Please select category below:
Normal Paper ☒
Student Paper ☐
Young Engineer Paper ☐

Automated feature selection for multi-channel anomaly detection

Leonard Whitehead[1], John Taylor[1,2,3], Wenyi Wang[1] and Biswajit Bala[1]

[1]*Defence Science and Technology Group, Department of Defence, Melbourne Australia*
[2]*CSIRO Data61, Canberra, Australian Capital Territory, Australia*
[3]*College of Engineering and Computer Science, The Australian National University, Canberra, Australia*

Leonard.whitehead@defence.gov.au, John.Taylor@data61.csiro.au

Abstract

Identifying and choosing the ideal feature inputs to a machine learning algorithm is a slow and complex task. Methodologies such as univariate techniques, dimensionality reduction techniques, stepwise selection techniques and expert knowledge are often used in evaluating the input features. These techniques, although often effective and powerful, are not considered to be automated feature selection techniques, because they require a thorough assessment and interpretation by the user. This paper introduces a search algorithm using unsupervised learning, which is built upon our previously developed methodology for applying machine learning to detecting faults in fielded machinery. The search algorithm explores the input feature space and selects the input feature(s) that are highly sensitive to anomalies within the given dataset. The algorithm uses a heat map that shows the sensitivity of the feature inputs to anomaly detection, which can be further developed into a fingerprint analysis method to isolate faults within mechanical systems. Currently, this algorithm is being tested using real world aerospace data for anomaly detection. Preliminary results are presented in this paper and the final analysis results will be reported in a future paper.

Keywords: Anomaly detection, automated feature selection, condition monitoring, fault fingerprinting, machine learning.

Introduction

Input features are known to have a direct effect on the performance of machine learning models. Feature selection (or variable elimination) is the process of selecting variables which efficiently describe the input data, reduce the effects of noise and provide good prediction results [1]. The automated feature selection for multi-channel anomaly detection is a wrapper-based feature selection algorithm, which uses an exhaustive search strategy (or grid search) with a two-sample Kolmogorov-Smirnov (KS) test for filtering purposes. The induction algorithm (i.e. the neural network) has been built on our previous research, which used the Levenberg-Marquardt (LM) optimizer [2]. The algorithm leads to a fingerprinting analysis via a heatmap which identifies

the set of sensors that are sensitive to the detected faults/anomalies within the dataset. The fingerprinting analysis can further be used to identify the location and/or potential causes of the mechanical fault.

Feature selection methods can be classified into the following categories: wrapper, filter, embedded and hybrid [3]. Filter methods use ranking techniques to evaluate and select the variables (or features) to be used as inputs for the prediction model. Correlation, entropy and distance metrics are all common examples of filter methods. Wrapper methods involve evaluating the input feature set based on the performance of the induction algorithm. Search algorithms are often used in the wrapper method, where the feature space is searched by an evaluation function applied to the induction algorithm until a stopping condition is met. Embedded methods use feature selection as part of the induction algorithm's training, such as decision trees. Lastly, hybrid methods are a combination of filter and wrapper methodologies.

Condition monitoring (CM) applications have used feature selection for many years, e.g. [4]. As CM applications increasingly exploit big data analytics and machine learning, the need for feature selection or feature extraction techniques has also increased. Saxena et al. [5] used a genetic algorithm (GA) for feature selection in the CM of rotating mechanical systems. The GA increased the classification accuracy and decreased the computational cost. Subrahmanya et al. [6] developed an algorithm with a Bayesian framework to simultaneously select the sensors and features to be used for on-board vehicle fault diagnostics applications. The success of this algorithm led to a reduction in the need for data transfer and to online processing for remote monitoring systems. Adams et al. [7] analysed both feature selection and feature extraction techniques for CM datasets on hydraulic actuators. For feature selection, the authors tested the so-called RelieF and variable importance algorithms. They discovered that the variable importance algorithm improved performance and reduced computational cost. The authors emphasized the need for feature selection in prognostics and health monitoring (PHM) applications to alleviate curse-of-dimensionality effects.

