A Data-Driven Approach for on-line Gas Turbine Combustion Monitoring using Classification Models

Carmine Allegorico[1], Valerio Mantini[2]

[1,2]*General Electric Oil and Gas, Florence, 50141, Italy*

allegorico.carmine@ge.com
valerio.mantini@ge.com

ABSTRACT

Given the critical nature of Gas Turbines in most industrial plants, it is a high priority to find ways of reducing maintenance costs and increasing the availability. Quickly detecting and identifying combustion anomalies enables the choice of an appropriate recovery strategy, potentially mitigating the consequences of unscheduled down time and increased maintenance costs. Monitoring the Exhaust Gas Temperature (EGT) profiles is a good means of detecting combustion problems: plugged nozzles and/or combustor and transition piece failures will always result in distorted exhaust gas temperature patterns. However the conventional monitoring systems do not allow robust discrimination between instrumental failures and real gas turbine issues; furthermore weak diagnostic methods can be source of numerous false alarms.

In this paper, we investigate the problem of monitoring the combustion chambers of a gas turbine and we attempt to address this issue by introducing a strategy for automatic and efficient patterns recognition by using Machine Learning Classification algorithms. Some historical events have been firstly retrieved and analyzed to discover which features are useful for classification. Based on the observations, two multiclass classification algorithms, one based on logistic regression, the other on Artificial Neural Networks (ANN), have been developed. Finally, real-world datasets have been used to benchmark the performance of the proposed algorithms against a traditional physics-based approach.

1. INTRODUCTION

Today industrial gas turbines are one of the most widely-used prime movers for power generation and mechanical drive applications. In the Oil&Gas field these engines are often used to drive compression trains (for example in gas pumping or injection stations or in natural gas liquefaction plants) and to provide power for the plant.

Maintenance costs and availability are two of the most important concerns to a heavy-duty gas turbine equipment owner. Gas turbines have to be built and operated with higher availability, reliability, and performance in order to ensure the customer with sufficient operating revenues and minimal fuel costs. Therefore, Remote Monitoring & Diagnostics (RM&D) of equipment like heavy duty gas turbine has become increasingly important and popular in the industry since it's considered a critical process in preventing costly unplanned maintenance and secondary damage.

To achieve this goal, a large number of critical parameters such as engine vibration, bearing temperature, combustion profile, etc. are continuously acquired to detect any changes in the normal operating conditions of the gas turbine engine. This large number of operational data from the everyday operation of a gas turbine is usually collected and analyzed as soon as new data sets arrive in the monitoring center. Anomaly detection rules and models are designed to scan through the data and notify the monitoring and diagnostic engineers, if any novelties or emerging problems are detected.

For example every day of the year, the RM&D center of General Electric Company in Florence (GE Oil&Gas), Italy collects more than 3,850 operating hours of data from a fleet of more than 700 globally installed equipment (gas turbines, compressors, steam turbines and electric generator assets). More than 70,000 signals are processed by automatic diagnostic rules and about 2,300 recommendations per year are sent to customers. Therefore in recent years, in parallel with the operational diagnostic service, it has become increasingly important the challenge of transforming big data into knowledge (Jlang & Foster, 2013) and to detect emerging problems at nearly real-time (early warning) with the development of advanced analytics. In a great number of industrial applications, this continuous supervision of critical parameters is driving the gradual transition from systematic maintenance to conditional maintenance strategies (Vachtsevanos, Lewis, Roemer, Hess, and WU, 2006).

2. COMBUSTION MONITORING

The diagnosis of any malfunction of the combustion system of a gas turbine is of great importance for long term engine reliability and availability. Main causes of damage of hot-section components are imbalanced fuel distribution and combustion instabilities.

Some of the common problems experienced in gas turbines operation are: random re-ignition, combustor blowout, abnormal combustion dynamics, and non-compliant emissions. Modern dry low NOx combustors can target very low emissions levels, but need to operate within very narrow equivalence ratio. Premixed combustors are often susceptible to thermoacoustic combustion instability, which can lead to large pressure oscillations in the combustor and decreased durability of components.

Other causes of combustion issues are clogged or loose fuel nozzles, which may lead to severe burning problems. Abnormal fuel mass distribution among nozzles may cause high emissions of either NOx (due to hot spots in the combustion zone) or CO and unburned hydrocarbons (due to cold spots and poor mixing or atomization). Those hot spots reduce the time taken for failure in creep (phenomenon of plastic deformation) of the combustion liners, transition pieces, turbine nozzles and blades. In fact creep life of metal components in the hot section of a gas turbine is extremely sensitive to metal temperature.

