

Elastic impedance based facies classification using support vector machine and deep learning

Nishitsuji, Yohei; Exley, Russell

DOI

[10.1111/1365-2478.12682](https://doi.org/10.1111/1365-2478.12682)

Publication date

2018

Document Version

Final published version

Published in

Geophysical Prospecting

Citation (APA)

Nishitsuji, Y., & Exley, R. (2018). Elastic impedance based facies classification using support vector machine and deep learning. *Geophysical Prospecting*, 67 (2019)(4), 1040-1054.
<https://doi.org/10.1111/1365-2478.12682>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Elastic impedance based facies classification using support vector machine and deep learning

Yohei Nishitsuji^{1,2*} and Russell Exley³

¹Petro Summit E&P Corporation, Tokyo, Japan, ²Department of Geoscience and Engineering, Delft University of Technology, Delft, The Netherlands, and ³Summit Exploration and Production Limited, London, United Kingdom

Received October 2017, revision accepted August 2018

ABSTRACT

Machine learning methods including support-vector-machine and deep learning are applied to facies classification problems using elastic impedances acquired from a Paleocene oil discovery in the UK Central North Sea. Both of the supervised learning approaches showed similar accuracy when predicting facies after the optimization of hyperparameters derived from well data. However, the results obtained by deep learning provided better correlation with available wells and more precise decision boundaries in cross-plot space when compared to the support-vector-machine approach. Results from the support-vector-machine and deep learning classifications are compared against a simplified linear projection based classification and a Bayes-based approach. Differences between the various facies classification methods are connected by not only their methodological differences but also human interactions connected to the selection of machine learning parameters. Despite the observed differences, machine learning applications, such as deep learning, have the potential to become standardized in the industry for the interpretation of amplitude versus offset cross-plot problems, thus providing an automated facies classification approach.

Key words: Elastics, Interpretation, Inversion, Reservoir geophysics, Rock physics.

INTRODUCTION

Machine learning applications are already becoming increasingly widespread in a variety of data-driven industries ranging from financial services to life sciences. The data-rich environment of the oil and gas industry is also becoming an advocate of this emerging technology. De-risking of exploration and development opportunities, using quantitative–interpretation driven by amplitude versus offset (AVO) workflows, is regarded as a crucial step in developing hydrocarbon resources. The quantitative nature of this work often involves the classification of facies using multiple elastic impedances and is therefore ideally suited to machine learning algorithms.

Machine learning algorithms, such as the support vector machine (SVM; Vapnik and Learner 1963; Vapnik 1995) method, have been used extensively in fields such as

pattern recognition. There are two main problem-solving capabilities to SVM: classification problems (Vapnik 1995) and regression problems (Smola and Schölkopf 2004). In this regard, SVM basically consists of three main elements: linear neural networks (NN), kernel-tricks (Schölkopf *et al.* 1999) and regularizations. The linear NN part consists of two layers (input and output) and broadly imitates the neuron to synapse model of a biological brain. The NN works with the maximal-margin approach that attempts to maximize the margin (most often Euclid distance) between data groups and their decision boundaries in the feature domain. With the kernel-tricks, in addition to linear decisions, non-linear decisions can also be handled. Finally, the regularization helps avoid overfitting problems, which when combined with the NN and kernel-tricks provides the general procedure for most SVM applications. SVM's simplicity in implementation provides the main reason as to why it is easily adapted in many applications.