In this work, the original intent was to use the KS test p-values as a filter to identify critical inputs in situations where there are more than two input features. The initial hypothesis for the p-value technique is that a lower p-value means better anomaly detection for a model and a lesser chance of overfitting. Therefore, these feature inputs will have a higher ranking than others in such circumstances. However, in practice, the KS test p-values method did not turn out to be an effective filter, and more effort is required in this area. As a result, this paper only presents findings from the first two search spaces (sensitivity & overfitting), where a filter was unnecessary, as the entire search space was explored.

Methodology

We developed an algorithm for anomaly detection using a real world aerospace dataset. The dataset consists of 22 channels of aircraft engine performance data which contain an engine failure event towards the end of the time series. The aim of the analysis is to detect any fault-induced anomalies prior to the failure event as early as possible. The aircraft has two engines, one with the fault (11 channels) and one healthy (11 channels). Physics-based a priori knowledge of the system indicated that channel #6 would be the most sensitive to this type of failure. Hence, the neural network (NN) models were implemented to predict anomalies in channel #6 from the other input channels. The feature-selection task here is to implement an automated process to find the optimal combinations (or set of features) from all the other channels as the inputs to the NN models. The process considers both the sensitivity of the features to the fault-induced anomalies, and the avoidance of overfitting or underfitting. The

whole time series is divided in two parts; the early part for training and the latter part for testing of the NN models.

The algorithm has two main processes; the training stage and the evaluation stage. In the training stage, several NN models (using the same structure) were created based on different combinations of input features extracted from a predefined search space. In the evaluation stage, the models were filtered (ranked) according to a set of criteria described below. The highest ranking feature combinations (see the example below) then formed a new search space. The algorithm iterated between the training and evaluation stages until a stopping condition was met.

> *Input feature combinations*: To create the search space, input feature combinations of the high ranked candidates are taken. For example, if the model with input features (3,4) and another model with input features (6,7) are high performing, then the input feature combination (3,4,6,7) will be in the next search space. In the event that an input feature combination has recurring features, e.g. (3,2,2,4), the set of these features are taken (3,2,4) to be used as inputs. The models of feature inputs that existed in previous search spaces are re-trained.

The performance of the NN models is evaluated using three criteria: (a) overfitting or underfitting, (b) sensitivity to the anomaly, and (c) the *p*-value from a two-sample KS test using items (a) and (b) as inputs. In general, models that have a high sensitivity and do not overfit or underfit are considered to be high performing models, models that have low sensitivity but do not overfit or underfit are ranked low, and models that overfit or underfit are considered the lowest performing models (i.e. invalid). The input features from high-performing models become the higher ranked candidates for the next search space, and those of low-performing models become the lower ranked candidates. The overfitting and sensitivity ratios are calculated from samples that are taken from baselines (defined below) with specific-sized windows. There are three baselines used here: the training, testing and anomaly baselines, which are described below.

> *Training baseline:* The region at the end of the training period which is fault free.
>
> *Testing baseline:* The region at the beginning of the test period which has no fault.
>
> *Anomaly baseline:* The region towards the end of the test region prior to the anomaly (still fault free).

The sensitivity and overfitting ratios are calculated by Eqs. (1) & (2) based on the 99^{th} percentile of the squared error loss of each sample in the region. The size of the region can be changed depending on the structure of the data but must be consistent throughout the running of the algorithm. We select a region large enough to produce consistent metrics. In this case, we use a sample size of 15,000, which was determined through trial and error parameter tuning.

