

Method and System for Predicting Hydraulic Valve Degradation on a Gas Turbine

James D'Amato¹, John Patanian²

^{1,2} GE Power, Atlanta, GA, 30339, USA

james.damato@ge.com

john.patanian@ge.com

ABSTRACT

This paper examines the development of a data-driven anomaly detection methodology for servo-actuated hydraulic valves installed in a gas turbine fuel delivery system. Degraded operation of these valves is a leading cause of unavailability for gas turbine driven power plants. Nearly eighty potential features were generated from the limited raw sensors and control system signals through a combination of domain expertise, statistical feature extraction, and insight gains from prior physics-based simulations. Important features were down-selected by examining the behavior of the features using several years of operating data in conjunction with known field failures. Univariate statistical techniques were used to eliminate candidate features with limited capability to distinguish healthy from abnormal operation. A final machine learning model was generated using a process of recursive feature elimination. This paper will also touch on the practical implications of deploying a machine learning model in a real-time production environment.

1. INTRODUCTION

This document has been developed to accompany the data analysis completed during development of an anomaly detection analytic for hydraulic gas control valves used in heavy-duty gas turbines. Gas control valves are a component of the gas fuel delivery system in which each valve controls fuel delivery to a separate manifold supplying the combustion system fuel nozzles. For Dry Low NO_x (DLN) 2.6 and greater fuel systems, this consists of regulating the distribution of fuel delivered to the Primary (PM1), Secondary (PM2), Tertiary (PM3), and Quaternary (QUAT) fuel systems, as shown in Figure 1. The fuel flow distribution to each combustion fuel system is a function of combustion reference temperature and Inlet Guide Vane (IGV) temperature control mode. The complete gas fuel system consists of the gas fuel Stop/Ratio Valve

James D'Amato et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

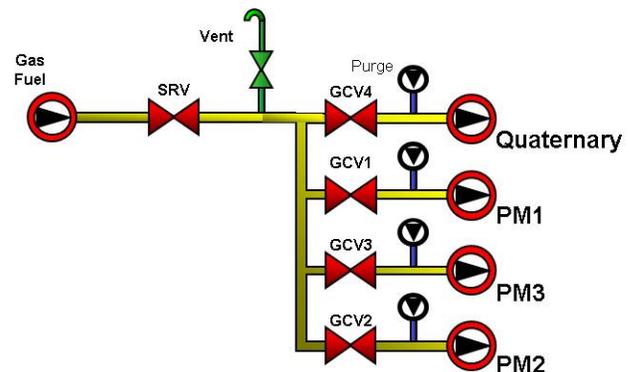


Figure 1. DLN2.6 Gas Fuel System.

(SRV), Primary Gas Control Valve (GCV1), Secondary Gas Control Valve (GCV2), Tertiary Gas Control Valve (GCV3), and Quaternary Gas Control Valve (GCV4). The SRV is designed to maintain a predetermined pressure at the control valve inlet. The GCVs are in place to regulate the desired gas fuel flow delivered to the turbine in response to the fuel command from the control system (Davis, 1996).

1.1. Commercial Implications

Gas turbine hydraulic GCV failures are one of the leading causes of gas turbine trips and failed start-up. As opposed to an operator initiated shutdown sequence, which is a gradual decrease in load and turbine speed, a trip is initiated by the safety protective logic within the control system by instantly stopping fuel flow to the gas turbine. The thermal shock of a gas turbine trip, especially a trip during base-load operation, causes increased degradation of gas turbine performance and hot gas path parts life (Bernstein & Allen, 1992) and (Ravi, Pandey, & Jammu, 2010). Similar thermal stresses are seen during start-ups as many hot gas path parts are commonly life-limited by the number of start cycles to which they are subjected to in service, rather than by the hours of service (Carter, 2005). A false start-up occurs when a start-up is initiated by the operator, however the control system will trip