*E-mail: yohei.nishitsuji@gmail.com

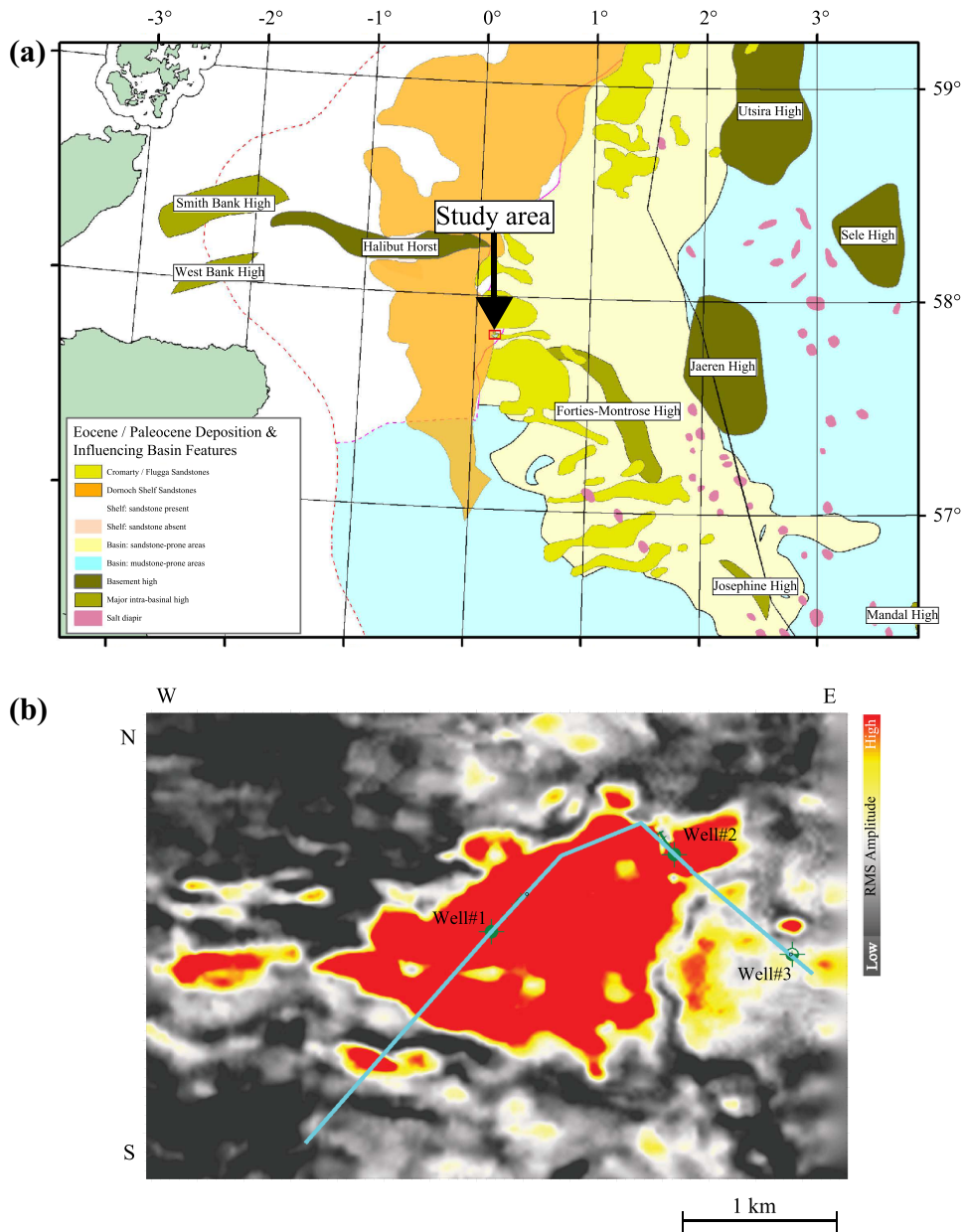


Figure 1 (a) Regional map showing study area and the deposition of Dornoch, Cromarty and basin wide Paleocene Forties sandstones. The depositional extents are modified from Mudge (2014). The field data used in this study is centred on the Avalon discovery in Block 21/6b. Avalon's reservoir is contained within the Cromarty Member that is predominantly a late stage depositional system restricted to the basin margins. (b) RMS amplitude through the reservoir interval of the Avalon discovery. The line of section, as depicted in Figures 11 and 12, is shown along with the locations of Wells #1, #2 and #3.

Deep learning (DL; Hinton, Osindero and Teh 2006; Hinton and Salakhutdinov 2006), on the other hand, provides a more recent advancement over SVM methods. In the broad definition, DL consists of a multi-layer perceptron, input layer(s), hidden layer(s) and output layer(s), which collectively solves problems (e.g. classification) by automated

feature extraction. The perceptron is a self-training algorithm for classifying data (Rosenblatt 1958). The simple (single) perceptron solves linear problems while the multi-layer perceptron has the ability to solve non-linear problems. DL is considered a deep neural network rather than the already established systems such as convolutional neural networks

