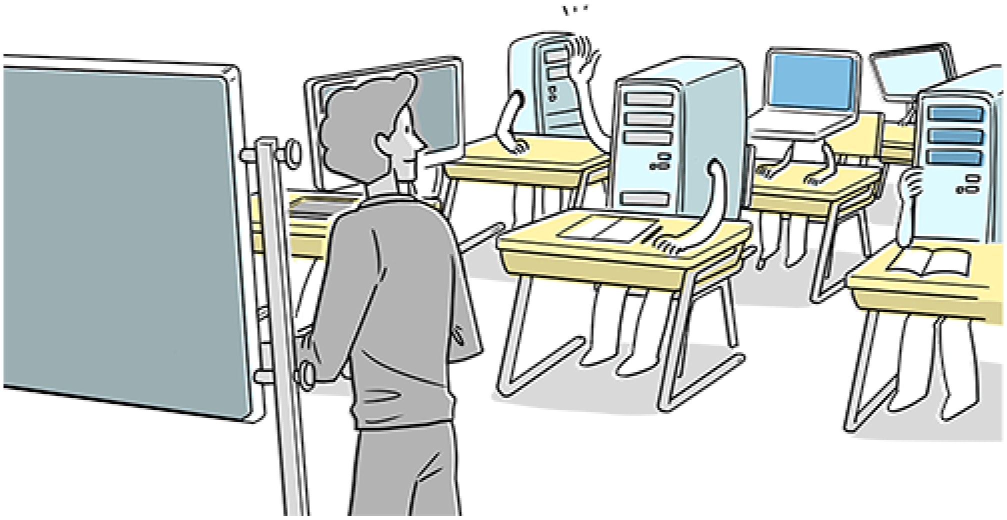
A Machine Learning approach to Ultrasonic Flaw Detection



Nick Root
nick.jes@autservices.com

January 2017

Abstract

Pipeline inspection firms work in remote and hostile environments. Teams follow weld crews along pipelines inspecting the work in near real time, so that a faulty weld can be redone at the lowest cost possible.

The pipes in question are often being built for high pressure oil and gas, so the stakes are high. Our state of the art, custom built ultrasound scanners are operated by crews of men that visually inspect the ultrasound data on laptops to make fast decisions on the quality of the work.

You may know from my other work that I believe in augmented decision making. It's not that humans always make bad decisions, but they are prone to very predictable biases and can be improved with the help of machines.

With such high stake decisions, and a whole crew of people out in the middle of nowhere when they could be someplace else, I set out to study the hardware and data to see how predictable weld flaws really are. The results were very positive

Introduction to the method

Support Vector Machines (SVM)

I became interested in this problem when I saw the complexity of the signal processing techniques required to get the ultrasound scans into human readable form, particularly when high scattering noise (clutter) overwhelms the flaw echoes.

I noticed that what the workers were reading was essentially just a series of linear data in various vector spaces: this was essentially pattern recognition and 5 years of labelled scan data (faulty/non faulty) was archived and available for model training, albeit after a lot of data cleansing and preparation.

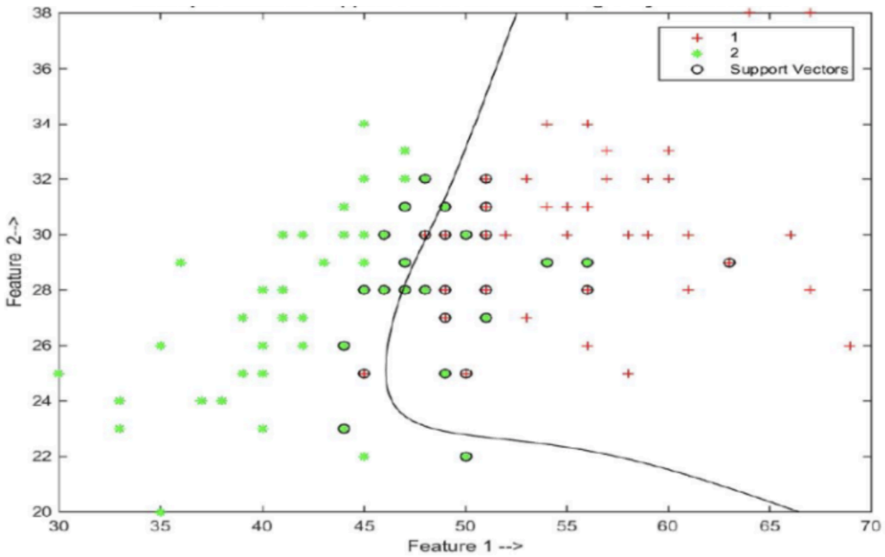
This dataset was very imbalanced, with over 95% of all weld scans being non-faulty. As I learned from dealing with very extreme data imbalance with fraud problems, the SVM machine learning technique handles sparse and imbalanced data very well, by assigning higher misclassification penalties to training instances of the minority class. This therefore seemed like a good place to start.

