Leak detection in compressed air systems using unsupervised anomaly detection techniques

Antoine Desmet[1], and Matthew Delore[2]

[1] *Komatsu Mining Corporation, Rutherford, NSW, 2320, Australia*
Antoine.Desmet@mining.komatsu

[2] *School of Electrical Engineering and Computer Science, University of Newcastle, Australia, 2308*
Matthew.Delore@uon.edu.au

Abstract

Critical components of mobile mining machinery, such as brake and lubrication, are typically powered by compressed air. The compressed air system is subject to leaks in either pipelines or the air-actuated components; and these leaks can cause accelerated wear or unexpected machine shutdowns. Given the size of these machines, fault-finding can be time-consuming. Remote diagnostics is possible by analysing the machine's air accumulator pressure with respect to component activation times. However, since every component draws air directly from the accumulator, the resulting drops in pressure all superimpose over the accumulator's charge and discharge cycles. The result is a highly dynamic trend, making visual diagnostic difficult for anything but major leaks. In this paper, we apply unsupervised anomaly detection techniques to detect developing air leaks. Our method uses machine learning to associate patterns in pressure drop from the accumulator with the activation of each air-powered component. We first apply a wavelet transform to the accumulator pressure trend to make patterns apparent in the time-frequency domain. We then use the Random Forest algorithm's *feature importance* to select the most informative wavelet scales. Finally, we trial two anomaly detection methods over the selected inputs: the first uses a clustering approach (LOF), while the second uses a neural-network approach (autoencoder). Once the learning phase (using historical data) is complete, we test the system on an intermittent leak which occurs only when a particular component is activated. We find that both systems perform well, and the LOF trades accuracy for speed with respect to the autoencoder.

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

1. Introduction

Mobile surface mining machinery uses compressed air to actuate some of its critical components. These machines are typically fitted with a single air system to lower costs associated with maintenance and parts. However the lack of decoupling between components means a leak in a particular component can potentially drain the main air supply, which will impact the supply of all other air-actuated components. For instance, an air leak in the system could impact the release of the air brakes, which triggers a machine shutdown. This single point of failure and potential for either damage or downtime motivates our work on air system diagnostics. In particular, this paper presents a technique to detect the early signs of developing leaks and to locate them.

The scale of the compressed air delivery network and the number of components it supplies contributes to the difficulty of troubleshooting the system. Furthermore, not all air-actuated components are fitted with telemetry sensors, due to practical and economical reasons. The telemetry data most relevant to our problem are:

- The pressure in the air accumulator. This is the accumulator that supplies every air component.
- The activation times of air-actuated components, recorded by the machine's control system.

From a diagnostics point of view, there are two broad classes of leaks: permanent and intermittent. *Intermittent* leaks are only visible when a particular component is activated. This type of leak occurs in the component itself, or in the air hose that supplies it. On the other hand, *permanent* leaks are a constant bleeding of air. They occur on parts of the system which are constantly pressurised: compressor, accumulator, air supply manifold, cable reeler, etc. Permanent leaks are simpler to detect as they cause a higher overall usage of air, but are difficult to pinpoint. On the other hand, intermittent leaks are difficult to identify since they are only observable

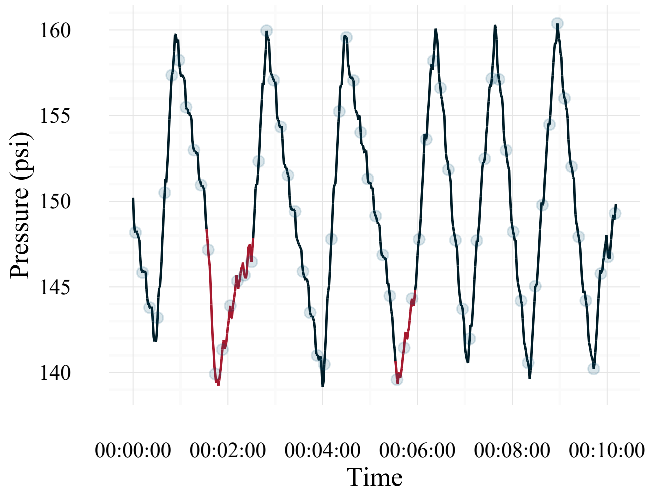

Figure 1. Sample time-series of the accumulator's air pressure, in psi. Sections coloured in red correspond to the activation of grease sprays (use compressed air as atomiser and propellant). Each dot correspond to the triggering of a filter-cleaning air pulse

during narrow time windows, however the leak's location can easily be reduced to whichever component(s) is active whenever the leak appears. For this reason, we will focus on intermittent leaks, though the algorithm we present applies to the detection of both types of leaks.

Each time an air component is activated, it draws a volume of compressed air from the accumulator which results in a pressure drop. Therefore, by associating component activations with observed pressure drops, a machine expert can theoretically diagnose which circuits consume abnormal amounts of air. However, in practice this is difficult for all but very large leaks: indeed the air receiver pressure trend is a combination of multiple signals that superimpose. The sample signal on Figure 1 shows when filter-cleansing air pulses occur: though they all have the same characteristics, their shape is influenced by the "sawtooth" pattern of the accumulator charging and discharging. With the addition of sensor noise, a *visual* comparison of pressure drops is nearly impossible. A physical modelling approach to this problem is not deemed practical either, due to the complexity of modelling air flows on such a large scale. Additionally, variations in components, hose types and lengths between machines would require tedious parametrisation and calibration of a physical model on an individual machine basis.