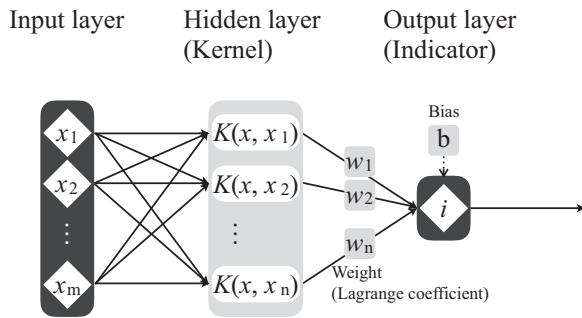


Figure 2 Schematic of the NN architecture used by the SVM approach in this study. The original data x in the input layer are classified with the indicator i , via the kernel K , with the weight w , and bias b .

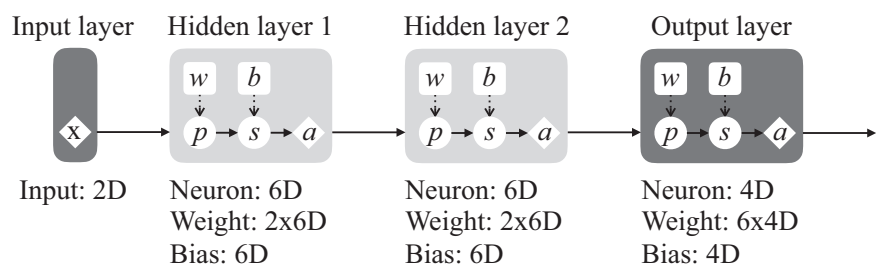
(CNN being mainly for image-recognition problems) or recurrent neural networks (used mainly for time-series problems such as speech recognition).

Past studies, specifically focusing on AVO, have predominantly used only SVM methods. Kuzma (2003) and Li, You and Liu (2015) presented an AVO regression problem using synthetic wells, whereas Li and Castagna (2004) introduced AVO cross-plot classification using SVM but again only using a synthetic dataset. In the following case study, we show the application of SVM and DL to a field dataset, obtained from a discovery in the UK Central North Sea, for the purpose of lithological classification using elastic impedance cross-plot products. Furthermore, we compare the results from SVM and DL with a more simplified linear projection and a Bayes-based classification approach (Zabihi Naeni and Exley 2017).

FIELD DATA

The case study shown in this paper centres on a Paleocene discovery, in Block 21/6b of the UK Central North Sea located at the north-western edge of the Central Graben, just south of the Buchan Field (Figure 1(a)). The discovery was initially identified using conventional, simultaneous, pre-stack, inversion, followed by an exploration well that successfully encountered a 26-m oil column in good quality sands. The

Figure 3 Schematic of the NN architecture used by the DL approach in this study. The original data x in the input layer are classified in the output layer via the hidden layers. p and s indicate multiplication and summation operations of the weight w , and the bias b , of the input. The activator is indicated as a . 6D, for example, indicates that it has six dimensions.



reservoir sands lie within the proximal part of the prolific northwest to southeast, late Paleocene, Forties and Cromarty depositional trends, which includes the giant Forties Field. Post discovery, the seismic data were re-inverted but on this occasion using a joint impedance–facies inversion (Kemper and Gunning 2014) to provide updated seismically derived elastic properties calibrated to the discovery well (Well #1 in Figure 1(b)).

The initial discovery well (Well #1) was used to provide the input rock physics data in order to train the support vector machine (SVM) and deep learning (DL) methods detailed in this paper. Post-training the equivalent elastic data (acoustic impedance, AI and v_p/v_s) output from the joint impedance–facies inversion was characterized in terms of potential facies using both SVM and DL. A further test was provided by the drilling of an appraisal well (Well #2 in Figure 1(b)), which along with an older well drilled outside of the discovery (Well #3 in Figure 1(b)) provided two ‘blind tests’ for the SVM and DL facies classification.