The consequences of hot-section component failures caused by overheating might be quite costly. In extreme cases, combustion liner failures can allow hot flames to impinge on the turbine pressure casing, which can result in catastrophic combustion casing failure (Figure 1). Even before casing failure occurs, broken pieces of the liner can pass into the expander section and cause extensive blade damage.

Monitoring the gas turbine exhaust temperature spread via thermocouples mounted at the gas turbine exhaust section (i.e. maximum - minimum) is a good means of detecting combustion problems. In fact, almost all gas turbine control systems monitor this parameter and issue an alarm when it reaches an OEM-specified value. However most modern diagnostic systems often do not display expected exhaust gas turbine spread profiles (EGT spread) and do not figure out the source of the high-temperature spread. Moreover many false alarms are often triggered as a result of instrumental problems.

In this paper, we discuss the application of a pattern recognition technique to the monitoring of the exhaust gas turbine temperature profile. Although physical insight is without any doubt an important step to enhance knowledge of the processes within the combustion chamber, large datasets can also be exploited with data-mining techniques based on black box models, such as classifiers or artificial neural networks (Hannes, Deneve, Vanderhaegen, & Museur, 2009).

Figure 1. Broken liner as the result of cracks propagation

The data-driven approach to fault diagnosis and prognosis is usually preferred when system models are not available or not robust enough (e.g., when the physics underlying is too complex to be modeled), but instead system monitoring data is available (Namburu, Azam, Luo, Choi, & Pattipati, 2007).

The key challenge in implementing this kind of approach is developing an algorithm that can flag anomalies without also sending out false alarms when something else changes such as engine operating conditions. Pure data-driven modeling techniques work well if sufficient labeled data are available. However in real-world applications like in gas turbine monitoring, obtaining sufficient labeled data is labor-intensive, if ever possible. In particular, true positive cases might be sparse or noisy and using small set of labeled data may cause model over-fitting or ill-formed model representation (Yan, Yu, Sherbahn, and Brahmakshatriya, 2013).

In this paper, an anomaly detection method based on classifiers technology is discussed in detail and implemented on E-class gas turbines. These black box models, trained on historical data (training set), are used to detect the presence of anomaly patterns in unseen data of the EGT profile (test set). These specific signatures not only can alert the operator to a possible problem, but they also identify its severity and can guide in understanding the possible root cause.

3. CLASSIFICATION

In machine learning and statistics, classification is the problem of identifying to which of a set of categories a new observation belongs, on the basis of a training set of data containing observations whose category membership is

known. Example of classification would be to predict whether a patient has a given disease or not, classifying a given email as "spam" or "non-spam", an online transactions as fraudulent or not, etc. It's worth noting that the response variable *y* is qualitative instead of quantitative. All these cases above are examples of binary classification problems because the variable *y* that we'd like to predict admits only two possible outcomes (usually coded as "0" or "1"), but the same concept can be extended to multi-class cases to deal with situations where the outcome can have three or more possible types (e.g., "disease A" vs. "disease B" vs. "disease C").

There are many possible classification techniques, or classifiers, that one might use to predict a qualitative response. Some of these are: logistic regression, Artificial Neural Networks, K-nearest neighbors, decision tree and Support Vector Machines (James, Witten, Hastie, and Tibshirani, 2013).

In this work, logistic regression and artificial neural networks techniques are investigated. Today logistic regression is one of the most popular and most widely used learning algorithms thanks to the interpretability of model parameters and ease of use. On the other hand, neural networks can be seen as nonlinear generalizations of logistic regression, and thus they are considered more flexible algorithms (Dreiseitl & Ohno-Machado, 2002).

3.1. LOGISTIC REGRESSION

In a binary classification problem, where the response *y* falls into one of two categories, 0 or 1, logistic regression models the probability that *y* belongs to a particular category. The surface that partitions the vector space into two sets, one for each class, is called decision boundary.

The function that satisfies the property that a prediction is between 0 and 1 is the hypothesis function $0 \leq h_\theta(x) \leq 1$, defined as $h_\theta(x) = g(\theta^T x)$, where the function g is the logistic or sigmoid function $g(z) = \frac{1}{1+e^{-z}}$, that takes the shape of the S-curve shown in Figure 2 for values of *z* in the range of real numbers from $-\infty$ to $+\infty$. Putting these two equations together, we obtain an alternative form of the hypothesis function.

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}} \qquad (1)$$

The output value of the hypothesis function is the estimated probability that the variable *y* is equal to 1 on a new input example *x*.

Suppose that the hypothesis output is 0.7, the interpretation is that for a patient with features *x*, the patient has a 70% chance of having a specific disease. More formally we can write this as $h_\theta(x) = P(y = 1|x; \theta)$ probability that *y* = 1, given feature *x*, parameterized by θ.