In this paper, we propose an algorithm that performs unsupervised anomaly detection on the compressed air system. In particular, we will use a Machine Learning algorithm to cope with the large number of patterns present in the air accumulator time-series. The preliminary step consists of preparing the signal to highlight the characteristics of each pressure drop, and the underlying sawtooth pattern. We then provide a fault-free dataset for the algorithm to discover relationships between pressure patterns and the activation of air-actuated components. Finally, we run this algorithm on new data to detect anomalies. The algorithm runs in an unsupervised fashion: it is not trained to search for particular types of faults. Instead, it produces a value for each time-step which represents how "abnormal" the signal is, compared to the training set. We expect this metric to peak at times where a leaking circuit is activated; or in the case of a permanent leak, to globally increase. We compare two anomaly detection techniques, the first is density-based (LOF), the second uses Neural Networks (Autoencoder). Our method has several advantages: 1. The learning algorithm allows faster deployment compared to physical modelling, 2. the model is trained on a particular machine's data. It will therefore be more sensitive than a generic one 3. the model is not geared towards the detection of a particular fault, it merely detects abnormal patterns of air consumption, and tries to identify their origin 4. the model is able to self-calibrate based on fault-free dataset.

2. RELATED WORK

An overview of detection methods for leaks can be found in (Datta & Sarkar, 2016). A class of solutions rely on sensing product escaping through the leak: changes in electrical resistance in tanks (conductive fluids) (Ramirez, Daily, Binley, LaBrecque, & Roelant, 1996), wire running alongside the pipelines whose conductance is affected by the leaking material, or light and radar-based thechniques detecting changes in absorbtion and scattering when escaping gasses mix with the atmosphere (Sandberg, Holmes, McCoy, & Koppitsch, 1989; Bevenot, Trouillet, Veillas, Gagnaire, & Clement, 2000). However none of these methods would apply to our case, since the compressed air leaked into the surrounding environment is indistinguishable from the ambient air. Another technique consists of detecting the acoustic signal of compressed gas escaping the pipleine (Fuchs & Riehle, 1991): while this could apply to our case, it may be difficult to identify these vibrations over the background noise of the mobile plant, and more so to differentiate air escaping through leaks against air being discharged by the air-actuated components. This method seems to be preferred in gas pipelines where the pipeline mainly operates in a "steady state", making the identification of unusual vibrations much simpler than in a dynamic system. Flow and mass measurement techniques have also been proposed, where the pipeline is broken down in sections with meters fitted between each of them (Murvay & Silea, 2012). The principle of mass conservation implies that a balance should exist at the start and end of each section, unless a leak exist. This solution involves modelling flow to predict expected flow at the other end. It may be particularly difficult to implement in the presence of transient flows, let alone consideration on the cost-effectiveness of the system, and the potential for the calibration and maintenance tasks to outweigh the benefits of the system. Perhaps the most applicable method is based on the detection of transient waves created as pulses of compressed air interact with changes in the pipeline, including leaks (Colombo, Lee, & Karney, 2009). This method would however require additional sensors with a finer resolution, and the calibration of "normal" transient waves on a per-machine basis.

If we place a constraint on using only sensors already present on the machine, there are two broad approaches for fault detection (Hodge & Austin, 2004). The first uses a model built to detect particular fault conditions. In the context of machine learning, this is a *supervised* classification task, where an algorithm trains over a dataset containing occurrences of both the fault and normal data. The second technique models only the *normal* case, and searches for deviations from the model. This is an unsupervised technique that detects anomalies, novelties or interventions, and the output is a discrete variable that represents "distance" from normality. We focus on the latter category as it does not require a training dataset of abnormal cases (Qin, 2012). Since the air compressor system has several failure modes, acquiring multiple instances of all faults over a diverse range of machines would be prohibitively time-consuming. Furthermore, a supervised approach limits the algorithm's scope to detect only *known* faults.

A model of the system operating in normal conditions can be built using physical models, however the complexity of the compressed air delivery network and its variability from one machine to another would make this task prohibitively difficult. Statistical models (gaussian, regression), rules (Montgomery, 2005) or PCA (Callegari, Gazzarrini, Giordano, Pagano, & Pepe, 2014) are all applicable, yet may not be optimal for dealing with large amounts of data with high dimensionality and non-linear relationships. Machine learning models are better suited for this task, for instance one-class SVMs (Song, Takakura, Okabe, & Nakao, 2013), clustering algorithms (Seungmin, Gisung, & Sehun, 2011) and autoencoders (Sarkar, Reddy, Giering, & Gurvich, 2016).

Since our objective is to detect abnormal patterns in univariate time-series, we could use Dynamic Time Warping (Zhao, Liu, Wang, & Liu, 2014): an algorithm that produces a metric indicating how much warping in time and amplitude is required to make the two samples conform. It is used to detect heartbeat anomalies in electrocardiograms (ECG) (Zhang, Kinsner, & Huang, 2009). However this method would fail as our signal is the result of multiple processes superimposing. The patterns generated by a process will constantly vary, based on the state of the other coupled processes. In this regard, our signal is closer to electroencephalograms (EEG) and the wavelet decomposition technique presented in (Cooper et al., 2015) seems relevant to our problem. Likewise, in (Khan, Chalup, & Mendes, 2016), the authors combine wavelet transforms and neural networks to perform a classification task on complex time series (voice recordings).

While the time-frequency decomposition performed by wavelets makes the patterns in our signal more apparent, it also increases the dimensions of our dataset. This means the anomaly detection algorithm will have to ultimately reduce the high-dimensional input into a single value which represents the degree of abnormality. A range of dimensionality-reduction algorithms exist: Support Vector Clustering (Boecking, Chalup, Seese, & Wong, 2014) and manifold techniques (Wong et al., 2012) are considered state of the art, but are generally slow to converge or train. These solutions are not compatible with our objective to create a process that operates in real-time. We will therefore use simpler methods, where the final dimensionality reduction is an aggregation of errors (or distances) on each dimension. The two algorithms that we will use are introduced in the Anomaly Detection section.

