Automated review of offshore maintenance records

Matthew Celnik, Principal Consultant, DNV GL, Helix Building, Kelvin Campus, Maryhill Road, Glasgow, G20 0SP

Chris Bell, Consultant, DNV GL, Cromarty House, 67–72 Regent Quay, Aberdeen, AB11 5AR

As part of offshore maintenance verification, the verifier conducts a review of the maintenance records for each asset. This ensures operators of offshore installations record and trend data for their installations correctly.

Verification teams currently do this review by selecting a small sample of maintenance records for each safety/environmental critical element (SECE). A small sample is chosen as it is too labour-intensive to review all records manually. However, this approach can lead to unintended bias; for example, by over- or under-sampling erroneous records. This reduces the value of the review to the operator: can we implement a time-efficient and effective way to review a full set of maintenance records?

In this paper, we present a computational method for analysing large numbers of offshore SECE maintenance records. The method uses a machine learning algorithm to analyse an entire maintenance record set for one SECE on an offshore installation. The algorithm can detect anomalies in the recording of maintenance records to produce a "target sample", allowing the reviewer to focus on records which deviate from the expected format/quality, as oppose to a random sample.

The algorithm automatically produces this target record sample very quickly. This is computationally efficient, allowing thousands of maintenance records to be analysed in a few seconds with a useful accuracy level. The verification engineer can then apply a focussed approach to verification, reducing manual effort and reducing potential for human error or bias in sampling. This increases project efficiency and allows DNV GL to make more useful findings and recommendations to the operator.

Introduction

The rapid increase in machine learning technology across the oil and gas sector offers offshore operators the chance to automate high-cost, error-prone tasks in which the cumulative effects of inconsistency and analytical error can adversely impact safety. For instance, as part of any assets' assurance process, it can be instructive to review maintenance records for insights, particularly trending issues and identifying potential improvements. The goal should be to ensure the asset is performing safely and effectively, with high reliability while adopting the most cost-effective strategies for all maintenance work.

The Offshore Safety Case Regulations [ref. 1] place a requirement on offshore operators to have safety and environmental critical element (SECE) performance standards in place, and have an independent verifier assure they are being met. Review of the maintenance records is a key element of this task; hence there is a legal obligation, as well as a financial incentive, to perform such reviews.

Maintenance and reliability teams around the UK have adopted various methods to help them review maintenance records for each SECE more effectively, for example:

1. By selecting a small sample of maintenance records for each SECE to review. A small representative set of data is chosen as it is deemed too labour-intensive to manually review all records. This approach can lead to a sampling bias in the results. For example, inaccurate maintenance records may be overlooked, reducing the value of this process.
2. Employ a team of people whose sole purpose is to review and track the maintenance records for each asset. Usually each person employed for this task is focussed on one SECE or equipment item. This approach has the advantage that more maintenance records are reviewed in detail, reducing the sampling bias but potentially leading to human bias / error in the results. This approach also has a high man-cost and time commitment (typically one week every month per SECE) in carrying out this type of review using skilled engineers.

Method 1 above has been traditionally used by independent verifiers, such as DNV GL, when verifying performance standard compliance in the UK.

In order to better analyse a large number of maintenance records in full, a potential solution is to use a machine learning (ML) classification algorithm with natural language processing. Verifiers can then analyse the entire set of maintenance records for each SECE on the asset. They can detect anomalies in the way maintenance records for each SECE have been recorded, allowing the reviewer to focus purely on records which have anomalies, as opposed to a random sample method currently used by most companies. Equally, such techniques could be utilised on any dataset for which analysis is repetitive, labour intensive, or prone to human error.

Dataset

This study considered a single fire and gas detector inspection dataset from an offshore operator. This is real data, which we have anonymised for this study. We chose this dataset because detectors are common elements offshore – hence there exist a lot of maintenance records – and are prone to failure or mis-calibration, so there should be a suitable number of FAIL records against which to predict.

The dataset includes the following key fields, among others:

- **Work order number**, a unique number given to each maintenance or inspection item;
- **Location**, identifying the detector, light or circuit being inspected or maintained;
- **Description**, which provides a detailed description of the work undertaken, typically completed by the technician/engineer responsible;
- **Test result** – the "class" – marked as PASS, FAIL or FAILFIX indicating whether the SECE or equipment item functioned as expected;
- **Target finish date**, when the maintenance or inspection was due to be completed; and
- **Actual finish date**, when the maintenance or inspection was completed.

The location and date fields are useful for secondary calculations, such as reliability and availability of detector types over time. Reliability being whether the detector functioned on demand, and availability being a measure of downtime. Comparison of target and actual finish dates gives a measure of the work deferral rate. This paper does not consider such calculations further as they are relatively simple to implement once the test results are known.

Instead, this work considers only the description field: a free-text field in which the test and the result are typically described. Using a combination of natural language processing and machine learning (support vector machine) algorithms, we explore the question: can a computer predict the test result by analysing the text?

The dataset contains 2,119 records in total, covering a period from April 1992 to May 2018. However, the vast majority of the records (2,036) are for the years 2008–2012, as illustrated in the table below:

Table 1. Work order counts in test dataset by year and recorded class.

Recorded class	1992–1999	2007	2008	2009	2010	2011	2012	2013+	TOTAL
PASS	-	-	-	401	396	381	256	25	**1459**
FAIL	-	-	-	11	11	15	3	-	**40**
Unclassed	28	30	452	-	-	18	92	-	**620**
Total	**28**	**30**	**452**	**412**	**407**	**414**	**351**	**25**	**2119**

We note some important features of the dataset at this stage. First, there are very few FAIL records overall, approximately 2% of the total (3% of the classed records). This means we have a highly skewed dataset, with few examples of the FAIL class. This could make fitting a machine learning model difficult.

Second, approximately 30% of the records are unclassed, that is they do not a have recorded test results. We cannot use those records to train a machine learning model, hence our available training data reduces to 1,499 records, which is still a sufficient number when considering two classes only (PASS and FAIL). There could be several reasons why the test results are not recorded, for example a change in management system could mean the previous results were not carried forward into the new system. In such cases, the method presented here can assist operators reconstruct the missing data.