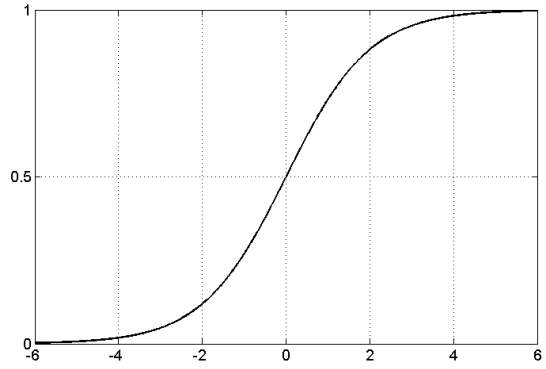

Figure 2. Sigmoid function

The parameter vector θ is the vector of unknown linear regression coefficients of the *n*th order polynomial $\theta_0 x_0 + \theta_1 x_1 + ... + \theta_n x_n = \theta^T x$. High-dimensional vector can be used in non-linear problems to get more complex decision boundary, but the model will be more susceptible to overfitting, which means that it may fit the training set very well, but fail to generalize to new examples.

In every supervised learning problem, a training set of *m* training examples is required

$$\left\{\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \cdots, \left(x^{(m)}, y^{(m)}\right)\right\}$$

where each example is represented by an *N+1* dimensional feature vector *x*, and its associated label *y* can be either 0 or 1.

$$x \in \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_n \end{bmatrix} \; x_0 = 1, \qquad y \in \{0,1\}$$

The training process of a classifier involves finding the best parameter θ vector for the logistic regression cost function $J(\theta)$, given the dataset of *x* and *y* values. This optimization problem consists in minimizing the sum of the square difference between the output of the hypothesis $h_\theta(x)$ and class label *y*, which is finding parameters θ that minimize the function:

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^i) - y^i)^2 + \lambda \sum_{j=1}^{n} \theta_j^2\right] \qquad (2)$$

where λ is the regularization parameter. This optimization problem can be solved with any standard numerical optimization algorithm, like the gradient descent or more advanced methods.

3.2. ARTIFICIAL NEURAL NETWORKS

In computer science and related fields, Artificial Neural Networks are computational models inspired by the neural structure of the brain that are capable of machine learning and

pattern recognition. Due to their high connectivity and parallelism, ANNs are able to link, in a non-linear way, a multi-dimensional input space with a multi-dimensional output space, allowing very high computational speed (Haykin, 1999).

The neural network architecture used in this paper for gas turbine combustion monitoring is the multilayer feedforward neural network (see Figure 3), in which the artificial neurons are arranged in layers, and the neurons of a layer are linked to all the neurons of the following layer, while, there are no links among neurons of the same layer. The input layer consists of a set of nodes (where no data processing occurs) equal to the number of ANN inputs, while the number of neurons in the output layer is equal to the number of ANN outputs.

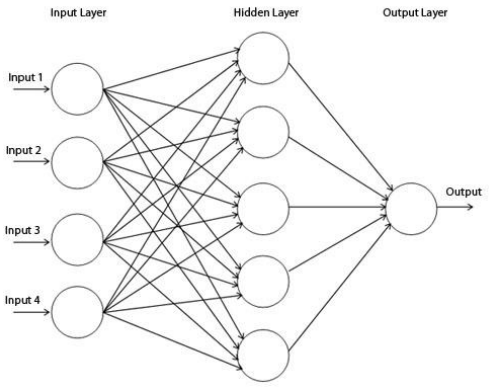

Figure 3. Artificial Neural Network architecture

Feedforward networks often have one or more hidden layers of sigmoid neurons also called activation functions (Eq. (1)) followed by an output layer of linear or sigmoid neurons. Multiple layers of neurons with nonlinear transfer functions allow the network to learn nonlinear relationships between input and output vectors. The linear output layer is most often used for function fitting (or nonlinear regression) problems, while sigmoid transfer function is used to constrain the outputs of a network (such as between 0 and 1). This is the case when the network is used for pattern recognition problems (in which a decision is being made by the network). All the calculations are performed in hidden and output layers. In particular, if x_{ij} is the *i*th input of the *j*th neuron and w_{ij} is the weight of x_{ij}, the neuron output y_j is determined by means of an activation function *f* applied to the weighted sum of the inputs plus the bias *b*.

$$y_i = f\left(\sum_{i=1}^{m_j} w_{ij}x_{ij} + b\right) \quad , \quad j = 1, \dots, n_N \qquad (3)$$

The process of training a neural network involves tuning the values of the weights and biases of the network to optimize network performance, which generally is the mean square error *mse*, namely the average squared error between the network outputs *y* and the target outputs *t*. It is defined as follows:

$$F = mse = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - y_i)^2 \qquad (4)$$

For training multilayer feedforward networks, any standard numerical optimization algorithm can be used to optimize the performance function, but there are a few key ones that have shown excellent performance for neural network training. These optimization methods use either the gradient of the network performance with respect to the network weights, or the Jacobian of the network errors with respect to the weights. The gradient and the Jacobian are calculated using a technique called backpropagation algorithm, which involves performing computations backward through the network.