SVM is a machine learning method used for classification and regression analysis of complex real-world problems that may be difficult to analyze theoretically. Typically, SVM has been used for pattern recognition but recently it has been applied to other fields such as financial market analysis[1], electric utility forecasting[2], weather forecasting[3] and also ultrasonic NDE[4].

[1] Y . Liang and Y . Sun, "An improved method of support vector machine and its application to financial time series forecasting," *Prog. Natural Sci.,* vol. 13, no. 9, pp. 696–700, 2003.
[2] B.-J. Chen, M.-W. Chang, and C.-J. Lin, "Load forecasting using support vector machines: A study on EUNITE competition 2001," IEEE Trans. Power Syst., vol. 19, no. 4, pp. 1821–1830, Nov. 2004
[3] H. Prem and N. R. Srinivasa Raghavan, "A support vector machine based approach for forecasting of network weather services," *J. Grid Computing*, vol. 4, no. 1, pp. 89–114, Mar. 2006.
[4] D. Isa and R. Rajkumar, "Ultrasonic Sensor Data Processing using Support Vector Machines", in *The First International Workshop on Nonlinear Dynamics and Synchronization (INDS08)*, 2008

SVM techniques deal with classifying a given data to its respective groups (classes). Typically, they are used in circumstances where there are only two decisions to be made, but they can be used in cases of multiple decision scenarios with variants of one vs. one group classifier or many vs. one classifier. Irrespective of the kind of SVM classifier, the primary work of the algorithm is to find a boundary to divide the data groups as shown in Fig. 1.

Figure 1: Example of a SVM using a Polynomial Kernel



Furthermore, SVM methods differ depending on the kind of decision boundary used - linear and nonlinear classifier. The concept of non-linearity is brought into play by using kernels which convert the data points from one dimensionality to another. Around these boundaries are the data samples which are used as margins or supports for the decision boundary.

These margins might follow a strict rule such that no other data samples lie between the margin and the boundary or they may allow some marginal errors to occur, hence depending on the type of margin used they are classified as hard and soft margins respectively. Generally soft margin is used for all the classification as it gives the freedom of choosing the support vectors with more degree of freedom bringing in the concept of generality and this degree of freedom is controlled by a parameter in the design equation. SVM needs data from two different groups for training to get the equation which can make a decision. Equation (1) describes the dataset belonging to both the classes

$$D = \{(x_i, y_i), x_i \in R \; and \; y_i \in +1 \; or -1\} \qquad (1)$$

where xi represent the data vector and yi represents the class to which each data *i* belongs. Suppose y% is either +1 or -1 (the labels for each class) then the equation that can find the decision boundary is given as Equation (2) which gives a complete description on the combined hyperplanes of both the labels.

$$y(Wx - b) \geq 1 \qquad (2)$$

In equation (2), *W* is a vector normal to the hyper-plane equation and *b* is the offset from the origin. W is found using quadratic programming and *b* is found back by using the values of x, y and W which was found in the previous step. The quadratic programming uses Lagrange multiplier, α, for optimization, which is one of the parameters used in algorithm. The condition for the quadratic programming is to maximize α. There are two other parameters C and ξ which accounts for soft margin optimization problem as shown in Equation (3), with the constraint given in Equation (4). The rest of the algorithm for finding the W remains the same but it needs to be noted that the equation for the Lagrange's multiplier optimization changes.

$$\arg min_{W,\xi,b} \{\tfrac{1}{2} \|W\|^2 + C \sum \xi\} \qquad (3)$$

$$0 \le \alpha \le C, \xi \ge 0 \qquad (4)$$

There are two phases, training and testing, in any machine learning application. There must be enough data samples to undertake both training and testing. Training also involves an intermediate step of validating so that the equation obtained is more generalised. If the dataset is not representative of the real world, then you get the common issue of generalisation error. This together with 'overfitting' your training data are the most common machine learning issues.