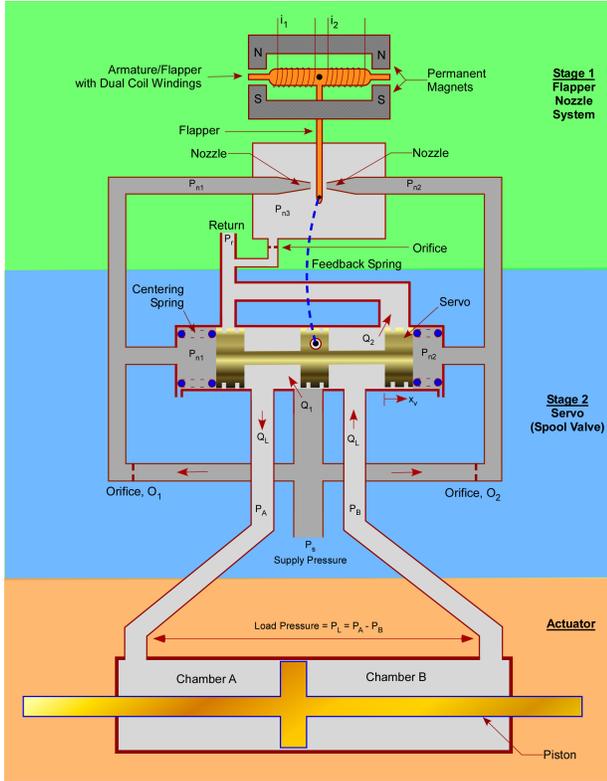


Figure 2. Hydraulic Servo-Valve Cross-Section.

the gas turbine during the start-up due to safety protective constraints.

1.2. Hydraulic-Valve Degradation

All subsequent hydraulic-valve degradation modes can be referenced with Figure 2. As identified by (Macaluso, 2016) and (Mornacchi, Vachtsevanos, & Jacazio, 2015), possible GCV degradation modes include the following:

1. Reduction in torque from the respective torque motors. This leads to a progressively slower response of the servo-valve.
2. Contamination within the fuel nozzles. As dirt and debris accumulate within the fuel nozzles, this leads to a slower response of the GCV.
3. Stiffness variation of internal feedback spring of the GCV, which is generally caused by yield in strength due to excessive loads or to normal aging of the component; involves hysteresis phenomena and instability.
4. Increase of the backlash at the mechanical interface between the internal feedback spring and spool. This is the result of a wear due to the relative movement between these two parts giving rise to an increasing hysteresis in the servo-valve response, which leads to an instability.
5. Variation of the friction force between spool and sleeve. This is due to a silting effect associated either with debris

entrained by the hydraulic fluid or to the decay of the hydraulic fluid additives which tend to polymerize when the fluid is subjected to large shear stresses.

6. Increase of the radial clearance between spool and sleeve and change of the shape of the corners of the spool lands due to wear between these two moving parts.

The aforementioned GCV degradation modes will eventually lead to a gas turbine trip and potentially an unplanned outage. Typical mitigation techniques involve the utility company to actuate the GCV during the outage and/or servicing the GCV.

1.3. Gas Control Valve Instrumentation

Each GCV has limited instrumentation, including a current applied by the control system and a position indication supplied from a Linear Voltage Distance Transducer (LVDT). There is also a control signal which indicates the commanded GCV position. To summarize, there are three control signals for each GCV (Mornacchi et al., 2015):

1. GCV position command: corresponding to the position request from the control system.
2. GCV position feedback: position indication acquired from the LVDT and used to close the control position loop.
3. GCV current feedback: generated by the control system coinciding with the compensated error. It is used to control the GCV.

The control system applies a current to actuate the hydraulic valve. The current applied is proportional to the error between the command signal from the controller and the feedback reading from the valve instrumentation. As the error between command and reference signal increases, the control system compensates by increasing current to the respective torque motor. A faulty valve is unable to follow the control system reference signal, and thus the error between command and reference increases and becomes erratic as the valve components wear or otherwise have a fault. When a valve is not commanded to move, a current is still applied to overcome the force of a feedback spring internal to the hydraulic servo. This is called the null bias current.

2. METHODS AND ANALYSIS

This section and the following subsections discuss the generated data set and the methodology applied throughout the exploratory analysis, feature engineering, data filtering, and feature selection activities. The overarching goal was to classify each GCV across the entire monitored gas turbine fleet as being healthy or degraded. Due to the lack of instrumentation on each GCV, significant work was applied upfront to improve model performance downstream. Furthermore, the upfront exploratory analysis and data engineering methodologies were implemented with the intent of classifying a GCV

as degraded with sufficient lead time of the GCV causing the gas turbine to trip, thus minimizing unplanned outages for the customer base.