Although the functional forms for logistic regression and artificial neural network models are quite different, a network without a hidden layer is actually identical to a logistic regression model if the logistic (sigmoidal) activation function is used.

Since artificial neural networks are aggregations of nonlinear functions (neurons), in classification problems ANNs are able to represent complex models that form non-linear hypotheses, differently from logistic regression that is only a linear classifier. The type of decision boundary that the network can learn is determined by the number of hidden layers.

4. MODEL DEVELOPMENT

The first step towards the development of a classifier for gas turbine combustion monitoring is the definition of the categories to be classified.

Polar plot of EGT profiles is often used in diagnostics to identify uneven temperature distributions. The calculation of the exhaust swirl angle is then used to map temperatures back to the originating combustion chamber. Based on experience, the 4 classes of Figure 4 have been identified, each of which is characterized by a specific temperature distribution in the polar plot.

For example, in a fault–free case (Class 1) the exhaust temperature profile is expected to be quite regular; it will be peaked on the abnormal thermocouple in presence of a sensor anomaly (Class 2) and asymmetric with more than one thermocouple far from the average temperature in the case of a cold (Class 3) or hot spot (Class 4).

The underlying idea in this paper is that a classification model can be trained on real cases of normal behavior, sensor anomaly, cold spot and hot spot to recognize their specific patterns when new data are presented.

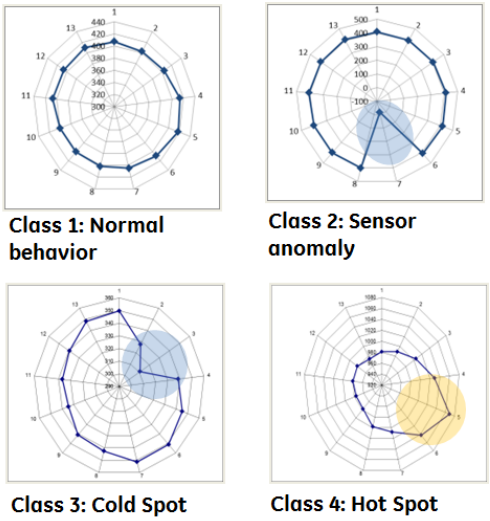

Figure 4. Polar plots of exhaust gas temperature profiles

This would allow greater performance than traditional diagnostic systems that are simply based on the monitoring of the exhaust spread.

4.1. REGULARIZED LOGISTIC REGRESSION TRAINING AND VALIDATION

For the 4-classes classification problem presented here, a multi-class classification algorithm called "one-vs-all" is implemented. This algorithm handles the training set as 4 separate binary classification problems, where each class *i* is separated from the remaining ones. In other words the logistic regression classifier $h_\theta^{(i)}(x)$ is trained for each class *i* to predict the probability that *y*=*i*, $h_\theta^{(i)} = P(y = i|x;\theta)$. To make the final prediction, the 4 classifiers are run simultaneously on the input *x*, and the class with the highest probability $\max_i h_\theta^{(i)}(x)$ is then selected.

For the creation of the training dataset, historical events ground truth data have been primarily collected from RM&D issue database. Operating data of about 150 heavy-duty gas turbines in a period of 2 years operation are available for the analysis. Since we focus on anomaly detection algorithm, these data include both abnormal units and normal units, which are referred as positive and negative cases respectively. Secondly, time series of classifier input data *x* of some historical cases are extracted from the data historian and analyzed to generate the training dataset as explained below.

The most reliable way to get a high performance machine learning system is to take a low bias learning algorithm and to train it on a massive training set. However in real-world applications true positive cases are sparse and only small labeled training set are available.

An artificial data synthesis method can be used to create new data from scratch or to amplify a given dataset. The second case has been put in place to turn the relative small training set available into a larger training set. For intellectual property protection, we are not allowed to give details and how this procedure was carried out and the number of feature *x* considered for the model.

Through the procedure explained above, a dataset of 11000 samples was generated and divided in three subsets for training, validation and test with following ratio 0.7, 0.15 and 0.15 respectively.

A first order polynomial was too simple for the data and resulted in underfitting (high bias), so a 2° order polynomial was used. The regularization parameter λ can significantly affect the results of the polynomial regression. In particular, a model without regularization (λ = 0) fits the training set well, but does not generalize. Conversely, a model with too much regularization (λ = 100) does not fit the training set and testing set well. A good choice of λ can provide a good fit to the data.

