a better performing and less complex model.

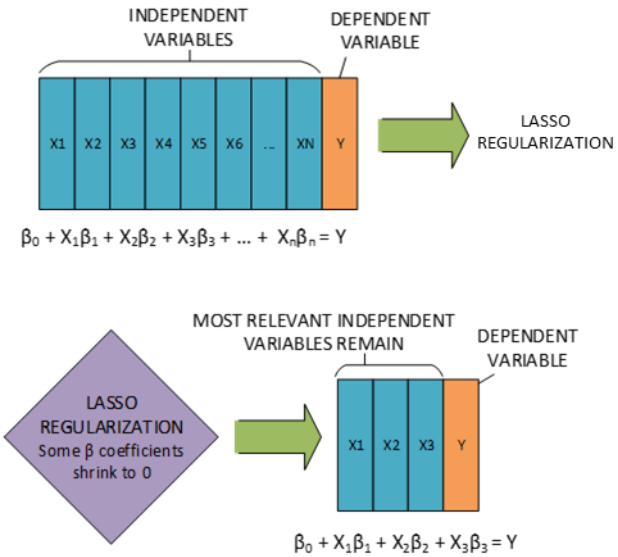

Figure 5: **Feature Selection with LASSO Regularization**

2.5.1. Tuning a Model (Adjusting Hyperparameters)

After the models are generated, a comparison of their performance on the validation set is performed. It is important to note that some models have input hyperparameters which are control inputs that define the final structure of a model. Adjusting these control inputs can impact the results of the model.

It is appropriate to tune selected models for best performance. Some models do not have hyperparameters to tune (e.g. multiple linear regression).

With tuned models it is again possible to check the results against the test set.

LASSO regression does have hyperparameters to adjust. The most significant hyperparameter is the alpha (α) or L1 term. Succinctly, increasing the α parameter adjusts the optimization function to shrink more β coefficients to zero. This reduces the number of relevant independent variables that get included in the model. Adjust α to find a balance between bias and variance for the model.

2.6. Results

A model that has been fit to historical data from a test article, tuned with a validation set, and checked against a test set can be applied to new data from the same or other applicable test articles. Feeding the model new data is the critical step in determining if current measurand output is within expectations.

Predictions should first be made on the test set of historical data that was held out of the training data. The test data can be used to evaluate the performance of the model before deploying the model to evaluate new data. Model performance is primarily evaluated by looking at error. Error is the difference between the model output and the true output for the dependent variable at a time step.

The action to take based on model detection of data abnormalities or erroneous measurands is organization and test article dependent. For the purposes of this paper, the action is to plot the relevant time series (including the predicted output and the actual output of the flagged measurand).

Using statistical analysis, a Root Mean Squared Error (RMSE), and an error distribution with standard deviations can be gathered.

RMSE is a measured of how good the fit is across the entire test set. It is the square root of the sum of the squared error in the prediction at every time step divided by the number of time steps. the higher the RMSE the greater the error in the prediction.

$$RMSE = \sqrt{\frac{1}{N}\sum_{N}\left(\widehat{Y}_i - Y_i\right)^2}$$

Error distribution is a measure of the distribution of error, ideally centered around 0.

The error statistics in the test set predictions are the core component to be compared against future data to detect problematic measurands. It is out of scope to explore all statistical measures and ways to analyze incoming data for abnormalities but AeroTEC has found success comparing test and new data RMSE and error distributions.

3. METHODOLOGY – IN PRACTICE

AeroTEC has implemented the presented method using data from a test vehicle. The purpose of the application was to use machine learning methods to detect and identify failed parameters in a set of test data. This method has aided in discovering parameter failures significantly faster than traditional engineering methods. This section will discuss the process described in section 2 as applied to a sample test data set.

3.1. The Dataset

In the rest of the Methodology – In Practice section, an actual set of flight test data is used to show the application of the theory. The data is numeric and is composed of the response of each sensor when sampled during each test. Rows of the data correspond to samples, and columns correspond to sensors (or measurands). The dataset in this practice section, prior to any cleaning, consists of 12 unique test runs of a single test article. Each test is over an hour long sampled at 1 Hz, with over 4000 unique columns (measurands). The dataset includes over 144 million unique data points. The tests are performed with unrelated goals in unique test environments for the sake of conservatism.