Due to the stochastic nature of NN training, each model is rerun several times with the same input channel combination to generate a sample set of sensitivity and overfitting ratios. A two-sample Kolmogorov-Smirnov (KS) test is performed to compare the distribution of the sensitivity and overfitting ratio samples for each input channel combination. The natural log of the mean sensitivity ratios and overfitting ratios are then plotted as a map using the Python *Matplotlib* library routine *imshow* [8] to produce a heat-map, as shown in Figure 2.

$$sensitivity\ ratio = \frac{Anomaly\ Baseline\ Loss}{Training\ Baseline\ Loss} \quad (1)$$

$$overfitting\ ratio = \frac{Testing\ Baseline\ Loss}{Training\ Baseline\ Loss} \quad (2)$$

Algorithm

1) Set the parameters, window size (*w*) and sample size (*m*).
2) For each single-channel input *i*:
    a. Train *m* models with input channel *i*.
    b. For every model trained in (a), calculate the sensitivity and overfitting ratio according to window size *w*. This creates the sample of sensitivity and overfitting ratios.
    c. Calculate and record the KS p-value[1] with the samples of sensitivity and overfitting ratios.
    d. Repeat steps (a) – (d) until all single channel inputs have been used once.
3) Plot the results (sensitivity, overfitting and p-values[1]) for fingerprinting analysis.
4) Calculate all two-channel combinations (this is the second search space).
5) For each channel combination *i* (e.g. 3 & 2) in the second search space:
    a. Repeat steps (a) – (c) in step 2, but also calculate the mean sensitivity and overfitting ratio of each sample.
    b. Repeat until all two-channel combinations inputs have been used once, note that the order does matter; i.e. (2,1) is not the same as (1,2).
6) Apply a natural log transformation to all recorded sensitivity and overfitting ratio means to aid visualisation needs (optional).
7) Plot the overfitting and sensitivity results from step 6 using *imshow*. Use fingerprinting analysis and compare the analytics to expert domain knowledge.
8) Apply a filter to create new feature input combinations from the current search space feature input combinations. This will form the new search space. Note, to implement step 7 visualisation, the search space size must be a square number because it is a matrix.
9) Repeat steps 5 – 8 until a stopping condition is met.

Preliminary Results and Discussion

For every search space, it is important to first check whether the fingerprinting conclusions are consistent. This means that the general results for each search space must produce similar outcomes. Figure 1 shows the key results for the single-channel search space, the sensitivity and overfitting values and the KS *p*-values associated with the overfitting and sensitivity ratios.

[1] This metric is under improvement

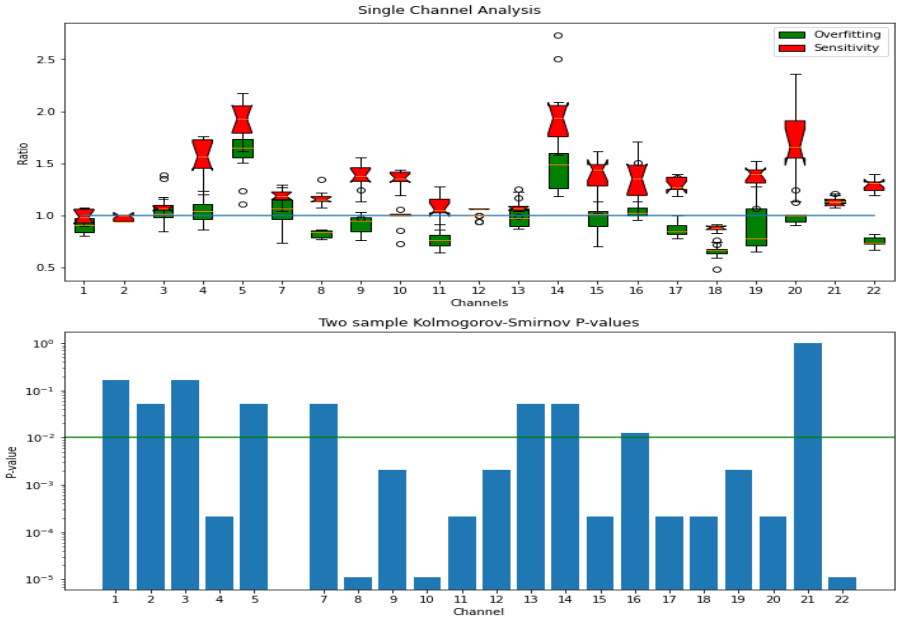

Figure 1 Single Channel Results. Overfitting, Sensitivity and P-value analysis.