We used the Matlab® *fminunc* optimization solver to optimize the cost function $J_{train}(\theta)$ with parameters θ on the training dataset. Concretely we passed to *fminunc* function the following inputs:

- The initial values of the parameters to be optimized
- A function that, when given the training set and a particular θ, computes the logistic regression cost and gradient with respect to θ for the dataset (x, y). This allows *fminunc* to use the gradient when minimizing the function.

For the regularized logistic regression, the Eq. (2) of the cost function becomes

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}log\left(h_\theta\left(x^{(i)}\right)\right) - \left(1-y^{(i)}\right)log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2 \quad (5)$$

Correspondingly, the partial derivative of regularized logistic regression cost for θ_j is defined as

$$\frac{\partial J(\theta)}{\partial \theta_0} = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)} \qquad for\ j = 0$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \left(\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)x_j^{(i)}\right) + \frac{\lambda}{m}\theta_j \qquad for\ j \geq 1$$

After that the optimal values of θ were found, the model was then validated on the cross-validation dataset computing the cost function $J_{CV}(\theta)$ for different values of λ. We found that, for the dataset considered, λ=8 is the value that works best in terms of having a small cross-validation and test set error (Figure 5).

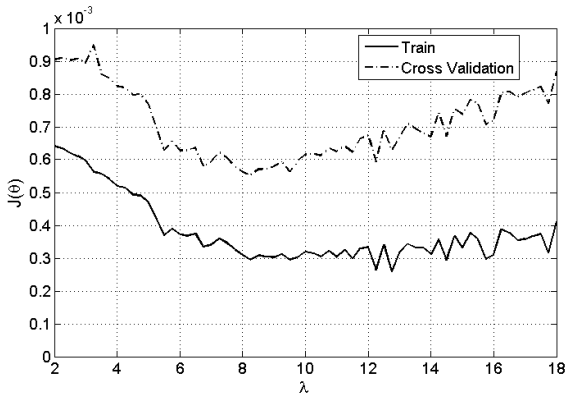

Figure 5. Selecting λ using a cross-validation set

Finally the model's performance has been evaluated on the test set, since it was not used in any part of training (that is, it was neither used to select the λ parameters, nor to learn the model parameters θ). The test error for λ = 8 was $J_{test}(\theta)$ = 0.64e-04 with 100% accuracy, that is the percentage of examples that the classifier got correct.

After learning the parameters θ, to help visualize the model learned by the classifier, we have plotted the non-linear decision boundary that separates the positive and negative examples in a 3-dimensional space.