3.2. Exploration & Cleaning

Per the methodology, the dataset needs to be explored and cleaned in a standardized fashion so that future test data may use the same procedures.

An easy way to handle the data is to load it into memory in a tool built for handling large datasets. Tools like Microsoft Excel work great for visualizing small datasets, but larger ones require the use of development environments and a programming language. Python 3 is used in this practice section.

Rows or columns with missing values are removed, or the values are interpolated, imputed, or otherwise handled. It is best practice to remove missing values entirely from large datasets.

In the case of this dataset, the following cleaning was performed:

- Data moved from database to individual HDF5 (binary files) for accessibility
- Removal of columns with low or no variability
- Removal of columns with significant number of missing values
- Removal of rows with missing values
- Handling of timestamps to be standard format, and read by Python

3.3. Selecting a Model

The critical component of data preparation is to fit an accurate model to the data to make predictions about a measurand's value based on other measurand values. The theory section on models discussed the variety of available models to use. In this section the focus will be on building a regularized multiple linear regression model. The reason is that AeroTEC has found consistent success in detecting unpredicted measurand behavior using this type of model. Following the theory section for selecting and using model will provide an appropriate model selection for specific cases.

The overall procedure for fitting and applying a model can be seen in Figure 6 and Figure 7. After a model has been trained on historical data (step 1), it is necessary to evaluate its performance on the validation set of data (step 2). This is where a variety of regression model metrics are useful to determine how 'good' the model is. While it is out of scope to review model evaluation metrics, some useful ones are: the coefficient of determination, the Akaike Information Criterion, RMSE and MAE. After evaluating these metrics for each model generated, one can get a better idea of what model to proceed with, what hyperparameters to edit for the model, and an expectation for performance on test set data (step 3 & 4). Note that after adjusting models on validation sets, it is critical not to adjust the model after test set evaluation. If performance is not meeting requirements, the process starts over entirely. If model characteristics are adjusted after fitting to test set data,

the likelihood for overfitting increases dramatically. After gathering RMSE, MAE and plotting error distributions of test set data model performance: it is appropriate to pass new data through the updated machine learning model (step 5). When the results are computed, evaluation can be performed using methods discussed in the results section below to determine if measurand data is within expectations.

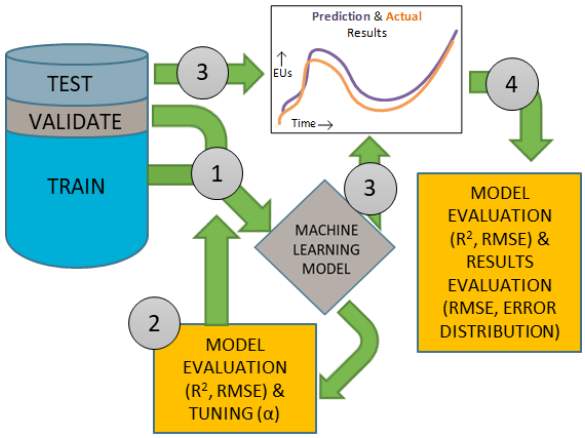

Figure 6: The Meat and Potatoes: Creating and Evaluating a Machine Learning Model

After gathering RMSE, MAE and plotting error distributions of test set data model performance: it is appropriate to pass new data through the updated machine learning model (step 5). When the results are computed, evaluation can be performed using methods discussed in the results section below to determine if measurand data is within expectations.