Data classes

A common feature of SECE management records across all operators is a column indicating the test results, usually in terms of performance standard compliance, but not exclusively. In machine learning terms, the test result is the *class* we are attempting to predict from the data features.

The simplest recording system is simple a PASS/FAIL flag. More complex systems have codes for FAILFIX[1], or several FAIL codes to indicate the failure type. This study classifies all FAILFIX and complex failure codes as FAIL only, to simplify the classification model. This approach is sufficient to support verification activity, though 2nd party activities supporting operations more require a finer approach.

Generally, the PASS/FAIL criteria are determined by the SECE performance standards, which form the basis of the verification activity. However, performance standard criteria can be complex, particularly for installation with diverse detection systems. For example, the criteria may allow for failure of 1–2 detector heads in an area, as long as sufficient coverage is maintained by other detectors. This can make ML predictions based on natural language processing exceptionally difficult. Consider the theoretical inspection record text covering four detector heads:

"Detector | Result

-------- | ------

GD-A Pass

GD-B Fail

[1] FAILFIX denotes a failure which was immediately fixed, before the log was updated. This study considers all FAILFIX as FAIL for assessment purposes.

GD-C Pass

GD-D Pass"

Is the overall class PASS, because 75% of the heads function, or FAIL because there is a single failure? That depends on the performance standard criteria, which can differ across assets, and even for a single asset if different areas have different criteria. This study does not codify the performance standards in such detail, but instead applies a simple approach: if something looks like a fail, it is a fail. Hence the above record would be classed as FAIL regardless of the performance standard. It turns out this approach is sufficient to support the verification engineer, who can make a quick comparison of flagged records against the standards. We plan to improve the text analysis and investigate performance standard comparison as future work.

Methodology

This study uses a linear support vector machine (SVM) algorithm [ref. 3] to classify inspection records based on the free-text field. Records are classed PASS or FAIL as described above. We used the open-source Python package scikit-learn (version 0.19.2) [ref. 1] for this work, which implements a linear SVM classifier with stochastic gradient descent learning [ref. 5]. A machine learning classification model is an example of supervised learning. This paper does not present in detail the function of SVM models, descriptions of which are widely available online and elsewhere. In broad terms, a classification model predicts a *class* for each dataset record, given the *features* of the record. In this study, the class is the PASS/FAIL flag discussed above. The record features are numerical properties derived from the free-text field. Feature derivation is discussed below.

The overall method used in this study is:

1. Split the dataset into training and testing subsets;
2. Pre-process the dataset (data cleaning);
3. Derive numerical features using natural language processing;
4. Train a classification model on a training subset;
5. Predict training subset classes to evaluate model performance;
6. Revise feature derivation and model parameters to improve fit (repeat from step 2 as necessary);
7. Predict testing subset classes to evaluate overall model performance; and
8. Estimate SECE or equipment availability and reliability (not considered further in this study).

Each of these steps are detailed in the following subsections. We implemented the above as a Python (version 3.6) program.

Data cleaning

Prior to using a ML model, the dataset must be processed. ML models typically describe data by a reduced feature set; where a feature is a numerical property of the record. The data cleaning process (pipeline) transforms the raw maintenance text data into the limited feature set. Figure 1 shows the cleaning pipeline applied for this study.

The first step is to merge all relevant text fields into a single block of text. Our dataset contained only a single text column, so we skipped this step.

The second step is to remove *boilerplate* text. By boilerplate, we refer to text appearing in all – or almost all – text records which does not describe the test result. Most often this is a copy of the work-order instruction, or some other note for the person undertaking the work. Such text can significantly influence the predictive power of the resultant model, as considered in the results section below. Unfortunately, the boilerplate text is generally specific to each operator and SECE, hence a custom data cleaning pipeline is required for each. For this study, we manually reviewed the records to determine the form of the boilerplate text, and compiled a list of 39 unique sentences/phrases to be excluded from the data. We implemented the boilerplate removal function using Python regular expressions [ref. 6] to ensure any case or whitespace typographical errors were captured.

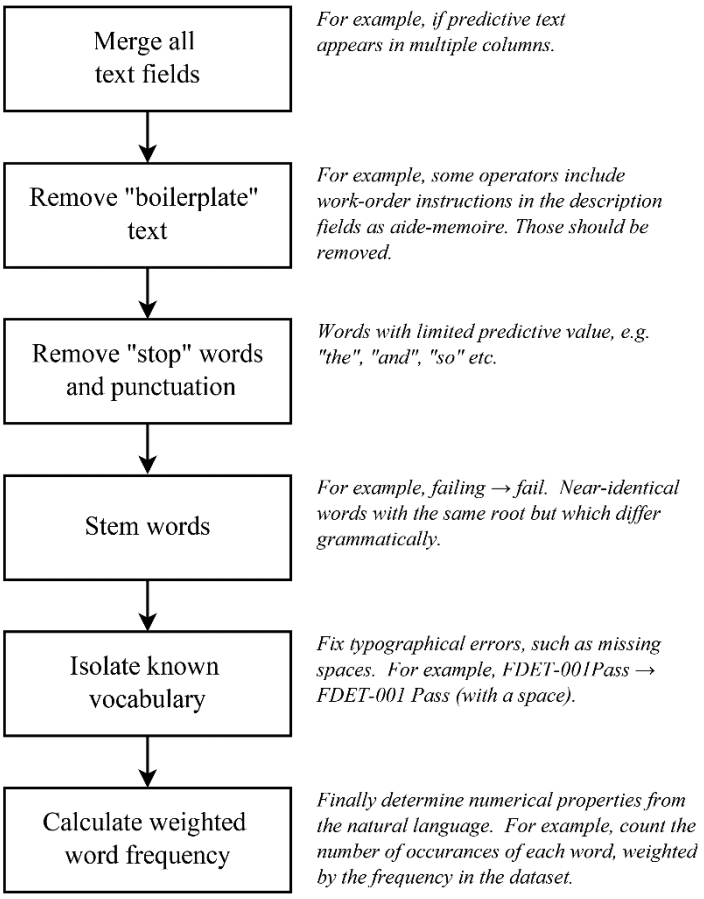
The third step is to remove *stop* words from the text, as defined in the Natural Language Toolkit (NLTK; version 3.4) [ref. 7]. Those words – such as *the*, *and*, *so* et cetera – are common and therefore provide limited predictive value. Removing them prevents them being falsely identified as important features when training the classifier.