2.1. Input Data Description

The data set for the gas control valve was generated from events captured in an events database for 7FA.02, 7FA.03, and 7FA.04 gas turbines between January 1, 2009 and May 21, 2014. Events extracted from a Hadoop Distributed File System (HDFS) cluster were filtered in which those events led to a forced outage or a failed start, and filtered by those events where failures were attributed to one of the four GCVs. In addition, some failures were not included in the training data sets, such as failures where the root cause was determined to be wiring, controls related, etc., which are generally failures that will not manifest as a pre-cursor that can be detected with sufficient lead time to take action on the anomaly. As previously mentioned, the four GCVs are controlled to a position, whereas the SRV is controlled to maintain a pressure signal, measured as FPG2. Table 1 lists the raw signals extracted. Historical data was extracted from HDFS and the following subsections will discuss the features that were generated from the raw data using a combination of statistics and domain based features.

2.2. Time-Series Feature Calculation

This step involved calculating derived features at the same time interval as the raw data. It is important that this step occurred prior to any data filtering, so that the derivative calculations were truly representative of the difference between consecutive time samples. Table 2 identifies the calculated features derived from the raw control system data.

2.3. Training Data Filtering

The training data was filtered on two criteria, that (1) the gas turbine was operating in either base-load or part-load conditions and (2) that all GCVs were actively controlled to a reference signal. The exception to this criteria was GCV4, which for some units is not in service during operation.

2.4. Aggregate Feature Calculation

Similar to the aggregate feature selection approach taken in (Ravi et al., 2010), the following aggregate features were calculated on a four hour window, with the raw data sampled at a five-minute interval:

1. Sum, mean, median and standard deviation of GCV1-4 position error
2. Shown in Equation 1, DPC_i represents the derivative of the Gas Turbine Generator Output (DWATT) w.r.t the GCV current feedback, where i represents the respective GCV. DPC_i was then aggregated to obtain the mean,

Table 1. Extracted Control System Signals

Feature	Description	Units
TNH	Turbine Rotor Speed	%
CTIM	Compressor Inlet Temperature	°F
DWATT	Gas Turbine Generator Output	MW
FAGPM1	GCV-1 Servo Current Feedback	A
FAGPM2	GCV-2 Servo Current Feedback	A
FAGPM3	GCV-3 Servo Current Feedback	A
FAGQ	Quat Servo Current Feedback	A
FAGR	SRV Servo Current Feedback	A
FPG2	Interstage Gas Fuel Pressure	psig
FPG3	Gas Pressure Upstream of SRV	psig
FPRG1OUT	Control Command Interstage Pressure	psig
FPRGOUT	Alias for FPRG1OUT	psig
FSRPM1_PCT	GCV1 % of Total Flow	%
FSRPM2_PCT	GCV2 % of Total Flow	%
FSRPM3_PCT	GCV3 % of Total Flow	%
FSRQT_PCT	Quat % of Total Flow	%
FSGPM1	GCV1 Position Feedback	%
FSGPM2	GCV2 Position Feedback	%
FSGPM3	GCV3 Position Feedback	%
FSGQ	Quat Position Feedback	%
FSGR	Servo Valve Position Indication	%
FSR	Fuel Stroke Reference	%
FSRG1OUT	GCV1 Position Command	%
FSRG2OUT	GCV2 Position Command	%
FSRG3OUT	GCV3 Position Command	%
FSRGQOUT	Quat Position Command	%
FTG	Fuel Gas Temperature	%

median, and standard deviation, for each i .

$$DPC_i = \frac{DWATT_t - DWATT_{t-1}}{FAGPM_{i,t} - FAGPM_{i,t-1}} \quad (1)$$

3. Shown in Equation 2, DVC_i represents the derivative of the GCV current feedback, where i represents the respective GCV. DVC_i was then aggregated to obtain the mean, median, and standard deviation, for each i .

$$DVC_i = FAGPM_{i,t} - FAGPM_{i,t-1} \quad (2)$$

4. Shown in Equation 3, DVF_i represents the derivative of the GCV total flow, where i represents the respective GCV. DVF_i was then aggregated to obtain the mean, median, and standard deviation, for each i .

$$DVF_i = FSGPCT_{i,t} - FSGPCT_{i,t-1} \quad (3)$$

5. Shown in Equation 4, DPF_i represents the derivative of DWATT w.r.t the GCV position feedback, where i represents the respective GCV. DPF_i was then aggregated to obtain the mean, median, and standard deviation, for