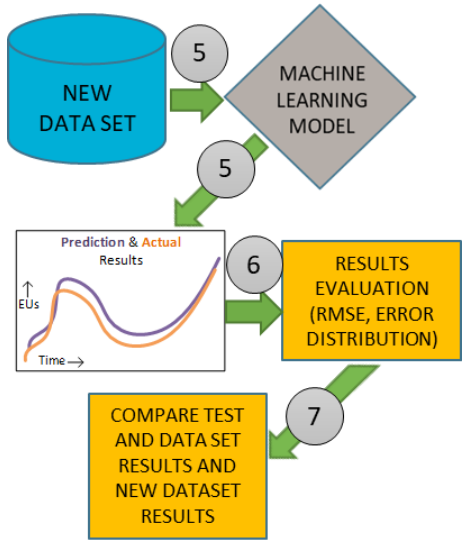

Figure 7: The Meat and Potatoes: Applying a Machine Learning Model to Data

There is nuance in selecting and adjusting a model. AeroTEC has found regularization to be a critical step to apply to numeric test article data.

3.3.1. Feature Selection using LASSO Regularization

A LASSO regularized model has less dependencies than a standard multiple linear regression model. This is important to reduce the complexity of the model and reduce the likelihood that its dependencies become too cumbersome to maintain. In the case presented here, regularization for the multiple linear regression model is achieved through the Least Absolute Shrinkage and Selection Operator (LASSO) method. The impact of the LASSO model is that LASSO removes independent variables that are largely unrelated to the output. A reduction in independent variables in a model has the following effects:

- Less failure prone. A model with less required measurands is less likely to lose one of its dependencies during test.
- Less complexity. Simpler models are more interpretable by engineering and are less likely to fit to noise in the data.
- More flexibility. Models with fewer dependencies are more likely to remain useable as test article measurands change.

LASSO reduces the β coefficient values associated with each X independent variable. See the summary visualization in section 2.5 above.

4. RESULTS

As described in section 2.6, results are evaluated after a model's baseline results have been evaluated on a test set. New data can now be passed through the model and data abnormalities can be flagged.

4.1.1. Using RMSE to Detect Abnormalities

The root mean squared error found in the test set can be compared to the root mean squared error of any new data a model is applied to. This method can be used to detect many types of data abnormalities, but it is particularly useful for catching low magnitude, long duration errors. Issues like sensor drift, or out of calibration sensors can both be detected by evaluating RMSE.

If the RMSE of any new test data exceeds some criteria set based on the validation set RMSE, the measurand is flagged for review. For instance, a basic criterion may be that the RMSE of the new data is over 2 times that of the test data.

AeroTEC has found success in detection of faulty measurands due to a variety of issues using a comparison of test data and new data RMSE values.

4.1.2. Using Error Distribution to Detect Abnormalities

Using the error distribution in the validation set is a method for detecting short duration high magnitude errors. An example of a short duration high magnitude error is a data spike due to a temporary loss of signal. An error distribution of model predictions on a test set defines expectations for level of error in the model's prediction. If actual and predicted results deviate by more than a set difference based on the test error distribution, the measurand can be flagged. Using the In-Practice flight test data set, an example is visualized below. Note the dashed red error bars signifying boundaries for flagging a bad parameter.

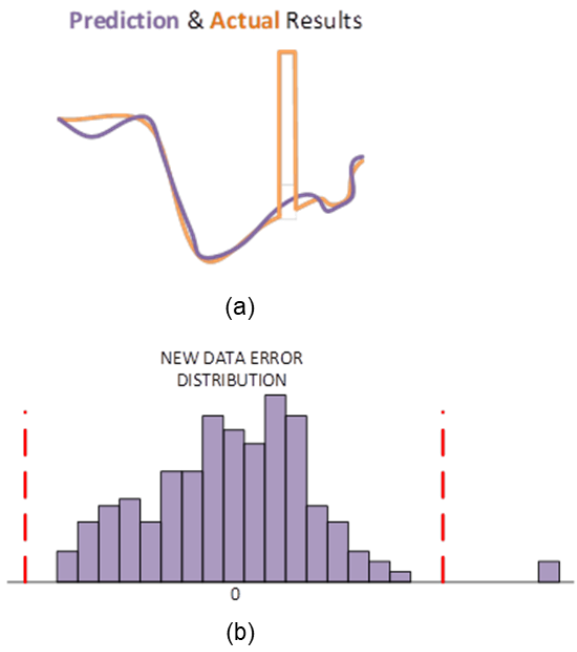

Figure 8: Visualization of Short Duration High Magnitude Error

Using error distribution to detect abnormalities has been used successfully at AeroTEC for detecting common sensor problems like thermocouple grounding, that are more difficult to detect using methods like RMSE that are more suited to longer trending error.