The fourth step is to *stem* words to further reduce the potential feature set. This study used the Porter stemmer implemented in NLTK. Stemming strips suffixes from words so they reduce to their root sense. For example, "failed", "failing", "failure" all become "fail". The sense is the same in all cases; they differ only grammatically.

The final cleaning step applied here is to isolate known vocabulary with white space. This reduces training problems associated with mis-typed words, for example "DET1Pass" becomes "DET1 Pass" – as we know the word "pass" is likely an important feature.

The cleaning pipeline is the most time-consuming part of the program to construct as there are many ways in which a human could mis-type or otherwise make a mistake. We could envisage adding additional steps too, for example a spelling-corrector. The importance of maintaining a high-quality dataset for the purposes of machine learning cannot be overstated.

Figure 1. Pre-processing pipeline (data cleaning)



Feature derivation

The final step in the cleaning pipeline in figure 1 derives the numerical features on which the classification model trains and predicts. This study uses the weighted frequencies of words in the text description field as the data features. We used the scikit-learn *term frequency inverse document frequency* (TF-IDF) normalization [ref. 12] for this purpose. This weights the word frequency by the inverse of its frequency in all records, hence less-common words are weighted relatively higher than more-common ones.

The scikit-learn TF-IDF vectorizer allows definition of a custom vocabulary to limit the words considered. If no vocabulary is provided, the vectorizer uses the whole vocabulary derived from all words in the dataset. We initially ran the model training using the whole dataset vocabulary, whereupon it appeared the model was basing prediction on certain *artificial* terms. For example, it would weight highly terms like “fr2”, “s02”, “a101” etc. Such terms, we realized, are parts of the detector identification numbers. While this may be a useful result, in that we can identify particular detectors or detector circuits prone to failure, these terms are of limited value when creating a generic classification model. Hence, we derived a limited vocabulary based on the important, yet common, words in the dataset.

We applied three vocabulary filtering techniques to determine whether limiting the word-sets, using knowledge of the maintenance process, could improve the accuracy of the ML predictions. These vocabularies (a–c) are:

a. Classify records based on the frequency of all words in the dataset (minus the boilerplate stop words). After boilerplate and stop word removal, the F&G dataset contained approximately 1,200 unique “words”, though that includes the detector IDs.

b. Classify records based on the frequency of a limited vocabulary (words and phrases – note, these are already stemmed): FAIL, PASS, ALARM, SCADA, OPER, RAIS, RESPOND, FOLLOW, ERR, NFF, “ALL OTHER”, RECTIFI, CONTAMIN, UNABL, SUSPECT, LEAK, CHANG, REPLAC, AVAIL, REPAIR, “WAS OK”, NOT, RESPOND, ANOMALI, ROUTIN, NO, FAULT, “NO FAULT”, COMPLET, ALL, ADJUST, ASSUR, and FUNCTION; and

c. Classify records based on the frequency of two words only: PASS and FAIL;

Model training

A machine learning model must be trained to make predictions. The training step uses a subset of the data as a basis and the ML model uses the training subset to determine its internal parameters to make predictions. For example, a SVM model determines weights for each data feature to calculate an overall score for each record. Records with a net-positive score are classed PASS; those with a net-negative score are classed FAIL. Words with higher predictive power will have a higher absolute score; this allows us to observe the most important words.

We noted above, the dataset is heavily skewed towards PASS records; 97% of classed records are PASS, 3% are FAIL. Applying a simple classification training to this dataset will be heavily biased towards PASS, simply because 97% of the time the model will think a record is PASS regardless of the features. To counteract this bias, the model applies a class weighting, inversely proportional to the rate of occurrence of each class in the training dataset. This increases the relative prediction score of features common to FAIL records.

For this study we used 75% of the records to train the models, and the remaining 25% to test the predictions. We made a key assumption about the data at this stage, which was tested later: the reported PASS and FAIL values are either accurate, or the errors are few and randomly distributed. If the errors are randomly distributed, then this should appear as noise in the prediction (the prediction accuracy will be lower, but it won't be systematically wrong). If the errors are not randomly distributed, for example if the operator is systematically mis-recording PASS and FAIL, then the prediction will exhibit the same skew. To evaluate this, we undertook a manual verification of the full supplied dataset.

To ensure a uniform training set, we constrained the randomized training/testing sampling to ensure the proportion of PASS and FAIL records in each matched the proportions in the total dataset.

Predictions

We split the dataset into training and testing subsets as a guard against over- or under-fitting. If the model is over-fitted to the training data, it will give excellent predictions on the training data, but poorer predictions on the testing data. If the predictions on the training and test data show similar distributions, then we have more confidence in the capability of the model when applied to other, similar datasets.

This paper presents prediction results as 2×2 "confusion matrices", which are read row-wise: each row represents a recorded class (PASS or FAIL), each column presents a predicted class. The cell values are the row-summed percentages of records predicted as PASS or FAIL. When there are few mis-predictions (low confusion) then the matrix has a strong backwards diagonal (top-left to bottom-right). When the confusion is high, the cell values are closer in value. Confusion matrices can present many classes together, but here we are interested in only two.

Results

Figure 2 shows the confusion matrices for the three vocabularies when training on the full text descriptions in the F&G detector dataset (including boilerplate text), using the PASS/FAIL classes as assigned by the operator (unverified). Using all words, or a limited vocabulary, achieves ~95% match of PASS records, though worse for FAIL records.