Feature selection

The feature selection process involves choosing which data fields the algorithm will use to try to predict the output. The previous ultrasound SVM applications referenced above use A-scan data itself or ultrasonic signal parameters such as time-of- arrival, center frequency and bandwidth as feature vectors. I went a step further by using subband filter outputs (i.e. split spectrum processing, SSP[5]) as feature vectors for the SVM in order to exploit the frequency diversity among the flaw and clutter echoes (see below). This I believe is a totally novel approach.

Split spectrum processing (SSP)

SSP is helpful in extracting frequency diverse information of A-scan data in the form of different frequency bands. There are three important steps employed in SSP implementation. In the first step, time domain signal is converted to frequency domain signal. The next step decomposes the signal spectrum into multiple channels (subbands) using Gaussian filters. The final step involves reconstructing the filtered frequency domain signals into time domains using post-processing methods such as order statistics.

The number of channels needed, the width of each channel and the width of overlapping can be varied to get the best possible waveforms in which the clutter noise is removed in almost all of the low frequency bands. The performance of both SSP and post processing algorithm mainly depends on factors such as the correlation of signals between different bands, statistical information shared among the different channels and the number of channels which are used to analyze the signal.

[5] J. Saniie, D. Nagle and K. Donohue, "Analysis of order statistic filters applied to ultrasonic flaw detection using split spectrum processing", *IEEE Trans. Ultrason. Ferroelectr. Freq. Control,* vol. 38, no. 2, pp.133 -140, 1991.

Since most of the information is trapped in the lower frequency sections of the signals, number of channels is typically limited to 8 or less. It can also be noted that the information related to the flaw is more pronounced and correlated among the lower frequency bands.

Training and optimisation

The above sections have introduced the key components that will be used in the algorithm. This section describes the procedure for training and the optimization of the SVM parameters.

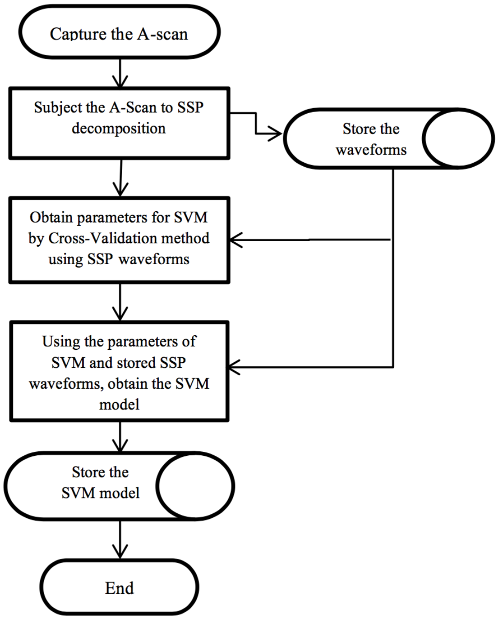
The raw ultrasonic signal (A-scan data) is subjected to Split Spectrum Processing before it can be analysed. This decomposition involves decomposing A-Scan into different frequency bands as explained in the Split Spectrum Processing. Following the SSP decomposition, the output of each subbands are concatenated to form the input feature vector for the SVM. The inputs for the training are stored. Some examples in this stored input are used for cross validation.

There are different methods in cross-validation; namely *Exhaustive* and *Non-exhaustive* cross-validation. In this paper, a Non-exhaustive cross-validation method is used; specifically *k*-fold cross-validation which is one of the most prevalent methods. In *k*-fold cross-validation, the dataset is randomly grouped in k groups and one group is chosen as validation dataset, while the remaining k-1 group is used for training.

Similarly, this procedure is repeated k times. An important fact is that all the groups have almost same number of examples from both the classes (i.e equal number of bad and good welds). The advantage is that no two examples are repeated either in testing or in validation, hence giving a close approximation of the real time accuracy. The parameters which give the maximum cross-validation can be chosen to make the final model of SVM training.