3. Feature derivation and selection

Figure 1 shows the overall shape of the air accumulator's pressure is a sawtooth. This results from the compressor cyclically recharging the accumulator to a high setpoint, and pausing while it discharges to a low setpoint. The shape of the sawtooth is continuously altered as components draw air from the accumulator: air usage either accelerates the discharge of the accumulator, or slows down its charging (depending on the compressor's state at the time). Monitoring the parameters of the sawtooth pattern should therefore reveal leaks: the slower charges and faster discharges caused by a leak would alter the compressor's duty ratio of charging and discharging states. While this method is likely to a succeed in detecting continuous leaks, this time resolution would be too coarse to identify intermittent leaks. Since component activation times are shorter than a compressor cycle, and indeed multiple components may activate during a duty cycle, it is preferable to isolate each pressure drop and analyse its features individually. This section will cover "features": the transformation(s) applied to the raw pressure signal time-series in order to form a set of inputs. The transformation(s) are chosen so that the "features" display clear associations or repetitive patterns, in such a way that a learning algorithm can identify and model them.

3.1. Wavelets

Since patterns have distinct frequencies, a time-frequency decomposition will help us analyse patterns in isolation. This type of transform decomposes the signal in a sum of signals that have set frequencies – much like a Fourier transform, however a windowed Fourier transform's temporal resolution is limited by its fixed window size. Instead we decide to use a wavelet transform, as it adapts its temporal resolu-

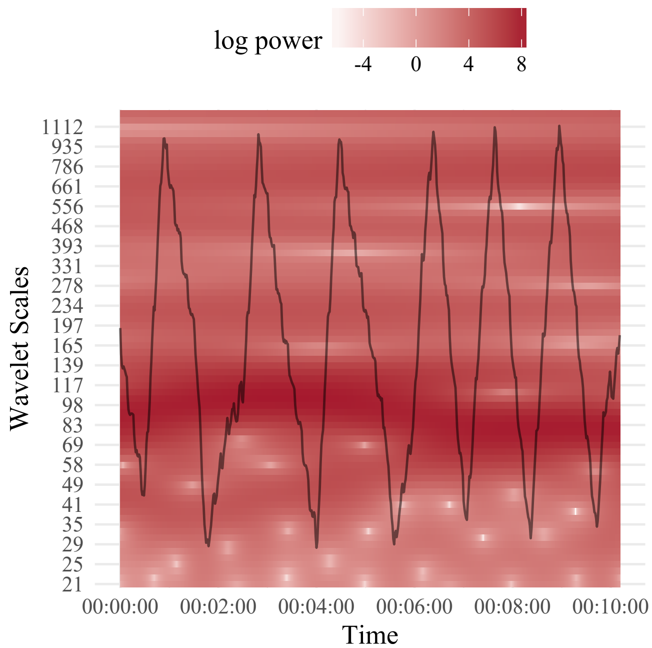

Figure 2. Wavelet filter output of fig.1's time-series, colorised by power (red is highest, log scale). The black line shows the original signal which has been rescaled to fit over the spectrogram (and therefore is not to scale). The figure illustrates how the wavelet transform represents variations in the signal: the as the period of the sawtooth pattern increases between 02:00 and 04:00, the spectrum shifts towards higher scales.

tion to the period of each spectrum component. Since our method is based on matching transients in spectrum with the (de)activation of air components, the superior temporal precision of wavelet transforms makes it a compelling choice. Unlike the FFT, wavelet transforms are not limited to decomposing the signal into sinusoidals: there are indeed many types of wavelets. We expect the closing and opening of air valves to result in sudden drops in pressures, and these may not be well represented by a sinusoidal decomposition. We experimented with Morlet wavelets after finding several examples of its application to anomaly detection and condition monitoring in the literature (Huang, Thareja, & Shin, 2006; Gelman, Patel, Persin, Murray, & Thomson, 2013). While a full review and comparison of wavelets is beyond the scope of this initial work, the wavelet transform opens an opportunity to further improve accuracy by comparing wavelet families, and identifying the one most suited to detect the relevant patterns in our signal. Figure 2 shows a sample of our wavelet decomposition for the sample signal presented on fig 1.

3.2. validation

Most air components are controlled by automata, therefore each activation should have identical characteristics. We also expect each component's pressure drop to produce some unique *signature* pattern. This pattern in frequency and amplitude should be highlighted by the wavelet transform. Our proposed anomaly detection method identifies leaks by validating the signatures of the components that are active at the time. This approach therefore hinges on the assumption that the wavelet transform presents these signatures in a *consistent* and *distinguishable* manner, and enough so to train a learning algorithm.

Table 1. Confusion matrix of the Random Forest classifier. 5000 samples, one sample per second.

		predicted class			
		upper	lower	inactive	rate
actual	upper	235	0	20	7.8%
	lower	2	73	2	5.2%
	inactive	96	50	4523	3.1%
				overall:	**3.4%**

To verify this assumption, we setup a supervised classification learning experiment. We first train a learning algorithm with solved cases, i.e. a large set of signatures and their matching component. Our hypothesis will be validated if the algorithm successfully learns how to identify the component, given its signature. Of the several machine learning algorithms available for this task, we choose a Random Forest (RF)(Breiman, 2001). RF is a learning algorithm that uses an ensemble of decision trees, where each tree is trained with a subset of the training data, and a subset of the available inputs. Since we will use the RF as a classifier, the output is the majority vote of each tree. We chose the RF classifier as it *natively* produces a "variable importance" ranking, which we introduce in the "feature selection" section. Besides, RFs have a good tendency to avoid overfitting and deal well with large input sets (our wavelet transform has 95 scales).

We focus our machine learning experiment on the most challenging area of the system: grease sprays. Grease sprays use compressed air to atomise and propel grease onto large areas of the machine that require lubrication. Two reasons motivate our decision to focus on this system: sprays are exposed to the environment and get damaged or blocked, making them of particular interest for reliability. Recognising the signature of sprays also challenges the learning algorithm, since the upper and lower air sprays are variants of the same system, and are expected to produce *comparable* signatures. Our experiment involves training the RF to recognise the signatures (or lack thereof) of either grease spray systems.

3.3. Results

Figure 3 shows a sample result of the classifier. The line is coloured in green or blue when components are active: upper and lower grease sprays, repectively. Otherwise, the line is coloured in grey when sprays are inactive. Areas shaded in red correspond classification errors.