Table 2. Calculated Features

Feature	Calculation Formula	Units
PM1ERROR	FSRG1OUT - FSGPM1	%
PM2ERROR	FSRG2OUT - FSGPM2	%
PM3ERROR	FSRG3OUT - FSGPM3	%
QUATERERROR	FSRGQOUT - FSGQ	%
DFSGPM1	FSGPM1 _t - FSGPM1 _{t-1}	%
DFSGPM2	FSGPM2 _t - FSGPM2 _{t-1}	%
DFSGPM3	FSGPM3 _t - FSGPM3 _{t-1}	%
DFSGQ	FSGQ _t - FSGQ _{t-1}	%
DFSGR	FSGR _t - FSGR _{t-1}	%
DFAGPM1	FAGPM1 _t - FAGPM1 _{t-1}	A
DFAGPM2	FAGPM2 _t - FAGPM2 _{t-1}	A
DFAGPM3	FAGPM3 _t - FAGPM3 _{t-1}	A
DFAGQ	FAGQ _t - FAGQ _{t-1}	A
DFAGR	FAGR _t - FAGR _{t-1}	A
DFSRPM1_PCT	FSRPM1_PCT _t - FSRPM1_PCT _{t-1}	%
DFSRPM2_PCT	FSRPM2_PCT _t - FSRPM2_PCT _{t-1}	%
DFSRPM3_PCT	FSRPM3_PCT _t - FSRPM3_PCT _{t-1}	%
DFSRQT_PCT	FSRQT_PCT _t - FSRQT_PCT _{t-1}	%
DDWATT	DWATT _t - DWATT _{t-1}	%
DFSR	FSR _t - FSR _{t-1}	%

Table 3. Univariate Feature Selection Results

Parameter	T-Test P-Value
position_error_total	0e+00
position_error_mean	0e+00
position_error_median	0e+00
dcurrent_sd	0e+00
current_sd	0e+00
current_mean	0e+00
current_median	0e+00
asddwattdcurrent_median	0e+00
position_error_sd	0e+00
dfsrdposition_median	0e+00
ddwattposition_median	0e+00
dflowpct_sd	1e-07

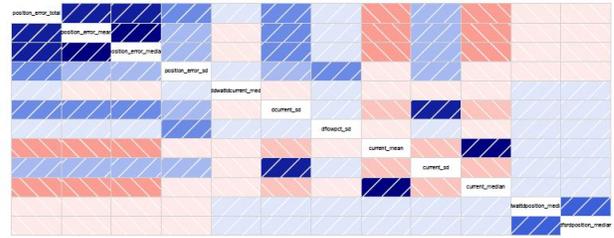


Figure 3. Correlogram of Features.

each i .

$$DPF_i = \frac{DWATT_t - DWATT_{t-1}}{FSGPM_{i,t} - FSGPM_{i,t-1}} \quad (4)$$

- Lastly, shown in Equation 5, DPF_i represents the derivative of the fuel stroke reference w.r.t the GCV position feedback, where i represents the respective GCV. DPF_i was then aggregated to obtain the mean, median, and standard deviation, for each i .

$$DPF_i = \frac{FSR_t - FSR_{t-1}}{FSGPM_{i,t} - FSGPM_{i,t-1}} \quad (5)$$

3. DATA ANALYSIS

The R programming language was selected for the data analysis and model building phase due to the compatibility with HDFS. Data were exported from HDFS and loaded into R as a table of labeled features.

3.1. Applying Data Transformations

Several fields were transformed and filters were applied to aid in the analysis. This was done to transform variables into similar ranges as well as to clean data where the valve was in a relatively fixed position. Additionally, the healthy population included were from units that had a post Jan 1, 2009 COD Date, with data from the first two weeks of operation.

3.2. Feature Selection

3.2.1. Univariate Feature Selection

Univariate feature selection was applied using a two sample t-test to identify a significant difference in means between populations of the healthy and unhealthy GCVs. Features shown in Table 3 yielded p-values less than 0.01 and were removed prior to applying multivariate feature selection.