The x-axis indicates the channel inputs. Note that the number 6 is skipped as this is the predicted channel – the target (or output of the NN models). The overfitting ratios are mostly located around the value 1, which is to be expected as the calculated loss at the end of the training set and at the start of test set should be approximately the same. Values that are far greater than one would indicate that the model has been overfitted and values less than one indicate that the model has been underfitted. In Figure 1, we can see that channels 5 and 14 both have relatively large overfitting ratios, indicating that the models are overfitting the data. Similarly, channels 4 and 15 have high sensitivity ratios with low overfitting ratios.

The *p*-value derived from the two-sample Kolmogorov-Smirnov test detects the difference between the sensitivity and overfitting distributions for each channel input. The lower the *p*-value, the more different the sensitivity and overfitting distributions are.

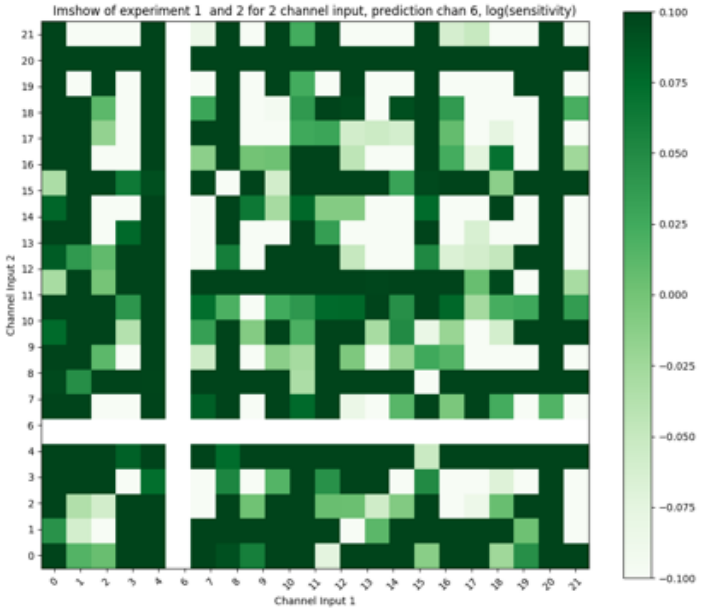

Figure 2 Heat-map (Imshow) Ln of Sensitivity Ratios from 2-channel combinations

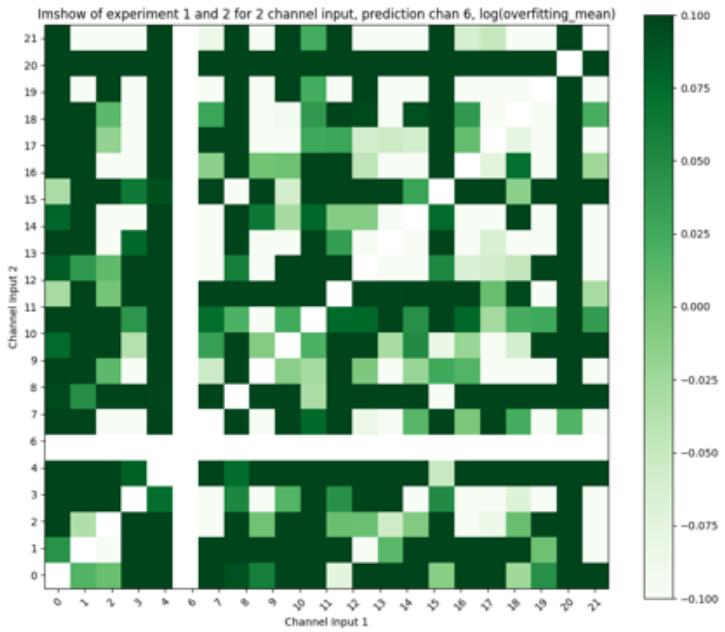

Figure 3 Heat-map (Imshow) Ln of overfitting ratios from 2-channel combinations (values >0 indicate overfitting, values <0 indicate underfitting)

The mean sensitivity ratio is a direct indicator of whether a model is detecting anomalies. However, if the model is underfitting or overfitting, then the sensitivity ratio could be spurious. The log of the sensitivity and overfitting ratios was displayed as an aid to visualise the results.