The sample shows the algorithm can recognise when compo-

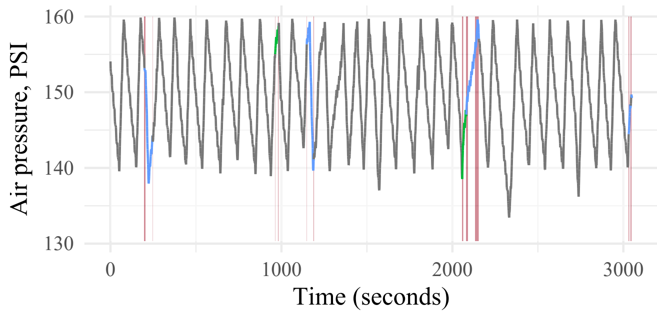

Figure 3. Sample result of the classifier. Red-shaded areas show classification errors. The different blue and green colours on the line indicate times where grease sprays were active

nents are active (3.1% error, table 1), and is also capable of distinguishing between the two lube spray systems – despite their similarity in design. The overall classification error rate is 3.4%, and we conclude that the wavelet transform indeed extracts “learnable” features from our signal.

3.4. Feature selection

Having verified our hypothesis, we now wish to reduce the number of wavelet scales we provide to the anomaly detection algorithm. This will reduce the effects of the “curse of dimensionality”: where the number of training samples required grows exponentially with the number of inputs (Hastie, Tibshirani, & Friedman, 2009). It should also benefit by reducing run-times, particularity for the LOF algorithm introduced in the following section.

To select the wavelet scales that provide the largest amount of information, we compute “variable importance” information. Variable importance can be obtained by iteratively replacing inputs by random values, and measuring the reduction in classification accuracy (Perzyk, Kochanski, Kozlowski, Soroczynski, & Biernacki, 2014). The random forest classifier algorithm we used also ranks inputs by the mean decrease in Gini index each time it is used at a tree’s decision node (Liaw & Wiener, 2002).

From a visual inspection of the wavelet transform’s output (Fig 2), and knowledge of the system, we expect the wavelet scales to contain information on the following:

Low Frequencies patterns with period greater than the compressor cycles (scales >120), of no interest for our analysis.

Medium frequencies fundamental sawtooth pattern (scales 70 to 120), and air component activations (scales 30-70).

High frequencies noise (scales <30).

Figure 4 shows the importance metric assigned to each scale. It satisfies our initial expectations: the extreme ends of the range contain little information. A peak forms around the wavelet scale 50, and the 30 most important scales for our pattern-recognition problem are found in the 30-100 interval.

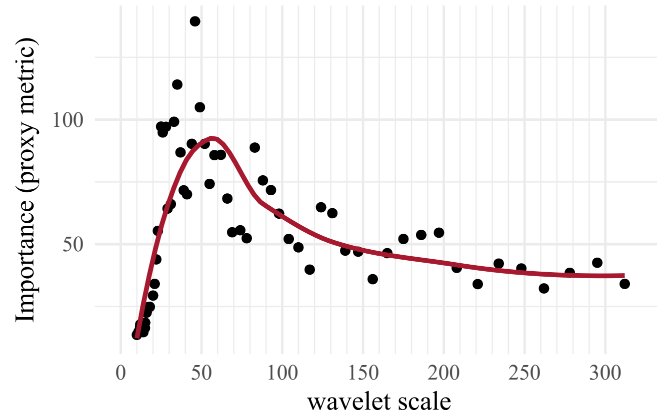

Figure 4. Importance of wavelet scales, as measured by the Random Forest algorithm, by mean decrease in Gini index. The red line shows a local regression (LOESS)

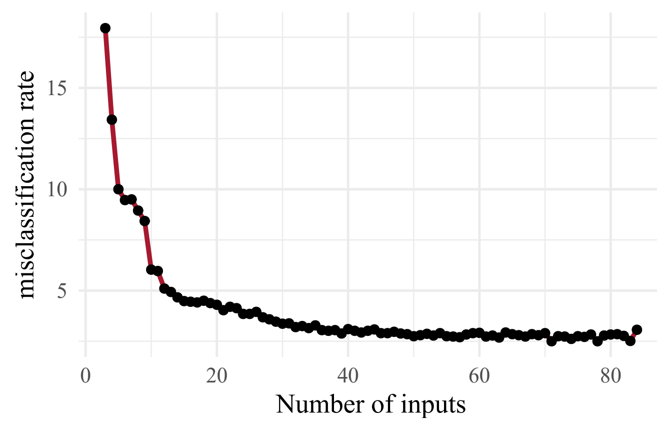

Figure 5. Random Forest’s misclassification rate as we incorporate more outputs (ouputs are sorted from the most to least informative).

Having ranked our inputs by decreasing order of importance, we now select how many to retain. For this, we iteratively train RFs while increasing the number of inputs, following the input ranking order. Figure 5 plots the Random Forest’s misclassification rate (Out-Of-Bag) against the number of inputs. We observe a steep decrease in misclassification rate as we add the 10 most informative scales. An “elbow” forms after the 10$^{\text{th}}$ one, and we decide to use this many inputs.

4. ANOMALY DETECTION

The previous section demonstrated the effectiveness of the wavelet transform at extracting the features of air pressure drops in a consistent form. This processing step allows a learning algorithm to recognise which component(s) cause some observed pressure drop. Using this, we could detect anomalies through discrepancies between the classifier's output and the "ground truth". For instance, if the classifier failed to detect the activation of a component, we could assume the "signature" of this component has changed, and therefore an anomaly is developing. However this simplistic approach presents several disadvantages:

- The RF may still succeed at classifying patterns significantly different from training examples, simply by *partially* matching some features. In this particular application, the RF's great ability to generalise is detrimental: it would conceal the changes we are searching for.
- The classifier's accuracy is too low for this approach: at a one second sampling rate, a mismatch would occur on average every 30 seconds.
- A continuous metric that represents "distance" from a normal pattern is preferred over a boolean value (i.e. "match" or "mismatch"). Indeed a discrete value lets us set a decision threshold to meet constraints, such as false positive rate.