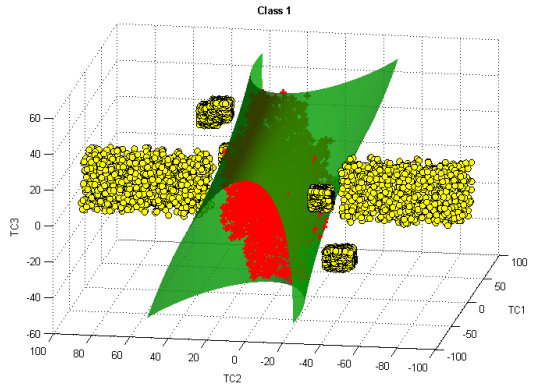

Figure 6. Class 1 classifier decision boundaries

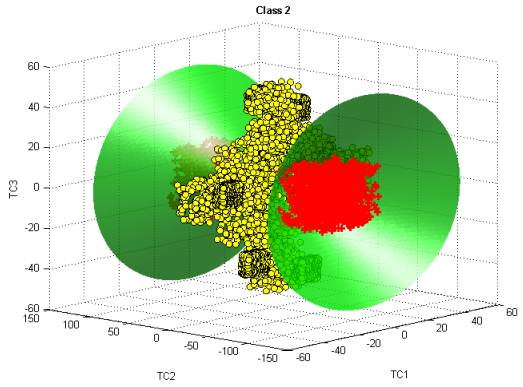

Figure 7. Class 2 classifier decision boundaries

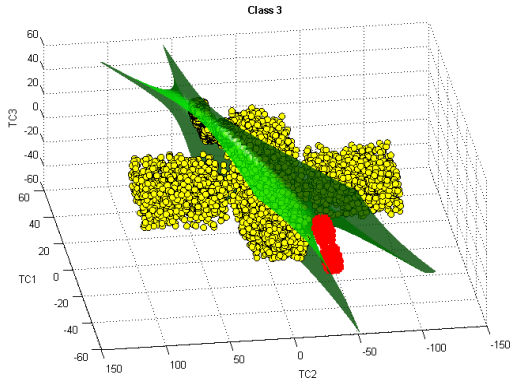

Figure 8. Class 3 classifier decision boundaries

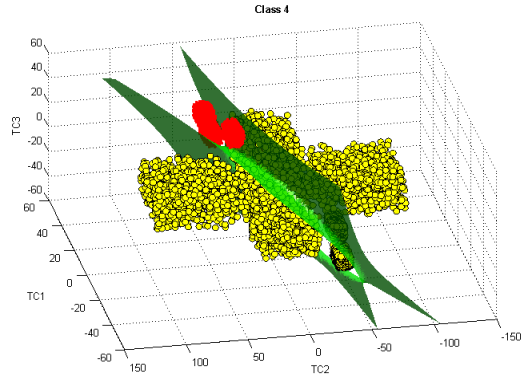

Figure 9. Class 4 classifier decision boundaries

In Figure 6 to Figure 9, the decision boundary for each of the 4 assigned classes is shown in green. The red dots are the positive example, while the yellows ones are the negative examples.

4.2. NEURAL NETWORK TRAINING AND VALIDATION

The same dataset of section 4.1 was used to build a neural network based classifier. The architecture selected for the network is the feed-forward with sigmoid transfer functions in both hidden and output layers. The network has four output neurons, because there are four categories associated with each input vector, thus each output neuron represents a category. When an input vector x of the appropriate category is applied to the network, the corresponding neuron should produce a 1 and the other neurons should output 0. The influence of the number of neurons in the hidden layer was evaluated by comparing the response of different ANNs with different numbers of hidden neurons. Ten neurons in the hidden layer were considered an acceptable compromise between ANN accuracy and computational time required for the training. Due to the high number of patterns used for the training, the *overfitting* phenomenon (when the model learns the training data so well that it loses the ability to generalize) is unlikely to happen.

The Matlab® Neural Network Toolbox was used for the training process. The best validation performance was found at iteration 93 with 100% of cases perfectly classified.

5. RESULTS ON REAL WORLD DATASETS

As explained in section 4.1, the training dataset was obtained through an artificial data synthesis method on the observation of some relevant cases. For the validation on real data, time series with a one-minute sampling rate are used. Datasets prepared have duration of about one week before and after the event for positive cases and total length of 5 months for negative cases.

Starting from combustion labeled cases stored in the RM&D issue database and other past events notified to customers, 5 datasets, one for each class, have been identified (25 fault-free cases, 25 cases of anomalies, 25 sensor failures/out of range, 25 cold spots and 4 hot spots). The hot spots cases are less numerous because they have a lower probability of occurrence. An additional class has been added to those seen previously in this paper, this new class contains out of range anomalies, which in most cases are broken probes with unreliable or full scale values. These cases are filtered by the algorithm without passing through classifier and must be correctly detected by the diagnostic system.

A criterion to evaluate performance of classification problem is the *contingency table* that contains information about the outcome of the classifier compared with the target, giving information about the true or false positives and true or false negatives. The True Positive is a Target correctly identified whereas the True Negative is a Target correctly rejected. The False Positive, also known as *Type I error*, is a test result that is read as positive when it is really negative, whereas the False Negative, also known as *Type II error*, is a test result read as negative when it is really positive.

Table 1: Sensor Failure Contingency Table

Sensor Failure		Target (Gold Standard)	
		Positive	Negative
Test Outcome	Positive	True Positive 25	False Positive 0
	Negative	False Negative 0	True Negative 25

Table 2: Anomaly Contingency Table

Anomaly		Target (Gold Standard)	
		Positive	Negative
Test Outcome	Positive	True Positive 24	False Positive 0
	Negative	False Negative 1	True Negative 25

Table 3: Cold Spot Contingency Table

Cold Spot		Target (Gold Standard)	
		Positive	Negative
Test Outcome	Positive	True Positive 25	False Positive 0
	Negative	False Negative 0	True Negative 25

Table 4: Hot Spot Contingency Table

Hot Spot		Target (Gold Standard)	
		Positive	Negative
Test Outcome	Positive	True Positive 4	False Positive 0
	Negative	False Negative 0	True Negative 25

The contingency tables from Table 1 to Table **4** summarize the results obtained with the classifier based on logistic regression for each class. It's evident that the performance of the classifier is very satisfactory, since it fails to predict only one case from the anomaly test set, whereas the other predictions are correct. These results are also summarized in the confusion matrix in Table 5.

A benchmarking with other two algorithms has been done using the same labeled cases. In particular we considered the Artificial Neural Network classifier described before and a Physics-Based (P-B) approach. The latter is a proprietary algorithm based on the monitoring of the EGT spread and is reinforced with the adjacency check between the coldest/hottest thermocouple and the second coldest/hottest thermocouple for the detection of cold and hot spot anomalies.