To properly interpret the 2-channel results, Figures 2 and 3 must be analysed together. Models that have a high sensitivity ratio with an overfitting ratio close to 1 (e^0) are strongly desired, as this indicates that the model is strongly detecting any anomalies that are present. Optimal means a high sensitivity ratio (capable of detecting the anomaly) and near unity overfitting ratio together with a low p-value (the two distributions from the sensitivity and overfitting ratios do not belong to the same class). For example, the square (3,2) in Figure 2 indicates that the NN with channel inputs 3 and 2 has a high sensitivity ratio whereas the square (3,3) represents the NN with channel input 3 has a low sensitivity ratio. However, Figure 3 square (3,2) has a very large overfitting ratio indicating an overfit and (3,3) has a very small overfitting ratio indicating underfit. Overall, despite (3,2) having a large sensitivity ratio, this value is inaccurate due to model overfitting.

A plot like Figure 2 can potentially be used for fingerprint analysis of the detected anomaly by applying domain-specific knowledge. For instance, the lower left quadrant of Figure 2 is generally darker than the upper right quadrant. This is because the sensors from the faulty engine are generally clustered in the low channel numbers, and those of the undamaged engine in the higher channel numbers. Channel 21 is an exception here, as it is from the faulty engine.

Future work

Future work will consist of further development of the methods described here, and on creating metrics for fingerprint analysis, where we seek to associate a pattern of sensitivities with the failure of a particular part in a piece of complex machinery, such as a gas-turbine engine. We will also continue to investigate the optimal design of a filter that searches for the best performing models for anomaly detection.

References

1. Chandrashekar, Girish, and Ferat Sahin. "A survey on feature selection methods." *Computers & Electrical Engineering* 40, no. 1 (2014): 16-28.
2. Wang, Wenyi, John Taylor, and Biswajit Bala. "Exploiting the Power of Levenberg-Marquardt Optimizer with Anomaly Detection in Time Series." *arXiv preprint arXiv:2111.06060* (2021).
3. Jović, Alan, Karla Brkić, and Nikola Bogunović. "A review of feature selection methods with applications." In *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*, pp. 1200-1205. Ieee, 2015.
4. Jack, L. B., and A. K. Nandi. "Genetic algorithms for feature selection in machine condition monitoring with vibration signals." *IEE Proceedings-Vision, Image and Signal Processing* 147, no. 3 (2000): 205-212.
5. Saxena, Abhinav, and Ashraf Saad. "Evolving an artificial neural network classifier for condition monitoring of rotating mechanical systems." *Applied Soft Computing* 7, no. 1 (2007): 441-454.
6. Subrahmanya, Niranjan, Yung C. Shin, and Peter H. Meckl. "A Bayesian machine learning method for sensor selection and fusion with application to on-board fault diagnostics." *Mechanical systems and signal processing* 24, no. 1 (2010): 182-192.
7. Adams, Stephen, Ryan Meekins, Peter A. Beling, Kevin Farinholt, Nathan Brown, Sherwood Polter, and Qing Dong. "A comparison of feature selection and feature extraction techniques for condition monitoring of a hydraulic actuator." In *Annual Conference of the PHM Society*, vol. 9, no. 1. 2017.
8. J. D. Hunter, "Matplotlib: A 2D Graphics Environment," in *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90-95, May-June 2007, doi: 10.1109/MCSE.2007.55.