In this section, we will compare two machine-learning anomaly detection techniques: Local outlier Factor (LOF) and Autoencoders (AE). Both methods address the shortcomings previously itemised. To compare these algorithms, we require a test set comprising of hundreds of both positive and negative cases. While finding negative cases in machine data is trivial, we do not have enough positive cases to support comparison. We therefore decide to synthesise data, using a model of the compressed air system. We continue to focus on grease sprays, and synthesise a test set of 200 instances of both normal and faulty lubricant spray activation. Our model allows us to simulate anything from a fully blocked to a completely disconnected spray nozzle. We decide to simulate a moderately leaking nozzle, where the air consumption rate increases by 25%. A change of this magnitude would likely go unnoticed: it is virtually undetectable to the naked eye, as shown on Figure 6, and would not trigger the machine's "low air pressure" alarm either.

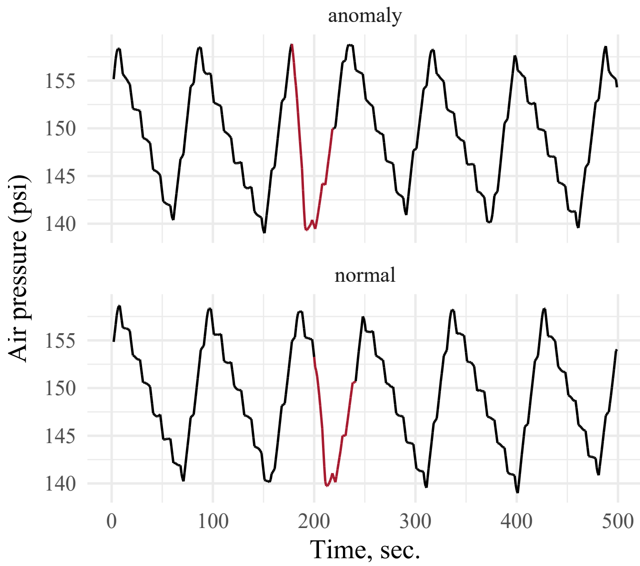

Figure 6. Examples of synthesised data, with lubricant spray activation times coloured in red. An abnormal activation is shown on the upper diagram, and a normal one is shown below. This highlights the difficulty of visually detecting excessive amounts of air consumption, especially since they can occur at any part of the compressor cycle.

4.1. Local Outlier Factor

Local Outlier Factor (LOF) is a type of density-based clustering technique optimised for outlier detection: the algorithm assigns to each data point "a degree of being an outlier" (Breunig, Kriegel, Ng, & Sander, 2000), which is a continuous value. LOF is based on the DBSCAN (Ester, Kriegel, Sander, & Xu, 1996) and OPTICS (Kriegel, Krger, Sander, & Zimek, 2011) density-based clustering algorithms. LOF stands out by comparing (i.e. dividing) the density of each record *against* the density of its k-neighbours. In the case of a perfect inliner, these two densities are equal and the metric is therefore one. The advantage of the LOF algorithm – in our case – is that it accounts for each air component's propensity to produce *consistent* patterns. For instance, the records from a highly-consistent component will form a well-defined, dense cluster, and the LOF algorithm will respond by increasing its sensitivity around the cluster. On the other hand, inconsistent patterns will form looser clusters and the LOF metric will be less sensitive.

We initially run the LOF algorithm over a dataset that combines 40 test samples with 5000 anomaly-free records. The anomaly-free records represent the machine in a similar state (in terms of active air components) to the 40 test samples. We expect the 5000 anomaly-free records to form clusters that define normality. If the test samples points are anomalous, they should be scattered away from the anomaly-free records, and therefore be assigned a high LOF score. This method initially failed: we found the 40 anomalous test samples would occasionally be similar enough to form a cluster of their own, and therefore score a typical inliner LOF. This problem can be addressed by increasing the *k-distance* parameter that controls the neighbourhood size, however this decreases the sensitivity of the algorithm. We obtained the highest performance by running the LOF one test sample at a time: indeed, since a *single* outlier cannot form a cluster, the test sample will systematically have a higher LOF.

4.2. Autoencoder

A clear disadvantage of the LOF algorithm is that the "anomaly-free" dataset needs to be re-assessed at each iteration, i.e. for every test record, which is computationally ex-

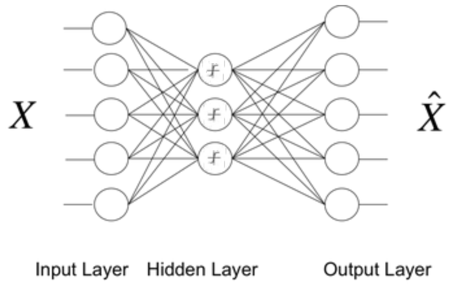

Figure 7. Architecture diagram of an autoencoder

pensive. On the other hand, an autoencoder (AE) only needs to be trained once on an anomaly-free dataset to produce a model of "normality". This mathematical model can then process test data at a much faster rate than the LOF.

Autoencoders have seen an increase in use as a means to automatically generate features or reduce dimensions (Yan & Yu, 2015), and also anomaly (Sakurada & Yairi, 2014), or fraud (Paula, Ladeira, Carvalho, & Marzago, 2016) detection. The AE we use (Arora et al., 2015) is a form of Feed Forward Neural Network (NN) with a single hidden layer. While this architecture is relatively simple, it is sufficient thanks to the wavelet transform we have applied. In the future, we plan on experimenting with "recurrent" AEs, to handle the time dimension in our data. More complex architectures with additional hidden layers and convolution units (*deep* nets) also have the ability to automatically generate features (Martinelli, Tronci, Dipoppa, & Balducelli, 2004). Future research will focus on these methods, with the aim of removing the need for the initial wavelet transform.