Table 5: Confusion Matrix

		Target Class					
		Sensor Failure	**Normal**	**Anomaly**	**Cold Spot**	**Hot Spot**	
Output Class	**Sensor Failure**	25	0	0	0	0	100%
	Normal	0	25	1	0	0	96.2 %
	Anomaly	0	0	24	0	0	100%
	Cold Spot	0	0	0	25	0	100%
	Hot Spot	0	0	0	0	4	100%
		100 %	100 %	96 %	100 %	100 %	99%

In order to compare the three different algorithms, it is necessary to define some appropriate metrics. Starting from the contingency table explained above, it is possible to derive various indicators like *precision* and *recall*.

In binary classification, precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved, while recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database (Labatut, Cherifi, 2011).

So, precision and recall are defined as:

$$precision = \frac{true\ positive}{true\ positive + false\ positive}$$

$$recall = \frac{true\ positive}{true\ positive + false\ negative}$$

From the definition above it turns out that a good classifier must have high precision and high recall. In fact, a low precision classifier produces high number of false alarms, whereas a low recall classifier gets a high number of missing alarms.

Analyzing the results obtained in Table 6 and Table 7, the logistic regression classifier shows better performance in terms of precision and recall compared to the other two algorithms used for the benchmark.

Table 6: Precision

Class	Precision LR	Precision ANN	Precision P-B Rule
Sensor Failure	100%	100%	100%
Anomaly	100%	100%	88.88%
Cold Spot	100%	100%	100%
Hot Spot	100%	100%	100%
TOTAL	**100%**	**100%**	**96.96%**

Table 7: Recall

Class	Recall LR	Recall ANN	Recall P-B Rule
Sensor Failure	100%	100%	96%
Anomaly	96%	80 %	64%
Cold Spot	100%	100%	80%
Hot Spot	100%	75 %	100%
TOTAL	**98.73%**	**92.40%**	**81.01%**

The ANN based classifier generates 6 false negative, failing to predict 5 test cases from the Anomaly dataset and one case from the Hot Spot dataset, without generating any false positive prediction. This result decreases the recall relative to the two classes with the false negative without impacting the precision metrics.

The P-B rule have a very low recall metric due to 5 false negatives in Cold Spot dataset, one in Sensor Failure dataset and 9 in Anomaly dataset. This rule also produces 2 false positives in Anomaly dataset affecting also the precision of the rule.

Another useful metric to compare the three algorithms is the F1-Score. This score weights recall and precision equally, and a good retrieval algorithm will maximize both precision and recall simultaneously. Thus moderately good performance on both will be favored over extremely good performance on one and poor performance on the other. F1-score is defined as:

$$F1 = 2\frac{precision \cdot recall}{precision + recall}$$

Table 8 confirms the performance results previously found. Despite the high accuracy of the ANN obtained in the training phase, the logistic regression showed a greater ability to place the unseen data in the right classes. This result can be probably explained by the fact that, given the strategy used for the training set generation, the logistic regression has a greater ability to fit the nature of the problem thanks to the analytical definition of its decision boundaries.

Table 8: F1-Score

Class	F1-score LR	F1-score ANN	F1-score P-B Rule
Sensor Failure	1	1	0.9795
Anomaly	0.9795	0.8888	0.7441
Cold Spot	1	1	0.8888
Hot Spot	1	0.8571	1
TOTAL	**0.9936**	**0.9605**	**0.8827**

6. CONCLUSIONS

The diagnosis of any malfunction of the combustion system of a gas turbine is critical in preventing costly unplanned maintenance and in reducing life-cycle costs of power plant operations. Monitoring the exhaust temperature spread is a good means of detecting combustion problems. However conventional monitoring systems do not allow robust discrimination between instrumental failures and real combustion issues; furthermore weak diagnostic methods can be source of numerous false alarms.

In this research, a Machine Learning technique, based on classification technology, is proposed to efficiently recognize anomaly patterns of common combustion problems. These specific signatures not only can alert the operator to a possible problem, but they also identify its severity and can guide in understanding the possible root cause. Two multiclass classification algorithms, one based on logistic regression, the other on artificial neural networks, have been trained on labeled patterns extracted from real cases of normal behavior, sensor anomaly, cold spot and hot spot collected in the RM&D center of Florence. An artificial data synthesis method has been used to amplify the original dataset, since only small labeled training set is available. After training process, the developed classification models and an additional physics-based algorithm have been tested on real combustion cases.

The final performance metrics pointed out better results for both data-driven methods compared to the physics-based model. The best performance, both in accuracy and recall, was achieved by the logistic regression algorithm. The ANN based classifier, despite having excellent accuracy, generated 6 false negative resulting in a lower recall.