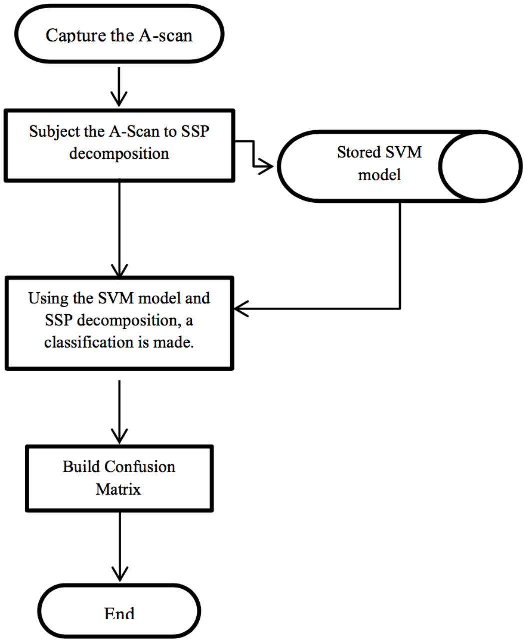
Next, all the training data which was used for validation and the unused data are combined together to create the final SVM model with the parameters which was obtained from k- fold cross validation. The model which was obtained must be stored. All the steps above constitute the training phase. A flow chart of training is given in Fig.2.

Figure 2: Flowchart for training with the proposed SVM classifier


Capture the A-scan
Subject the A-Scan to SSP decomposition
Store the waveforms
Obtain parameters for SVM by Cross-Validation method using SSP waveforms
Using the parameters of SVM and stored SSP waveforms, obtain the SVM model
Store the SVM model
End

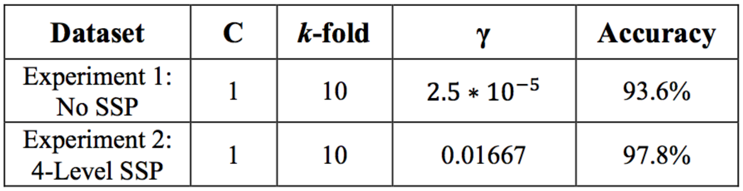
After the model is obtained, it is subjected to testing with real data. In testing, remaining unused signals (that were not used for training) are subjected to SSP decomposition and the time-domain signals corresponding to different frequency bands are concatenated to form the feature vector input for SVM model. SVM model (generated during the training phase) predicts the class to which the given signal belongs to. A flow chart of testing is given in Fig. 3.

Figure 3: Flowchart for testing with the proposed SVM classifier


Capture the A-scan
Subject the A-Scan to SSP decomposition
Stored SVM model
Using the SVM model and SSP decomposition, a classification is made.
Build Confusion Matrix
End

The training and the testing algorithm has been implemented using LIBSVM-3.2 library which provides options for both classification and regression. The Cross validation parameters (shown in Table I for different experiments described in the next section) to be determined include *C* (penalty parameter of the error term), γ which is the Radial Basis Function (RBF) kernel parameter used for transformation of the input feature space and *k.* The value of *k* is varied to get the best accuracy. *k* equal to 10 provides the most robust performance.

Table 1: Selection of cross validation parameters



Dataset	C	*k*-fold	γ	Accuracy
Experiment 1: No SSP	1	10	$2.5 * 10^{-5}$	93.6%
Experiment 2: 4-Level SSP	1	10	0.01667	97.8%

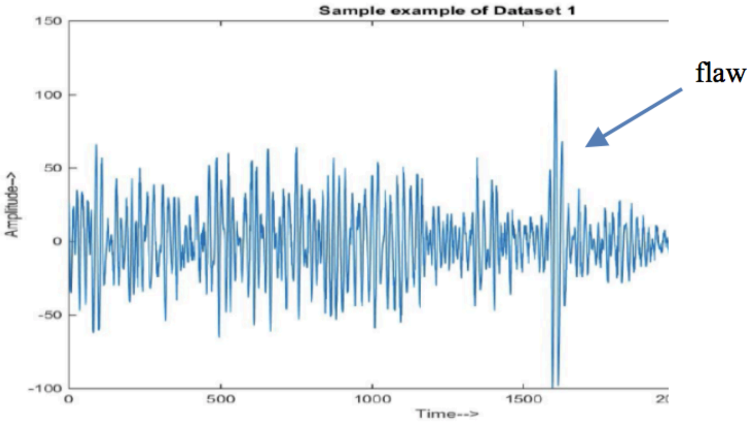
Experimental setup and results

Multiple experiments were conducted to verify the effectiveness of the SVM based flaw detection algorithm.

The dataset has 282 A-scan signals acquired from a steel block (type 1018, grain size 50μm). Steel block has several holes (1.5mm diameter) at different but known locations. All A- scan data have been captured by a broadband ultrasonic transducer (Panametric A3062) of 0.375inch diameter with 5 MHz center frequency at a sampling rate of 100 MHz with 1024 samples per A-scan.