A Feed-Forward NN has a layer of hidden neurons that performs non-linear transformations of inputs, and forwards the result towards the output neurons. The network identifies the correct transformations through an iterative training phase: samples from a training set are run through the NN, and the error is the difference between the NN's output, and the answer for that sample (found in the training set). Learning occurs through "back-propagation" of this error: each neuron's transformation is slightly adjusted to reduce that error. As this process is repeated over a large training dataset, the error reaches a minimum and the neural network is said to have converged.

While most NNs are used for classification and regression tasks, an autoencoder differs from typical NNs in that it aims to output an exact copy of the n input values on the output side. Figure 7 shows a typical AE architecture: to prevent the hidden layer from merely learning the identity function, the number of hidden neurons h is constrained below the number of input/output n. This effectively requires the autoencoder to discover and leverage relationships in the input data to compress the n input features into the h hidden neurons, by discarding redundant information. In the context of our application, imagine a set of three wavelet scales: W_1, W_2 and W_3 were found to have equal power values for normal spray activations. The autoencoder would only forward the power of wavelet scale W_1 to the hidden layer and discard W_2 and W_3. Knowing the values are supposed to be equal, reconstructing W_2 and W_3, from W_1 is straightforward. Assume we now run an abnormal record through this autoencoder. This record is abnormal as it presents some different relationship between the tree wavelets, where $W_1 \neq W_2 \neq W_3$. As the autoencoder only forwards $W1$, it will effectively discard information on the differences in W_2 and W_3, and rebuild them as being all equal. A mismatch between the input values, and the reconstructed values will occur.

4.2.1. implementation

We use the property illustrated in the previous example to detect anomalies: specifically, we compute the RMS reconstruction error of each input/output pair, and use their mean as a metric for anomaly. Once the model is trained, running test samples through the AE is fast, especially compared to the LOF. This speed advantage lets us use all 95 wavelet scales.
An inherent challenge with AEs is to find the optimal number of hidden neurons. Provided with too many hidden neurons, the AE does not need to compress information, and is able to memorize all details of the sample and reproduce it error-free – whether it is anomalous or normal. On the other hand, too little hidden neurons forces the AE to over-compress the sample and discard valuable information, preventing the reconstruction of *any* sample. In both cases, the reconstruction performance for normal and abnormal samples becomes indistinguishable. Through grid search, we found an AE with a single hidden layer of 95-20-95 neurons had the best discriminative performance.

Finally, we observed an increase in sensitivity when we purposefully "corrupted" our training dataset. For each wavelet scale, we randomly permuted 50% of the values. While the complexity of the AE makes it virtually impossible to conclusively determine how this counter-intuitive step improves sensitivity, we can formulate some hypothesis. The leaks we simulated resulted in a greater amplitude in air usage, yet the overall pattern remained the same. We can therefore assume the *groups* of wavelet scales that represent this pattern all increased their power in same proportions. Let us recall the autoencoder compresses data by forming relationships between groups of inputs, so if an abnormal sample exhibits similar group relationships – only with higher *amplitudes* – the sample may be reconstructed with little error. This is in contrast with the LOF that monitors the amplitude of each feature independently. If groups with related values are easy to identify, the AE will map each group of inputs to a single hidden neuron, and effectively process each group independently from one another. Corrupting the input may have made these strongly-interrelated groups harder to discern. The AE has

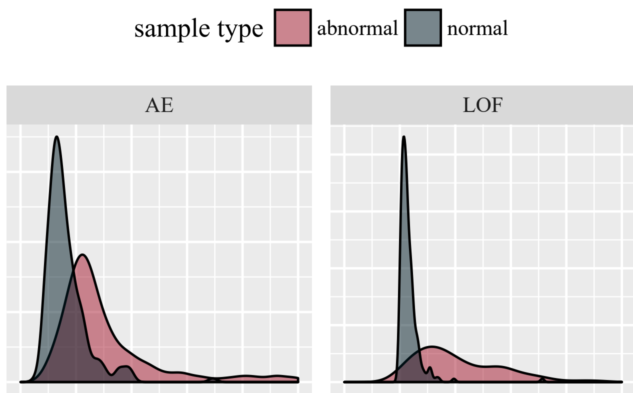

Figure 8. Density of the scores for anomalies and normal samples, for each algorithm. The densities have been rescaled to facilitate shape comparison, therefore we dropped units and axis

to "mine" the training dataset deeper: instead of forming few groups with strong relationships, if has to create larger, overlapping groups involving weaker relationships. This is an advantage, as it makes the AE look at the overall pattern, instead of processing segregated groups of inputs in isolation.

5. RESULTS

We present on Figure 8 the distributions of scores from each algorithm, differentiated by sample type (faulty or non-faulty). Each sample is the algorithm's mean score across the 40 seconds of the spray activation. In an ideal scenario, the distributions of scores for positive and negative cases show have no overlap. This would allow us to draw a vertical line (decision threshold) on Fig. 8 that would perfectly divide both densities. This is clearly not the case: instead, the decision threshold will have to be a compromise. Setting the decision threshold:

Towards the lower end of the overlap region will detect more positive cases, but also generate false positives as the right tail of the *normal case* distribution extends past our threshold.

Towards the higher end of the overlap region will report less false positives, as most *normal cases* will be lower than the decision threshold. However, we will miss some abnormal cases (false negatives) that fall below the decision threshold.