Prediction using only the words "pass" and "fail" is particularly poor. This is because the boilerplate text contains many instances of those words as notes to the operator. For example, the below sentence is repeated in most records:

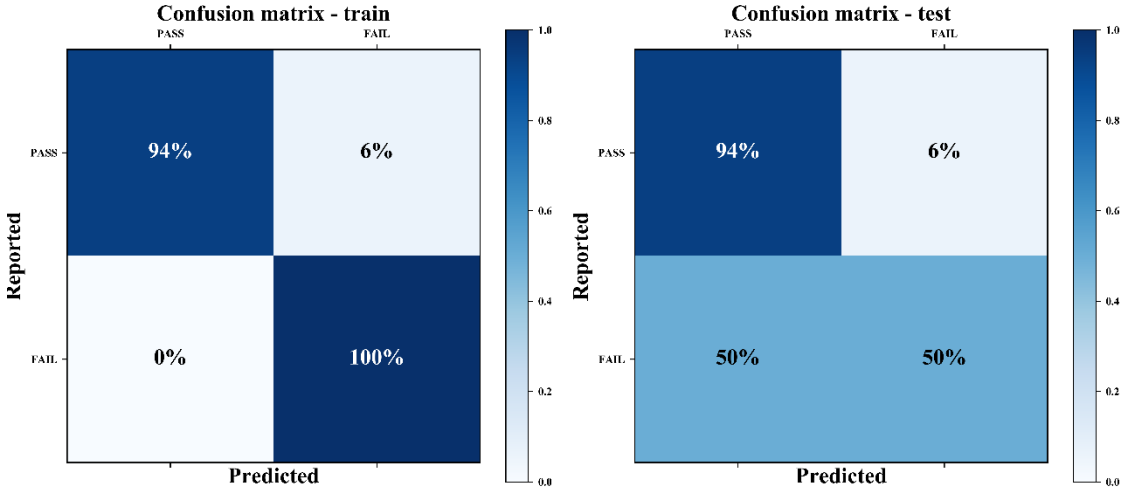
"N.B. If the first head tested fails then repeat the test for the next head alphabetically until a head passes the test"

Such text masks the true text features, decreasing the predictive power of the model.

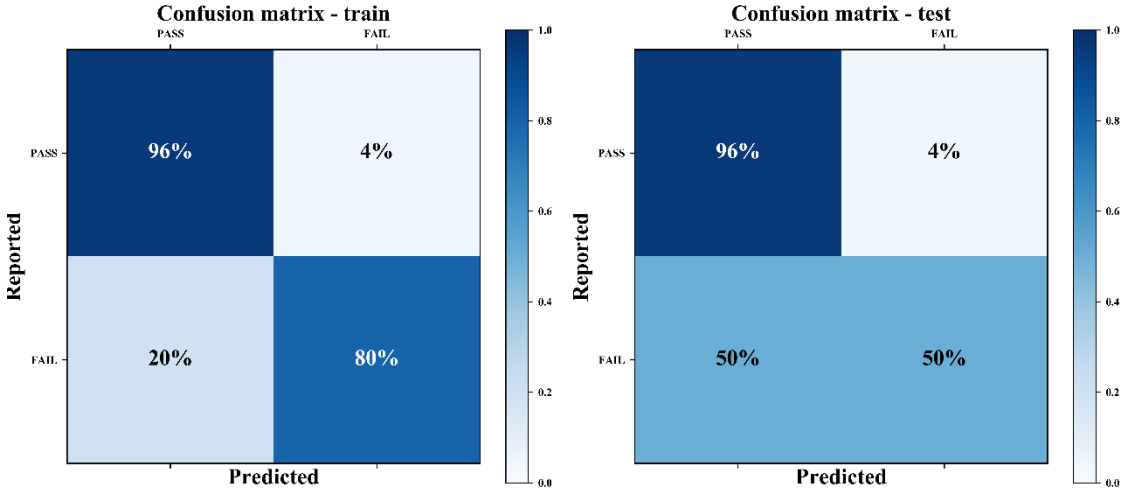
Comparing the training and text data predictions for vocabularies (a) and (b) only, there is a good correspondence when predicting on the reported PASS records. However, the model completely fails to distinguish reported FAIL records in the test data. This suggests the model is over-fitted to the training data, as it does not work for the test data at all.

Figure 2. Prediction matrices using different vocabularies (a–c) on unverified classes, boilerplate included

(a) Using all words to define features



(b) Using a limited vocabulary of features



(c) Using "pass" and "fail" only as the features

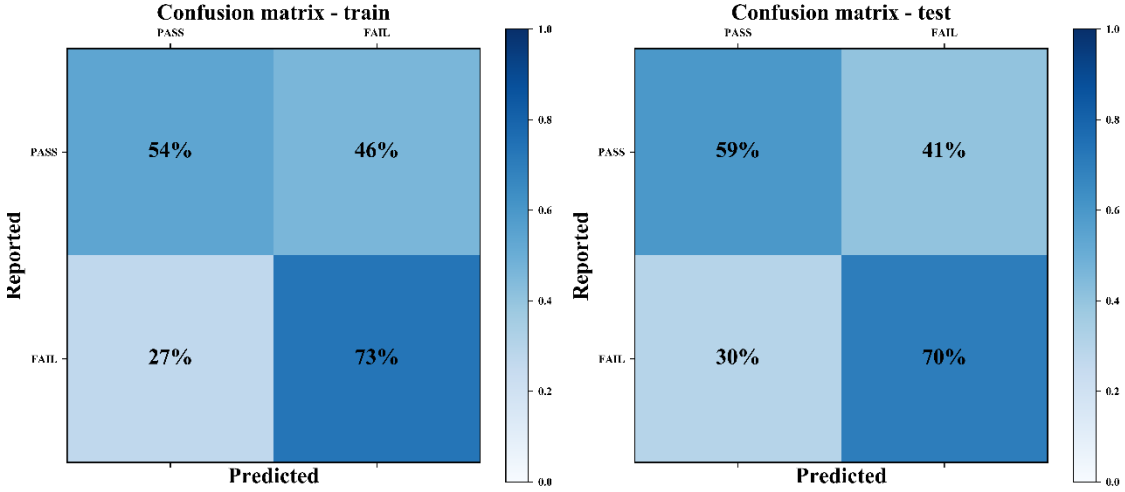


Figure 3 presents the confusion matrices for the same data as figure 2, except this time with all identified boilerplate text removed. There is a marked improvement in the predictions using all vocabularies, and far better agreement between the training and test data predictions.