Future research could investigate how to enhance the insight into the complex combustion system behavior relying not only on EGT profiles. For instance, multi-sensor fusion may provide robust and complete description of the combustion process combining information coming from additional sensors, such as combustion dynamic pressures and pressure ratio across the combustion fuel nozzles.

REFERENCES

Vachtsevanos, G., Lewis, F., Roemer, M., Hess, A., & WU, B., (2006). *Intelligent fault diagnosis and prognosis for engineering systems.* Hoboken, NJ: John Wiley & Sons, Inc.

James, G., Witten, D., Hastie, T., & Tibshirani, R., (2013). *An Introduction to Statistical Learning.* NY, USA: Springer Science + Business Media.

Hastie, T., Tibshirani, R., & Friedman, J., (2001). *The Elements of Statistical Learning.* NY, USA: Springer Science + Business Media.

Jlang, X., & Foster, C., (2013). Remote thermal performance monitoring and diagnostics – Turning data into knowledge. *Proceedings of the ASME 2013 Power Conference*, July 29 – August 1, Boston, Massachusetts, USA.

Hannes, L., Deneve, M., Vanderhaegen, E., Museur, T., (2009). Combustion dynamics data mining techniques: a way to gain enhanced insight in the combustion process of fielded gas turbines. *Proceedings of ASME Turbo Expo 2009: Power for Land, Sea and Air*, June 8-12, 2009, Orlando, Florida, USA.

Namburu, S. M., Azam, M. S., Luo, J., Choi, K., & Pattipati, K. R. (2007). Data-driven modeling, fault diagnosis, and optimal sensor selection for HVAC chillers. *IEEE transactions on automation science and engineering*, vol. 4, no. 3. doi: 10.1109/TASE.2006.888053.

Yan, W., Yu, L., Sherbahn, J., and Brahmakshatriya, U., (2013). On Optimizing Anomaly Detection Rules for Gas Turbine Health Monitoring. *Proceedings of the Annual Conference of the Prognostics and Health Management Society 2013*, October 14-17, New Orleans, LA.

Haykin, S., (1999), *Neural Networks - A Comprehensive Foundation, 2nd Ed.*, NJ, USA: Prentice Hall International.

Breeze, P., (2005), *Power Generation Technologies*, Oxford: Elsevier.

Labatut, V., Cherifi, H., (2011). Accuracy Measures for the Comparison of Classifiers. *The 5th International Conference on Information Technology*, arXiv : 1207.3790

Beale, M., Hagan, M., Demuth, H., (2011). *Neural Network Toolbox™ User's Guide.* Natick, MA, USA: The MathWorks, Inc.

Ghoreyshi, M., Singh, R., (2004). Using neural network for diagnostics of an industrial gas turbine. *Proceedings of ASME Turbo Expo 2004: Power for Land, Sea and Air*, June 14-17, 2004, Vienna, Austria.

Landgrebe, T.C.W., Paclik, P., Duin, R.P.W., Bradley, A.P., (2006). *Precision-recall operating characteristic (P-ROC) curves in imprecise environments* (123-127), Hong Kong. doi: 10.1109/ICPR.2006.941.

Dreiseitl S., Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: a methodology review. *Journal of Biomedical Informatics*, Oct-Dec, 2000; 35(5-6):352-9.

BIOGRAPHIES

Carmine Allegorico is senior engineer at the GE Oil&Gas in Florence, Italy. He received his Master Degree in Mechanical Engineering from the University of Napoli Federico II. He currently works in the Remote Monitoring & Diagnostic Center and is involved in the development and implementation of complex analytics for on-line diagnostics and predictive maintenance; furthermore he is technical leader of projects for advanced remote system monitoring, prognostic and asset health management, leading collaborations with Universities and GRC. As expert of Gas Turbines and data analysis, Carmine also provides support for performance studies, operability of turbomachineries and development of machine learning algorithms. Before joining the RM&D Center, Carmine has worked for 5 years in the design engineering team doing a variety of tasks ranging from the design of new products, performance testing and troubleshooting, dynamic model simulations and design of GT control philosophies.

Valerio Mantini is a Systems and Analytics Engineer at GE Oil&Gas in Florence, Italy. He received his Master Degree in Control System Engineering from University of Rome Tor Vergata.

He currently works in the Remote Monitoring & Diagnostic Center developing and implementing diagnostic algorithms and techniques for early warning sensor fault detection; furthermore he has to follow test campaigns on pilot projects to install advanced instrumentation and analyze data acquired. Before joining the RM&D Center, Valerio has worked more than 3 years in space sector developing the thrust vector control system for VEGA launcher, 6dof simulator and post-flight analysis on official launcher telemetry.