A Receiver Operating Curve (ROC) is the best way to illustrate the compromises in selecting an optimal decision threshold. Figure 9 shows the ROC curve where each point on a line correspond to a possible decision threshold. All curves start at the bottom-left corner, and correspond to a decision threshold set infinitely high: the algorithm is not sensitive at all, and systematically labels each case as being negative. Consequently, there are no false positives, but positive cases

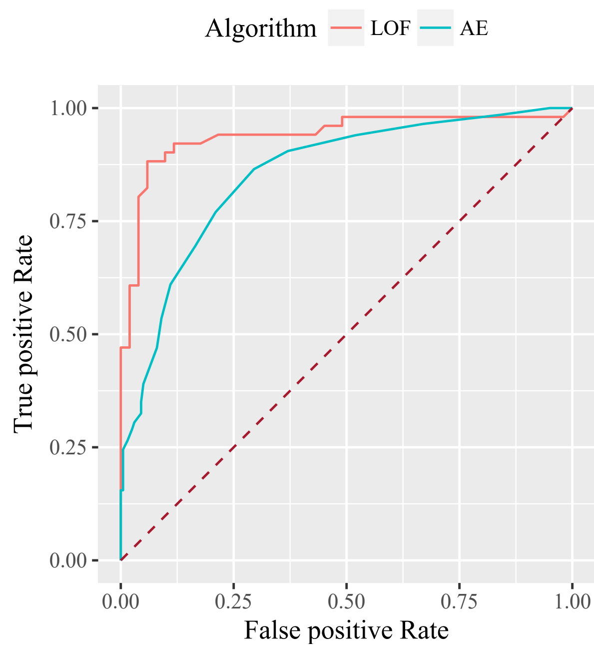

Figure 9. Receiver Operating Curve (ROC) of the two algorithms presented

are never reported as such either. The curves end at the top-right corner, where the decision threshold is set infinitely low: the algorithm is too sensitive and labels each case as a positive. Inherently, it can never miss a positive case, but all other cases become false positives. The red dashed diagonal line shows the performance of an algorithm performing random guesses, with the probability of "positive" ranging from zero to one. The optimum performance is therefore the top-left corner: where the algorithm detects all positive cases, but never reports false positives. Based on this, it is clear that the LOF algorithm outperforms the AE. The Area Under the Curve (AUC) is a single metric that lets us compare the overall performance of both algorithm. An ideal curve reaches the top left corner, and its AUC is one. A random guess (diagonal) has an AUC of 0.5. The AE algorithm here has an AUC of 0.83, and the LOF's is 0.94. This confirms the advantage of the LOF algorithm.

6. CONCLUSION

In this paper, we presented an anomaly detection technique for leaks in compressed air systems. We first presented and validated the wavelet transform as input transformation. This step also helped us identify the surprisingly small number of wavelet scales required to effectively represent patterns of air consumption. We then performed unsupervised anomaly detection on the time-series, comparing a density-based clustering algorithm (LOF) and an autoencoder. Both algorithms outperformed the "random guess" baseline by a large margin. We found the LOF algorithm outperformed the AE. However, the LOF's sensitivity came at the cost of processing one sample at a time, and each iteration involved processing the entire reference dataset that descibed "normal". This would make real-time processing impossible on common computing platforms. A variation in the LOF algorithm implementation could improve this, where the training set would only be evaluated once and memorised; following this, only the densities in the neighbourhood of the test points would need to be evaluated. While the AE underperformed, it's speed could have allowed us to add even more inputs, such as the derivatives of each wavelet scales, to possibly increase sensitivity. Since there is no definitive procedure to parameterise AEs, we performed a grid search for the optimal parameters. Our search could have also missed a set of parameters that would have increased performance. We also found that the AE's performance increased when we "corrupted" its inputs: the AUC went from 0.7 to 0.83. We believe this step forced the AE to encode the data beyond "superficial" relationships: instead of checking groups of inputs in isolation, it considered the samples as a whole. However the complexity of the AE makes it difficult to setup an experiment to verify this hypothesis.

Ultimately, we believe the LOF outperformed the AE as the LOF algorithm considers the amplitudes of each feature of the samples when calculating the density ratio with anomaly-free clusters. In contrast, the AE detects anomalies through violations of relationships that existed in the training data. There is a possibility that some anomalous samples still presented similar relationships, though at abnormal amplitudes.

We are currently running some experiments on real machine data, and the preliminary results indicate the LOF algorithm detects permanent leaks, and leaking air brake chambers. In this process, we also noted how frequently the air system changes. Indeed, the probability that a maintenance event occurs at any point in time on the air system as a whole is large, simply because of its sheer scale. Maintenance activities may change the signature of some components, which require a re-training of the anomaly detection system. The next stage of our research should also investigate the minimum required amount of training data: this will minimise the anomaly detection algorithm's necessary downtime, while a new training dataset is being recorded.

REFERENCES

Arora, A., Candel, A., Lanford, J., LeDell, E., , & Parmar, V. (2015, August). Deep learning with h2o [Computer software manual]. Retrieved from `http://h2o.ai/resources`

Bevenot, X., Trouillet, A., Veillas, C., Gagnaire, H., & Clement, M. (2000). Hydrogen leak detection using an optical fibre sensor for aerospace applications. *Sensors and Actuators B: Chemical*, *67*(1), 57 - 67. doi: http://dx.doi.org/10.1016/S0925-4005(00)00407-X

Boecking, B., Chalup, S. K., Seese, D., & Wong, A. S. (2014). Support vector clustering of time series data with alignment kernels. *Pattern Recognition Letters*, *45*, 129–135.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. doi: 10.1023/A:1010933404324

Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). Lof: Identifying density-based local outliers. In *Proceedings of the 2000 acm sigmod international conference on management of data* (pp. 93–104). New York, NY, USA: ACM. doi: 10.1145/342009.335388

Callegari, C., Gazzarrini, L., Giordano, S., Pagano, M., & Pepe, T. (2014). Improving pca-based anomaly detection by using multiple time scale analysis and kullbackleibler divergence. *International Journal of Communication Systems*, *27*(10), 1731–1751. doi: 10.1002/dac.2432

Colombo, A. F., Lee, P., & Karney, B. W. (2009). A selective literature review of transient-based leak detection methods. *Journal of Hydro-environment Research*, *2*(4), 212 - 227. doi: http://dx.doi.org/10.1016/j.jher.2009.02.003

Cooper, P. S., Wong, A. S., Fulham, W. R., Thienel, R., Mansfield, E., Michie, P. T., & Karayanidis, F. (2015). Theta frontoparietal connectivity associated with proactive and reactive cognitive control processes. *Neuroimage*, *108*, 354–363.