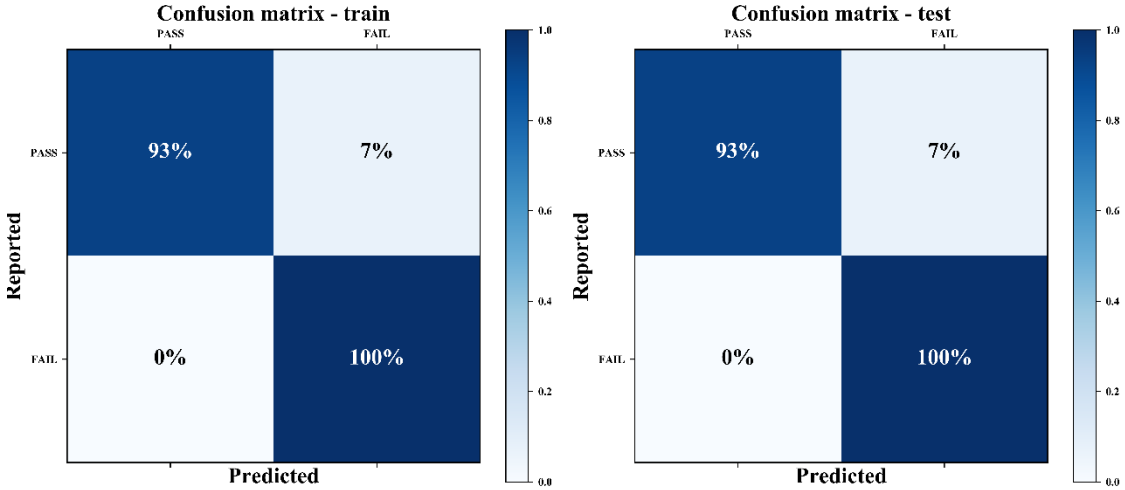
Vocabulary (c) – “pass” / “fail” only – now provided a good 98% accuracy when matches reported PASS records, though is less good at predicting reported FAIL records (~70%). Of course, there are very few FAIL records in the test dataset (10) as there are very few in the overall data.

The predictions for vocabularies (a) and (b) are good at ~93–98%. This accuracy level is already sufficient to guide the offshore verifier, and to significantly improve usefulness of conclusions to the operator. Using all words, vocabulary (a), achieves a 93% accuracy on PASS records, and correctly matches all FAIL records even in the test dataset.

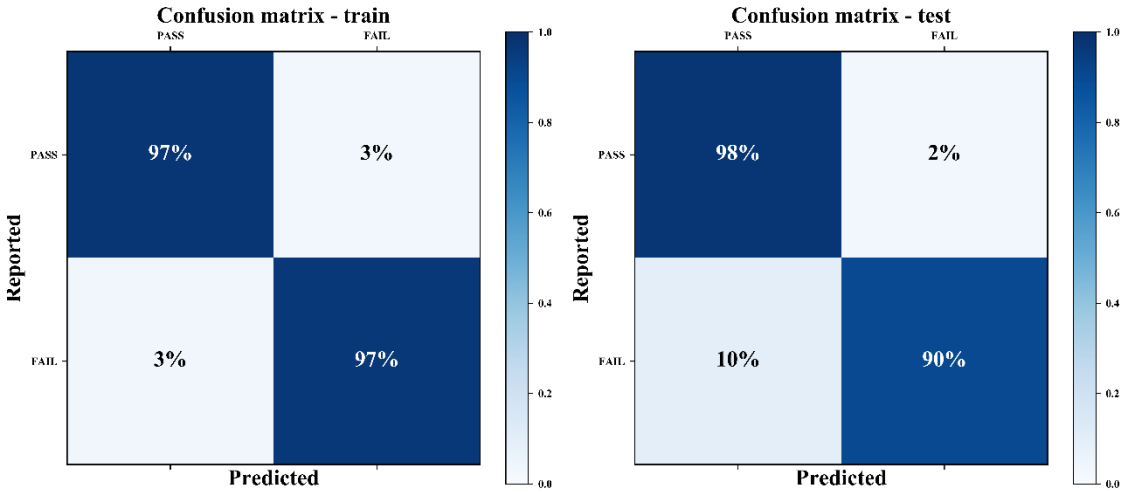
Reducing the vocabulary (b) further improves the prediction on PASS records, though it mis-matches one FAIL records in the test dataset. Manual inspection of that records revealed it covered six gas detector heads: three passed inspection and three were unavailable due to temporary ducting. Hence, while three heads did not technically fail, they were nevertheless unavailable and hence the initial FAIL class is correct, and the model prediction is in error. This highlights the process by which the ML predictions can focus verifier effort.

Figure 3. Prediction matrices using different vocabularies (a–c) on <u>unverified</u> classes, boilerplate removed