4.2. Deploying

AeroTEC has deployed models by generating a report after every new test detailing what measurands may have failed per automated results evaluation. The reports can be delivered to the parties responsible for ensuring proper measurand operation. The reports include time series plots of both the prediction and the actual measurand result as seen on the left side Figure 8.

5. DISCUSSION

Finding a solution for detecting failed measurands or data abnormalities using machine learning offers a unique capability to quickly assess the quality of test data. The rapid detection of failed measurands is valuable to a test provider as it can be used to diagnose system failures and reduce the number and frequency of repeated tests. Industry experience in test has shown that managing many measurands on a test article often becomes a herculean task. This paper has introduced and provided an example for any organization or individual to start the process of using machine learning methods to reduce test cost and improve timeliness in detecting problems with data.

Details of the methodology in certain sections have been purposefully omitted to preserve AeroTEC's intellectual property. An example of an omitted detail is in determining the algorithm's sensitivity for detecting failed measurands. Sensitivity could be determined by setting a difference in test set RMSE and new data RMSE that causes a measurand to be flagged as producing questionable data. The same determination could exist for evaluating error distributions.

Evaluating measurands for valid data after testing is a necessary component of preparation for data analysis, and of preparing for the next flight. Unfortunately, traditional methods of data evaluation are very time consuming and expensive. Machine learning models are a fast, reliable, and inexpensive method to provide data analysis. The 144-million-point data set was evaluated using a single commercially available laptop computer.

The time commitment for this method comes from setting up the models for a dataset, especially the first time through. However, that time investment is a one-time occurrence rather than repeated effort for every test event. The benefit of this time investment upfront is that the models will last if they are still generalizing to the new test data. Every new test that is analyzed by the models is time saved for the organization in doing the analysis manually. At AeroTEC, we're using the results from machine learning detections of data abnormalities to drive decision making on what measurands to troubleshoot, and in what way.

Regarding future use of this capability, AeroTEC has tested multiple use cases outside of data abnormality detection. There are cases in which models can be used in place of new sensor installations when the cost difference makes sense weighed against the requirement for a true measurement. For instance, after flying a large envelope with a trailing cone, it is possible to build a model of its output, review the accuracy of the predictions, remove the trailing cone, and still gather predicted trailing cone response. This has the benefit of no longer requiring trailing cone equipment, leak checks, or procedures while freeing up a data system channel. AeroTEC has tested this capability and found accuracies within the error boundaries of the originally installed sensor.

Another application of this technology is to run both modeled and true sensor responses in a telemetry room or onboard to make "knock-it-off" calls if the true response changes significantly when compared to the model. An example is a wing anti ice sensor that may rise significantly faster than a modeled value. The baseline model response is a suitable sanity check for an engineer to make a call to stop the condition before physical damage or an unacceptable risk level occurs.

6. CONCLUSIONS

AeroTEC used machine learning methods to develop a solution for detecting questionable data from measurands on a test article, then replacing with modeled data if required. The result is a solution that can monitor every measurand on a test article during or after test for unusual behavior. Measurands with problems are addressed faster with better direction on troubleshooting.

This result affects everyone from the technician performing the installation up through engineering and data analysts to the business management. Ensuring accurate data is maintained through a test program creates improvement to test efficiency and cost.

AeroTEC is excited to continue applying and testing this powerful technology and is interesting in working with

organizations and individuals who want to evaluate this technology for their work.

7. REFERENCES

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. An Introduction to Statistical Learning : with Applications in R. New York :Springer, 2013. Print.

Géron, Aurélien, author. Hands-On Machine Learning with Scikit-Learn and TensorFlow : Concepts, Tools, and Techniques to Build Intelligent Systems. Sebastopol, CA :O'Reilly Media, 2017. Print.

Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011