Datta, S., & Sarkar, S. (2016). A review on different pipeline fault detection methods. *Journal of Loss Prevention in the Process Industries*, *41*, 97 - 106. doi: http://dx.doi.org/10.1016/j.jlp.2016.03.010

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In (pp. 226–231). AAAI Press.

Fuchs, H., & Riehle, R. (1991). Ten years of experience with leak detection by acoustic signal analysis. *Applied acoustics*, *33*(1), 1–19.

Gelman, L., Patel, T. H., Persin, G., Murray, B., & Thomson, A. (2013). Novel technology based on the spectral kurtosis and wavelet transform for rolling bearing diagnosis. *International Journal of Prognostics and Health Management, ISSN*, 2153–2648.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction* (2nd ed.). Springer.

Hodge, V., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial intelligence review*, *22*(2), 85–126.

Huang, C. T., Thareja, S., & Shin, Y. J. (2006, Aug). Wavelet-based real time detection of network traffic anomalies. In *2006 securecomm and workshops* (p. 1-7). doi: 10.1109/SECCOMW.2006.359584

Khan, M. M., Chalup, S. K., & Mendes, A. (2016). Parkinsons disease data classification using evolvable wavelet neural networks. In *Australasian conference on artificial life and computational intelligence* (pp. 113–124).

Kriegel, H.-P., Krger, P., Sander, J., & Zimek, A. (2011). Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *1*(3), 231–240. doi: 10.1002/widm.30

Liaw, A., & Wiener, M. (2002). Classification and regression by randomforest. *R News*, *2*(3), 18-22. Retrieved from `http://CRAN.R-project.org/doc/Rnews/`

Martinelli, M., Tronci, E., Dipoppa, G., & Balducelli, C. (2004). Electric power system anomaly detection using neural networks. In M. G. Negoita, R. J. Howlett, & L. C. Jain (Eds.), *Knowledge-based intelligent information and engineering systems: 8th international conference, kes 2004, wellington, new zealand, september 20-25, 2004, proceedings, part i* (pp. 1242–1248). Berlin, Heidelberg: Springer Berlin Heidelberg.

Montgomery, D. (2005). *Introduction to statistical quality control*. Hoboken, N.J: John Wiley.

Murvay, P.-S., & Silea, I. (2012). A survey on gas leak detection and localization techniques. *Journal of Loss Prevention in the Process Industries*, *25*(6), 966 - 973. doi: http://dx.doi.org/10.1016/j.jlp.2012.05.010

Paula, E. L., Ladeira, M., Carvalho, R. N., & Marzago, T. (2016, Dec). Deep learning anomaly detection as support fraud investigation in brazilian exports and anti-money laundering. In *2016 15th ieee international conference on machine learning and applications (icmla)* (p. 954-960). doi: 10.1109/ICMLA.2016.0172

Perzyk, M., Kochanski, A., Kozlowski, J., Soroczynski, A., & Biernacki, R. (2014). Comparison of data mining tools for significance analysis of process parameters in applications to process fault diagnosis. *Information Sciences*, *259*, 380 - 392. doi: https://doi.org/10.1016/j.ins.2013.10.019

Qin, S. J. (2012). Survey on data-driven industrial process monitoring and diagnosis. *Annual Reviews in Control*, *36*(2), 220–234.

Ramirez, A., Daily, W., Binley, A., LaBrecque, D., & Roelant, D. (1996). Detection of leaks in underground storage tanks using electrical resistance methods. *Journal of Environmental and Engineering Geophysics*, *1*(3), 189-203. doi: 10.4133/JEEG1.3.189

Sakurada, M., & Yairi, T. (2014). Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the mlsda 2014 2nd workshop on machine learning for sensory data analysis* (pp. 4:4–4:11). New York, NY, USA: ACM. doi: 10.1145/2689746.2689747

Sandberg, C., Holmes, J., McCoy, K., & Koppitsch, H. (1989, Sep). The application of a continuous leak detection system to pipelines and associated equipment. *IEEE Transactions on Industry Applications*, *25*(5), 906-909. doi: 10.1109/28.41257

Sarkar, S., Reddy, K. K., Giering, M., & Gurvich, M. R. (2016). Deep learning for structural health monitoring: A damage characterization application. In (Vol. 7). Prognostics and Health Management Society.

Seungmin, L., Gisung, K., & Sehun, K. (2011). Self-adaptive and dynamic clustering for online anomaly detection. *Expert Systems with Applications*, *38*(12), 14891 - 14898.

Song, J., Takakura, H., Okabe, Y., & Nakao, K. (2013). Toward a more practical unsupervised anomaly detection system. *Information Sciences*, *231*, 4 - 14. (Data Mining for Information Security)

Wong, A. S., Chalup, S. K., Bhatia, S., Jalalian, A., Kulk, J., Nicklin, S., & Ostwald, M. J. (2012). Visual gaze analysis of robotic pedestrians moving in urban space. *Architectural Science Review*, *55*(3), 213–223.

Yan, W., & Yu, L. (2015). On accurate and reliable anomaly detection for gas turbine combustors: A deep learning approach. In *Proceedings of the annual conference of the prognostics and health management society.*

Zhang, G., Kinsner, W., & Huang, B. (2009). Electrocardiogram data mining based on frame classification by dynamic time warping matching. *Computer methods in biomechanics and biomedical engineering*, *12*(6), 701–707.

Zhao, J., Liu, K., Wang, W., & Liu, Y. (2014). Adaptive fuzzy clustering based anomaly data detection in energy system of steel industry. *Information Sciences*, *259*, 335 - 345.

BIOGRAPHIES

Antoine F.D. Desmet received his undergraduate degree in electronics engineering from ESEO, Angers, France; Master by research from the University of Wollongong, Australia; and PhD from Imperial College London, UK. He is currently working for Komatsu Mining Corp., applying Machine Learning tonot sure anomaly detection on mining machinery.

Matthew Delore received his undergraduate degree in Computer Engineering at The University of Newcastle, Australia. He currently provides solutions to the financial services sector using machine learning and robotic process automation. His research interests include natural language processing and anomaly detection.