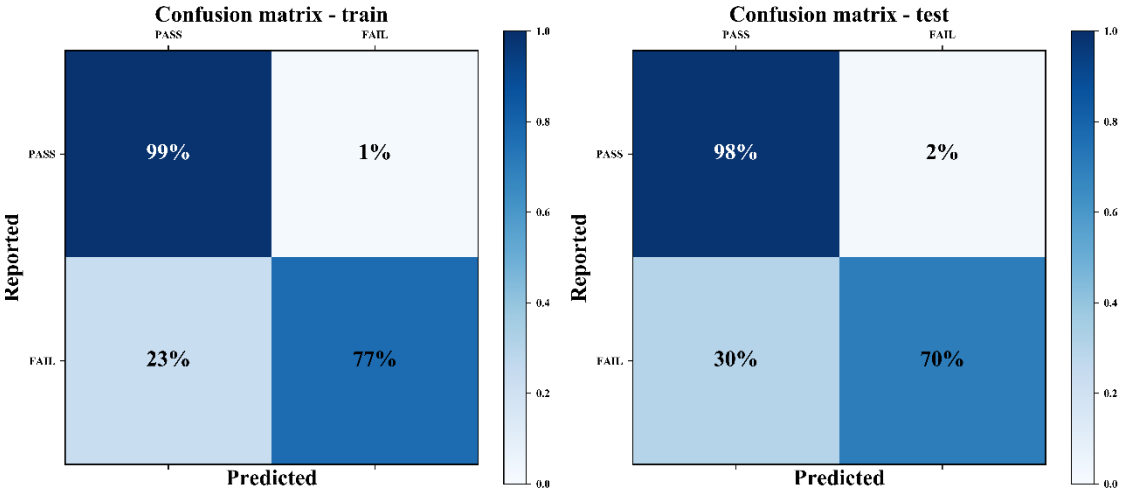
(a) Using all words to define features



(b) Using a limited vocabulary of features



(c) Using "pass" and "fail" only as the features



Typically, a verifier will be more interested in reported PASS records which the ML model predicts as FAIL, because that suggested an under-reporting of potential problems. Considering only the limited vocabulary (b), the model predicted 2% of the test dataset PASS records to actually be FAIL; this equates to eight records in total. The next section discusses manual verification of the dataset to improve predictions; here we considered the verified classes of the test record discrepancies.

Table 2. Test records reported as PASS but predicted as FAIL

Record	Record description summary[2]	Reported class	Predicted class	Verified class
MP653865	Gas detectors all passed function tests but failed to communicate with control room (link down). The DNV GL verifier marked this as a fail.	PASS	FAIL	FAIL
MP654456	As above.	PASS	FAIL	FAIL
MP654876	As above.	PASS	FAIL	FAIL
MP653708	As above	PASS	FAIL	FAIL
MP661933	One out of four heat detectors failed its function test. Whether this counts as a performance standard failure is unknown, nevertheless it should have been flagged therefore the DNV GL verifier marked this as a fail.	PASS	FAIL	FAIL
MP667139	Record indicates no fault found, hence DNV GL verified it as a pass, and the ML prediction is incorrect.	PASS	FAIL	PASS
MP639638	Four out of eight heat detectors failed function tests and were replaced, testing successful upon replacement. The DNV GL verifier marked this record as a fail as it should have been fail-fix, not pass.	PASS	FAIL	FAIL
MP648676	As above, one out of four heads failed and was immediately replaced.	PASS	FAIL	FAIL

Of the eight records above, the ML prediction agreed with the human verifier for seven, highlighting the power of this approach to improve the verification activity.

[2] The record description is summarized here to ensure it is anonymous.

Table 3. Non-zero feature counts for each dataset/model.

Dataset	Boilerplate	Total word count	Non-zero features (vocabulary a)	Non-zero features (vocabulary b)	Non-zero features (vocabulary c)
As provided	Included	1501	99	22	1*
	Excluded	1279	73	17	2
Manually verified	Included	1645	87	22	2
	Excluded	1450	87	19	2

** For the unverified dataset with boilerplate included, vocabulary c made predictions based only on the occurrence of the word "fail".*

Manual verification

To verify if our initial assumption is correct, that the recorded dataset included only a few random errors, an offshore verifier at DNV GL performed a manual verification of the entire dataset (2,119 records). The verifier review identified 37 (of 1459) records marked as PASS which should have been FAIL, and 3 (of 40) records marked FAIL which were really PASS. Figure 4 shows the row-wise confusion matrix for the manual verification. This gives a mis-reporting rate of 2.7%. Anecdotally from past experience, we anticipated a mis-reporting rate of 5–10%, hence the verified rate appears good. Nor does there appear to be a systematic bias in the mis-reporting, which would typically be shown as a higher fraction of PASS records verified as FAIL (indicating an under-reporting of potential problems). In this dataset, it appears there are fractionally more FAIL records verified as PASS, however, the very small number of FAIL records means we cannot draw a strong conclusion here. Including previously unclassed records, DNV GL determined 6% of the records to be FAIL, and 94% PASS. This compares to 3% / 97% for the unverified records, suggesting a slight under-reporting of fails.

Table 4. Verified record counts

Reported class	Verified PASS	Verified FAIL	Total
PASS	1422	37	**1459**
FAIL	3	37	**40**
UNCLASSED	565	55	**620**
Total	**1990**	**129**	**2119**
Percentage	**94%**	**6%**	**100%**

Overall, the manual verification gives us confidence in the predictions based on the unverified dataset. Nevertheless, we re-trained the model using the verified classes.