3.2.2. Correlated Features

As discussed in (Breiman, 2001), features having high correlation can introduce correlation bias into the predictions, therefore features which possessed high correlation (correlation greater than 0.90) were removed. A correlogram generated in R, shown in Figure 3, represents the correlation of features where the darker colors represent higher correlations. Concretely, a darker blue color corresponds to two features being directly correlated and a darker red color corresponds to two features being inversely correlated.

3.2.3. Multivariate Feature Selection

Following the univariate feature selection and removal of highly correlated variables, recursive feature selection was used to further down-select statistically important features. For recursive feature elimination, bootstrap models were developed from the data partitions in a training data-set and

tested on a validation data-set. The Bootstrap models select the set of features that maximize the area under the Receiver Operability Curve (ROC).

Following the multivariate feature selection, the following variables were selected for modeling:

1. Derivative of the current feedback standard deviation
2. Median current feedback
3. Total position error
4. Median position error
5. Standard deviation of the current feedback

4. MODEL SELECTION

As discussed in Section 2, the overall goal was to determine if a GCV is healthy or degraded, thus the model output of instances should admit only binary values (Kotsiantis, Zaharakis, & Pintelas, 2007). Accordingly, classification modeling techniques were selected for this work. This section and the following subsections discuss the process and considerations taken for selecting the *classifier* as well as an overview of the selected classification algorithms.

4.1. System Constraints

Model selection was constrained by the following and listed in order of priority:

4.1.1. Computational Runtime Performance

As discussed in Section 2.1, the upfront goal was to apply this classification technique to the entire GE 7FA gas turbine fleet (more than 600 assets). Additionally, a system constraint exists where the model must execute once per day, regardless of the data resolution. The current system executes a tremendous number of analytics and models daily for diagnostic and prognostic purposes, therefore computational overhead during runtime execution was the most significance constraint for model selection. Certain computational intensive activities could not be minimized during runtime, such as data retrieval and pre-processing, so it was important to minimize computational overhead generated from the model during runtime.

4.1.2. Output Interpretability

The monitoring and diagnostics system raises alerts that are dispositioned by an engineer who is not particularly familiar with the specificities of the classification algorithm. This engineer will view alerts raised by the model and further classify them as being either true or false positives. This can have an adverse impact on the model results by creating inadvertent false negatives, or by increasing the disposition time for the respective engineers. Consequently, it is important for the model to be relatively simple to interpret.

4.1.3. Scalability

Although the goal is to initially apply this model to the entire 7FA gas turbine fleet, it would be advantageous to scale this model and/or technique to additional gas turbine frame-sizes, assuming similar features exist, while minimizing model management costs. Model management overhead refers to model training time, number of models to manage, production model size in memory. Typically, a tradeoff exists between model learning time and number of production models. However, since the models were trained on an HDFS cluster, the cost of having multiple models to maintain greatly outweighed the cost of learning time. Lastly, since the production model execution system supports model compression, this was the least weighted system constraint.

Given the GCV classification problem at hand and the existing system constraints, the following classification algorithms were chosen for comparison:

1. Decision Tree
2. Random Forest
3. C5.0 Algorithm

Other popular pattern recognition classifiers were considered, such as Bayesian classification, Neural Networks, and Support Vector Machines, however they were disqualified due to the output interpretability constraint.

4.2. Decision Tree

Decision trees, also called Classification and Regression Trees (CART) are defined by recursively partitioning the input space, and defining a local model in each resulting region of input space. This can be represented by a tree, with one leaf per region. An example of a simple decision tree using two arbitrary inputs and thresholds to predict color is shown in Figure 4. The first node asks if x_2 is greater than 5. If yes, the second node asks if x_1 is greater than 8. If yes, the color is classified as green and if no, the color is classified as blue. The same methodology follows for the opposite paths. The result of these axis parallel splits is to partition 2d space into four regions (colors), as shown in Figure 4 (Murphy, 2012).