Signal-to-noise ratio (SNR) of Dataset lies in the range of 1db to 10db. Fig. 4 shows a sample A-Scan signal from the dataset. Approximately half (140/282) of the dataset is used for training and the remaining dataset is used for testing.

Figure 4: Simple A-scan data from dataset



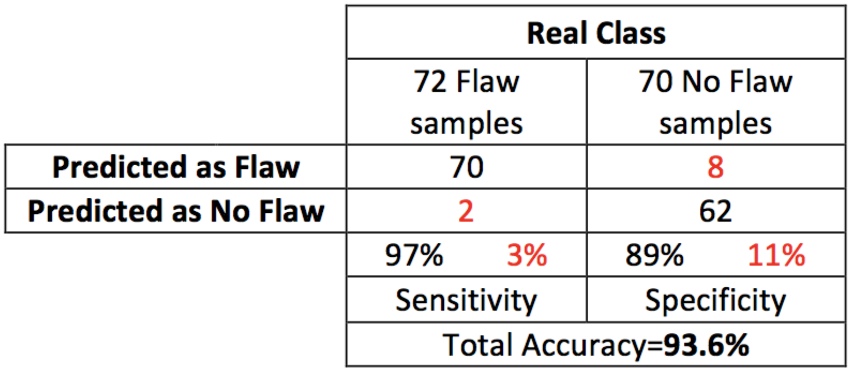
In Experiment 1, no SSP decomposition is done. Hence, A-scan data is directly used as feature vector in the SVM model. In the Experiment 2, dataset has been subjected to SSP decomposition of 4-channels (subbands) as feature vector input to SVM in order to investigate the robustness of the algorithm**.**

Results are presented with respect to the Confusion matrix (see table 2 and 3)

Table 2 shows the results of the Experiment 1. SVM model created for dataset without applying SSP decomposition has an accuracy of 93.6%. Specificity is 89%.

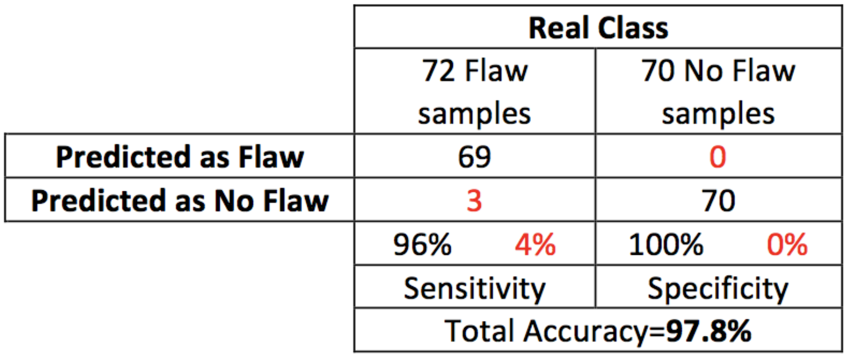
Table 3 represents the Confusion matrix for the dataset being subjected to 4-level SSP sub-band decomposition corresponding to Experiment 2. It shows overall detection accuracy of 97.8%. Specificity is 100% which means no false detection occurs as opposed to 8 false detection of flaws in Experiment 1.

Table 2: Confusion matrix for Experiment 1 (no SSP)



	Real Class	
	72 Flaw samples	70 No Flaw samples
Predicted as Flaw	70	8
Predicted as No Flaw	2	62
	97% 3%	89% 11%
	Sensitivity	Specificity
	Total Accuracy=**93.6%**	

Table 3: Confusion matrix for Experiment 2 (4-channel SSP)



	Real Class	
	72 Flaw samples	70 No Flaw samples
Predicted as Flaw	69	0
Predicted as No Flaw	3	70
	96% 4%	100% 0%
	Sensitivity	Specificity
	Total Accuracy=**97.8%**	

Conclusion

The results confirm that SSP can be used as a feature vector input for the SVM and can provide improved accuracy over the direct implementation in which no subband decomposition is applied. In particular, false flaw detections are significantly reduced, this is key in a real world environment.

Performance of the algorithm is promising and provides a pathway for further investigation of SVM based regression analysis. Other types of feature vectors can also be considered for improving the performance of the classifier. Another research direction would be application of the proposed algorithm in determining and characterizing different types of flaw echoes.

In terms of practical application. I recommend further testing on a larger volume of flaws, then further work needs to be done to investigate how the method can applied to live weld testing. The algorithm could potentially be coded into the firmware of the scanner, or more likely, built into an application that can be deployed on the operators laptop.