Figure 4. Manual record verification error matrix

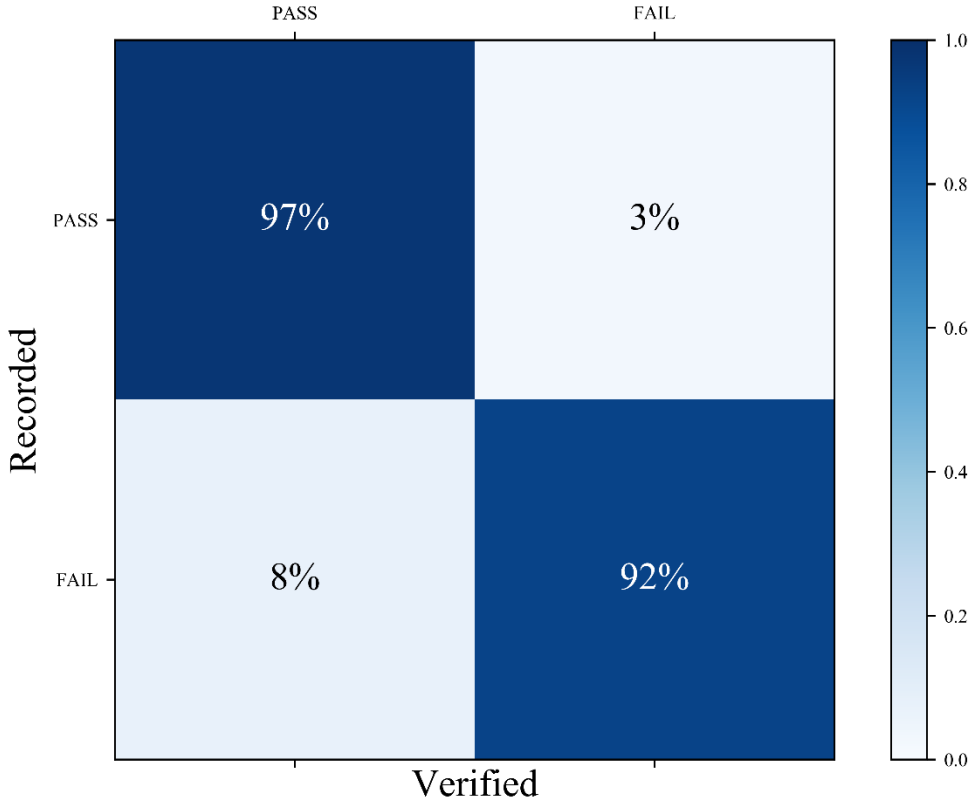


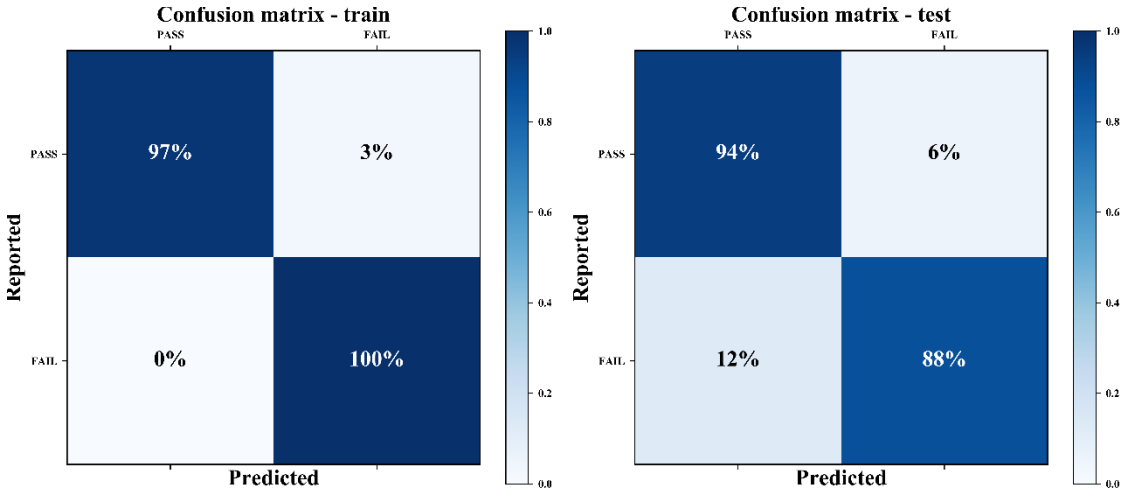
Figure 5 shows the confusion matrices using the verified classes, with boilerplate removed. We now have a larger sample of FAIL records as the verifier identified PASS records which should be FAIL, and assigned classes to the previously unclassed records. We therefore anticipate an improvement in the predictive power of the model.

When using all words, vocabulary (a), the PASS predictions are similar to the unverified dataset, but the FAIL predictions show a deterioration in the test dataset. This suggests the model is overfitting, probably because it is using too many text features. Limiting the features, vocabulary (b), now shows a marked improvement over using all words, and good agreement between the training and test datasets. There is perhaps still a slight over-fitting, as the test agreement is slightly worse than for the training dataset, but in general the fit is good at 97% accuracy.

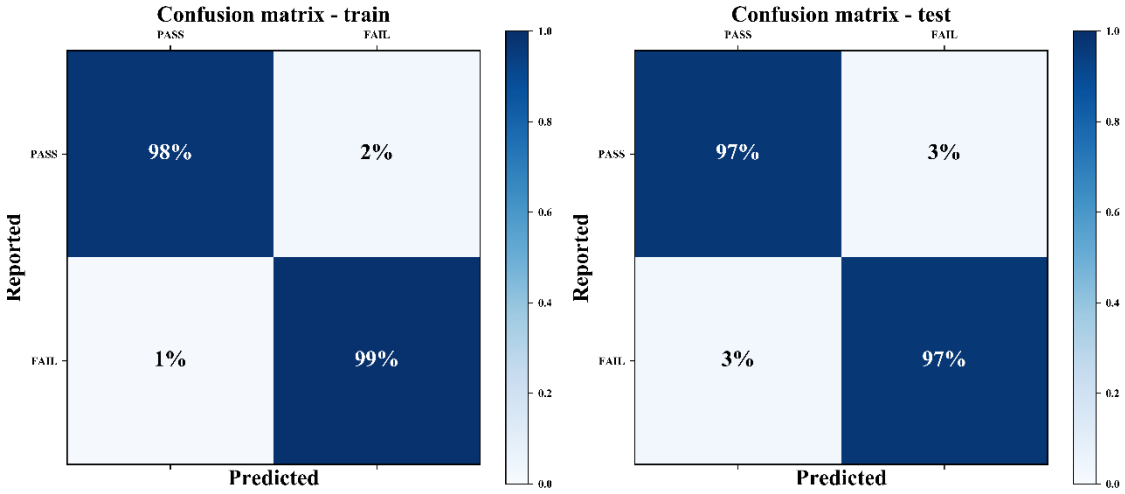
The final vocabulary (c) using the words “pass” and “fail” only now shows a poor ability to fit the PASS records, though improved fit to FAIL records. Models based on this very simple vocabulary appear very sensitive to the available training data, suggesting they are under-fit.