Decision trees are advantageous for several reasons: they are easy to interpret, they can easily handle mixed discrete and continuous inputs, they are insensitive to monotone transformations of the inputs, they perform automatic variable selection, they are relatively robust to outliers, they scale well to large data sets, and they can be modified to handle missing inputs. However, decision trees also have some disadvantages. The primary one is that they do not predict very accurately compared to other kinds of models. This is in part due to the greedy nature of the tree construction algorithm. A related problem is that trees are unstable: small changes to the input data can have large effects on the structure of the tree, due to

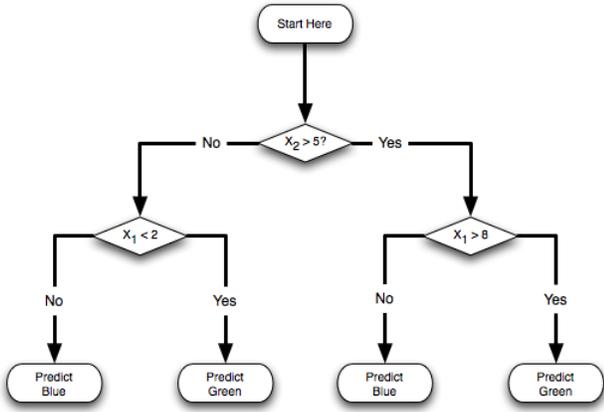


Figure 4. A simple decision tree on two inputs to predict color.

the hierarchical nature of the tree-growing process, causing errors at the top to affect the rest of the tree (Murphy, 2012).

4.3. Random Forest

Random forests are a modified form of decision trees with improvements to reduce the variance of estimates. Random Forests can train M different trees on different subsets of the data, chosen randomly with replacement, and then compute the ensemble shown in Equation 6, where f_m is the m 'th tree. This technique is known as bagging, also known as *bootstrap aggregation*.

$$f(x) = \sum_{m=1}^M \frac{1}{M} f_m(x) \quad (6)$$

Additionally, the technique known as random forests attempts to decorrelate the base learners by learning trees based on a randomly chosen subset of input variables, as well as a randomly chosen subset of data cases. Such models generally have very high predictive accuracy and have been widely used in many applications. Unfortunately, random forests sacrifice interpretability for the higher predictive accuracy (Murphy, 2012). Additionally, for this analysis all of the default settings were used for the Random Forest model creation with no optimization applied, such as pruning or boosting.

4.4. C5.0 Algorithm

The last classification model selected for comparison was the C5.0 algorithm, a more recent version of C4.5 (Quinlan, 1993). The C5.0 algorithm yields similar results to other classification algorithms, such as the Chi-Square Automatic Interaction Detector (CHAID), decision trees, neural networks, Logit classifiers, and discriminant analysis (Coenen, Swinnen, Vanhoof, & Wets, 2000). The C5.0 algorithm differs

from a decision tree in that it converts a standard decision tree into a set of rules, where each rule has an assigned classification label with a corresponding confidence. When this set of decision rules is ran over a number of observations, each observation is classified by the output with the highest confidence. The observation is then assigned the output label with the higher confidence. Although the C5.0 algorithm does not have the predictive accuracy of a random forest, it offers improved interpretability and scalability in that the set of classification model rules are programming language independent.

5. RESULTS

This section evaluates the models considered for comparison in Section 4 on the test data set. A set of evaluation criteria is established, so each model can be compared fairly on the same test data set.

5.1. Evaluation Criteria

Each aforementioned classification algorithm predicts an output for each observation within the test data set. Since this is a binary classification problem, let y represent the actual state of the GCV and \hat{y} represent the predicted state of the GCV, where 0 represents a healthy GCV and 1 represents a degraded GCV. A GCV can be correctly predicted as being degraded (true positive) or healthy (true negative), shown in Equation 7 and Equation 8, respectively (Murphy, 2012).

$$tp = p(\hat{y} = 1 | y = 1) \quad (7)$$

$$tn = p(\hat{y} = 0 | y = 0) \quad (8)$$

Alternatively, a GCV can be incorrectly classified in the two converse scenarios. A GCV can be incorrectly predicted as being degraded (false positive) or a GCV can be incorrectly predicted as being healthy (false negative), shown in Equation 9 and Equation 10, respectively (Murphy, 2012).

$$fp = p(\hat{y} = 1 | y = 0) \quad (9)$$

$$fn = p(\hat{y} = 0 | y = 1) \quad (10)$$

To measure the model effectiveness, *precision* and *recall* were used. For this problem, precision (P) is defined as the ratio of correctly classified, degraded GCVs versus the total number of predicted degraded GCVs. This is shown in Equation 11. Recall (R) is defined as the ratio of correctly degraded GCVs detected versus the total number of actual degraded GCVs.

Table 4. Model Performance Results

Model	Precision	Recall	F_1 Score
Decision Tree	87.6%	87.3%	87.45%
Random Forest	97.4%	95.9%	96.64%
C5.0 Algorithm	94.5%	94.6%	94.55%

$$P = \frac{tp}{tp + fp} \quad (11)$$

$$R = \frac{tp}{tp + fn} \quad (12)$$

There is typically a tradeoff with classification algorithms between precision and recall. For example, if the model has high precision it will typically have low recall, and vice versa. The balanced F measure (a.k.a. F_1) is used to effectively compare the three classification models, which is a single measure that trades off precision versus recall. Concretely, the F measure is the weighted harmonic mean of precision and recall, shown in Equation 13 with β^2 defined in Equation 14, where $\alpha \in [0, 1]$ and thus $\beta^2 \in [0, \infty]$. The *balanced F measure* equally weights precision and recall, which corresponds to making $\alpha = \frac{1}{2}$ or $\beta = 1$. Values of $\beta < 1$ emphasize precision, whereas values of $\beta > 1$ emphasize recall. The F_1 measure was chosen to evaluate overall model performance in comparison to the *accuracy metric* due to the high bias that existed in the data, which is common for anomaly detection problems. From Equation 12, 100% recall can be achieved by classifying all test observations as having a degraded GCV, and therefore 50% arithmetic mean can be achieved by the same process. This strongly suggests that accuracy was an unsuitable measure to use (Manning, Raghavan, & Schütze, 2008).

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \quad (13)$$

$$\beta^2 = \frac{1 - \alpha}{\alpha} \quad (14)$$

5.2. Model Comparison Results

The three models were executed on the test set and the performance results are shown in Table 4. The target F_1 score to achieve was 90%. The Decision Tree did not meet the target F_1 score and was disqualified. Both the Random Forest and C5.0 models exceeded the F_1 threshold, having scores of 96.64% and 94.55%, respectively. The run-time performance of the two algorithms were generally equivalent, however the interpretability and scalability of the C5.0 algorithm are much higher than that of the Random Forest. The C5.0 model was selected to move forward into production.

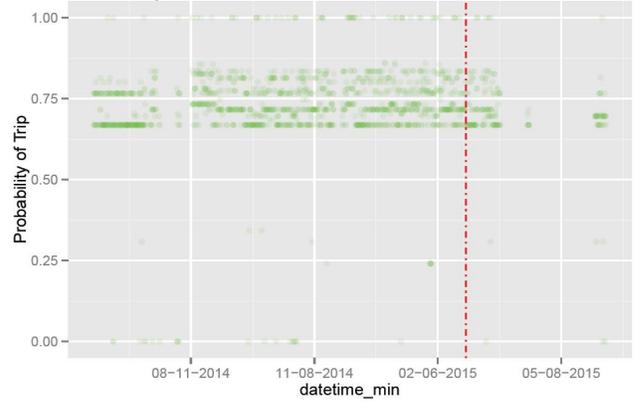


Figure 5. Gas Turbine GCV Probability of Trip vs Time.

5.3. Production Validation

After the C5.0 model was implemented in a production environment, the classification algorithm was able to provide early warning signs of GCV degradation approximately six months prior to a trip occurring. Shown in Figure 5, the vertical red dotted line represents that the respective gas turbine tripped due to a degraded GCV during late February, 2015. The green dots on the plot represents the confidence of a trip output from the C5.0 model, where the darker dots represent high density in trip confidence. Following the gas turbine trip, the GCV was actuated and immediately restarted to reduce unplanned downtime. This is what causes the C5.0 model to still yield a high trip confidence after the trip occurred. The gas turbine was finally shutdown during March, 2015, where the GCV was serviced. This is shown on Figure 5 by the decrease in trip confidence during the associated timeframe.

Figure 5 shows that the trip confidence to be approximately 75%, months prior to the trip actually occurring. Placing a static threshold on the probability of a trip would lead to an abundant number of false alarms. Instead, the trip confidence was compared to the confidence of a healthy GCV and if the trip confidence exceeded the confidence that the GCV was healthy, for a tunable threshold of time, an alert was raised to notify the customer. Due to the long-term degradation behavior of GCVs, this persistence technique was optimal for maximizing alert precision.