Figure 5. Prediction matrices using different vocabularies (a–c) on <u>verified</u> classes, boilerplate removed

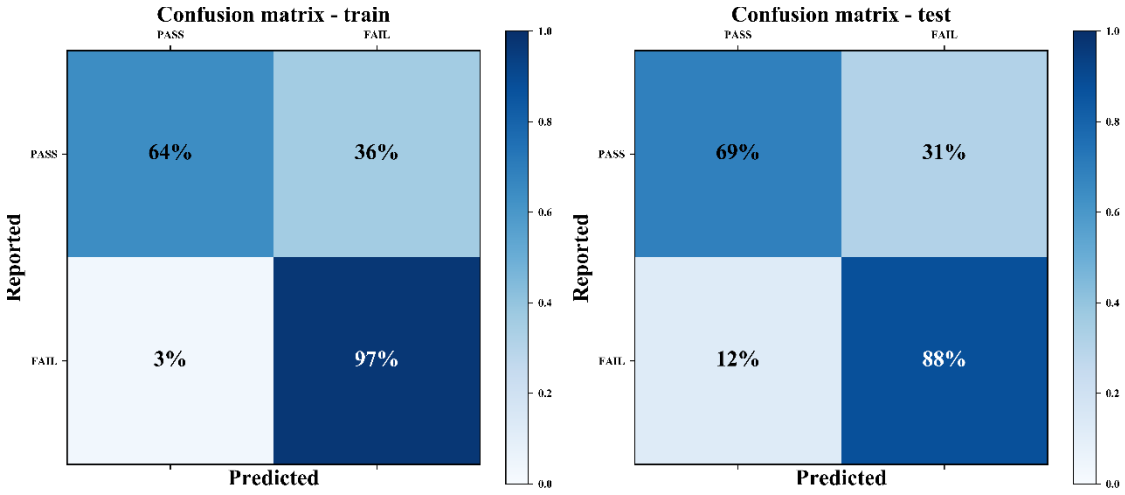
(a) Using all words to define features



(b) Using a limited vocabulary of features



(c) Using "pass" and "fail" only as the features



Considering the vocabulary (b) results, the model predicted 3% FAIL where the verifier had marked PASS, which equates to 29 records. It is possible the verifier also mis-reported, as reviewing over 2000 records is a repetitive task prone to human error. However, upon further inspection the records were found to indicate PASS as verified, which suggests there is still room to improve the model predictions if necessary. The ML model particularly has difficult correctly predicting records with limited text. For example, it consistently marked records as FAIL which had only the following description:

"4 monthly function tests completed. N.F.F."

where N.F.F. means "no fault found". While it is relatively quick for a verifier to determine this as a pass, the objective of this work is to focus effort on potential discrepancies in the records. Such records could be screened out by including a simple text-search step prior to the ML model, or if the operator includes more detail in the records themselves. DNV GL intend to further refine this approach as we roll it out across our verification services.

Conclusions

This work demonstrates that a trained SVM algorithm can rapidly identify records with potential anomalies, although the predictive power is dependent on the pre-processing cleaning steps, particularly removal of boilerplate text. We can gain additional prediction improvements by limiting the feature set to a known vocabulary of important words and phrases, informed by expert knowledge. This method works well, even for highly skewed datasets with few recorded fails.

This method gives maintenance and reliability teams a more focussed approach to check records by specifically targeting the anomalous maintenance records. Crucially, by reducing the amount of time spent reviewing non-erroneous maintenance records, this approach increases project efficiency and allows more useful findings and recommendations to be made. DNV GL intend to progressively roll-out this approach to all offshore verification activities based in Aberdeen area over the near future.

As part of this work, we identified several pitfalls, particularly in the way operators currently record their data. As data analytics becomes more prevalent in the offshore oil & gas industry, we anticipate these issues will be of greater importance:

1. Where possible, operators should store numerical test results as separate fields in the management system, particularly those results directly related to performance standard criteria. This will greatly simplify subsequent machine analysis and trending.
2. Operators should consider revising how they report tests in their management systems, particularly to remove boilerplate text (notes to operators, work order instructions etc.) Such text detrimentally impacts the predictive power of the ML models, and its removal is an initially laborious process, as it requires a human to parse the records.
3. Where possible, operators could revise existing records, with additional fields, to facilitate future verification and trending activities. Data analytic/machine learning techniques such as presented here can support such activities by automatically filling in missing data.
4. Training the offshore workforce on the potential use of the maintenance logs for trending, and how they can facilitate this, could also be a worthwhile endeavour.

Performing quality checks and data audits is just one application for this methodology. We can combine the resultant predictions with availability and reliability calculations to help operators trend their asset performance over time, potentially identifying areas for improvement or efficiency gains. This approach also enables operators to reconstruct data missing from their management systems, as well as identify potential systemic mis-reporting issues.

References

1. UK Health & Safety Executive, The Offshore Installations (Offshore Safety Directive) (Safety Case etc.) Regulations 2015, Guidance of Regulations, L154, 1st edition, 2015.
2. Scikit-learn documentation, https://scikit-learn.org/stable/index.html.
3. Scitkit-learn documentation, section 1.4, support vector machines; https://scikit-learn.org/stable/modules/svm.html.
4. Scikit-learn documentation, section 4.2.3.4, feature extraction – Tf-idf term weighting; https://scikit-learn.org/stable/modules/feature_extraction.html#tfidf-term-weighting.
5. Scikit-learn documentation, package reference, SGD classifier; https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html.
6. Python 3 documentation, standard library reference, regular expressions; https://docs.python.org/3/library/re.html.
7. Natural Language Toolkit (NLTK), online documentation; https://www.nltk.org.

Software

This study used publicly available Python packages for data-processing, machine learning and plotting. The table below lists all packages used.

Table 5. Python software packages used

Software package	Version	Software package	Version
Python	3.6.6	Numpy	1.13.3
Matplotlib	2.0.2	Pandas	0.23.4
NLTK (natural language toolkit)	3.3.0	Scikit-learn	0.19.2

Acknowledgements

This study uses real offshore maintenance data provided by a DNV GL client. While we have anonymised the data for presentation, we would like to express thanks to the data provider for facilitating this work.