6. CONCLUSIONS

Being that GCVs are one of the leading causes of gas turbine trips and failed starts, the diagnosis of a degraded GCV on a gas turbine is critical to reduce unplanned maintenance and downtime. By simply placing thresholds around the current feedback and/or the slew rate of the GCV is not a feasible means of detecting hydraulic valve degradation. This methodology requires high resolution data to capture these features, which adds a significant amount of computational overhead and an abundant number of false positives. If lower

resolution data is used, the sampling frequency may yield a high amount of false negatives. Furthermore, this technique does not provide a solution to monitor the long-term degradation of the GCV.

This work proposes a purely data-driven solution to efficiently detect and classify degraded servo-actuated hydraulic valves installed in a gas turbine fuel delivery system. Additionally, the proposed solution provides prognostic capabilities to alert operators up to four months prior to a GCV related trip, without customers installing additional instrumentation. Nearly eighty potential features were calculated from the limited instrumentation using a combination of domain expertise, statistical feature extraction, and insight gains from prior physics-based simulations. Based on the problem at hand and system constraints, three classification algorithms were analyzed on more than 500 gas turbines, each gas turbine having more than a year and half of data. The three algorithms were measured by their respective precision, recall, and F_1 score. The C5.0 algorithm was selected as the optimal classifier and was implemented in a production environment, where it currently has a precision value greater than 90%.

Future work could investigate how to scale this methodology to additional gas turbine frame-sizes. This could involve incorporating additional frame sizes into the original data set and re-training the model or creating a new, independent model following this same methodology. Also, future work could expand this technique to other types of hydraulic valves with limited instrumentation, such as the SRV. Lastly, improving the instrumentation on a GCV system to expand the feature space would be advantageous to improving model fidelity. Such features may include the differential pressure across the valve and the hydraulic fluid temperature.

NOMENCLATURE

A	Amperes
MW	Megawatt
$psig$	Pounds per square inch gauge
$^{\circ}F$	degrees Fahrenheit

REFERENCES

- Bernstein, H. L., & Allen, J. M. (1992). Analysis of cracked gas turbine blades. *ASME Journal of Engineering for Gas Turbines and Power*, 114(2), 293-301.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32.
- Carter, T. J. (2005). Common failures in gas turbine blades. *Engineering Failure Analysis*, 12(2), 237-247.
- Coenen, F., Swinnen, G., Vanhoof, K., & Wets, G. (2000). The improvement of response modeling: combining rule-induction and case-based reasoning. *Expert Systems with Applications*, 18(4), 307-313.
- Davis, L. B. (1996). Dry low no_x combustion systems for ge heavy-duty gas turbines. *ASME 1996 International Gas Turbine and Aeroengine Congress and Exhibition*, 3.
- Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging Artificial Intelligence Applications in Computer Engineering*.
- Macaluso, A. (2016). Prognostic and health management system for hydraulic servoactuators for helicopters main and tail rotor. *Third European Conference of the Prognostics and Health Management Society*, 7(76).
- Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.
- Mornacchi, A., Vachtsevanos, G., & Jacazio, G. (2015). Prognostics and health management of an electrohydraulic servo actuator. *Annual Conference of the Prognostics and Health Management Society*.
- Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.
- Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Ravi, Y., Pandey, A., & Jammu, V. (2010). Prediction of gas turbine trip due to electro hydraulic control valve system failures. *ASME Turbo Expo 2010: Power for Land, Sea, and Air*, 3(2), 299-306.

BIOGRAPHIES



James J. D'Amato is a Senior Data & Analytics Scientist with GE Power. Originally from Long Island, NY, he holds a Bachelor of Science in Electrical Engineering from the Georgia Institute of Technology and a Master of Science in Computer Engineering from the University of Cincinnati. Prior to joining GE, he served four honorable years with the United States Marine Corps, holding a variety of leadership positions. Currently, he supports GE Power Services where he designs and implements advanced data-driven anomaly detection techniques for industrial power-system assets. His interests include incorporating his prior knowledge towards the development and implementation of advanced data-driven techniques to yield safe, cost-efficient, and environmentally friendly solutions.



John Patanian is Principal Engineer, analytics for GE power and has over 20 years experience in software development, machinery diagnostics, product management, controls optimization, and thermodynamic performance. He holds a masters degree in Computer Science from the University of Washington and a Bachelor's degree in Mechanical Engineering from Rensselaer Polytechnic Institute. He holds two U.S. Patents and served in the ASME PTC46 committee on performance testing of Combined Cycle power plants