METHODOLOGY

The support vector machine approach

Structural risk minimization

Support vector machine (SVM) learning is based on two fundamental theories termed statistical learning theory and structural risk minimization (SRM; Vapnik 1995). Although statistical learning theory aids in finding the maximal-margin that corresponds to the solution of an optimization problem, SRM provides us with a trade-off between hypothesis and model complexity, which is called the Vapnik–Chervonenkis dimension (VC-dimension; Vapnik and Chervonenkis 1974). Since applications of SVM for exploration geophysics and rock physics are not familiar, a brief description with step-by-step equations follows. For a simple binary classification problem, let us assume that we have a training dataset, T ,

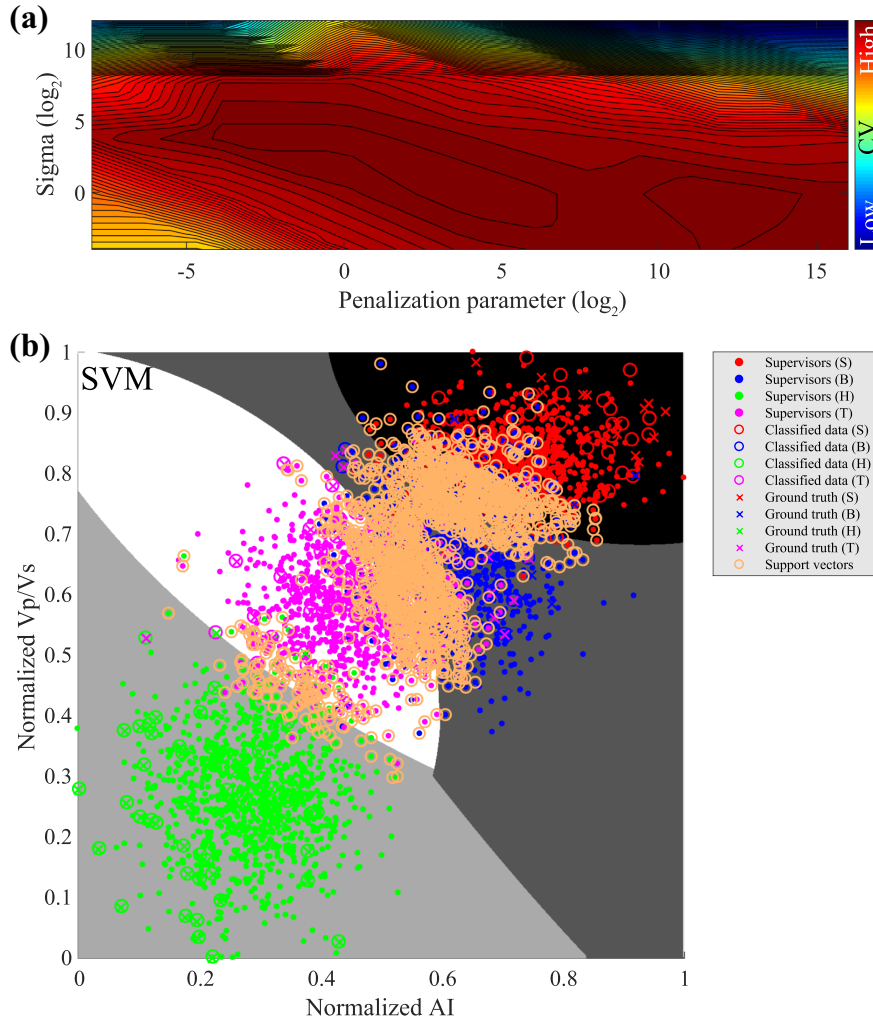


Figure 4 Synthetic elastic impedance cross-plot (AI and V_p/V_s) results from the SVM where (a) is the cross-validation (CV) of the hyperparameters and (b) is the classification. The filled circles are the supervisors (training data), the hollow circles are the classified (test data) and the crosses are the ground truth. The hollow orange circles are the support vectors that were used to establish the boundaries in this realization. S, B, H and T in the legend box stand for shale, brine-bearing sand, hydrocarbon-bearing sand and tuff facies. The four different background colours correspond with the classified facies labels. Confusion matrices of the training and test accuracies are shown in Figure 6(a) and (c).

which contains n th number of vectorized samples \mathbf{x}_i , and their associated labels y . Such a dataset can be written as:

$$T = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}, \quad \mathbf{x}_i \in \mathbb{R}^m, y_i \in \{0, 1\}. \quad (1)$$

We would like to find a learning function that classifies the unlabelled or unknown dataset. The number of feature dimensions in the input domain is m . In this case, the empirical risk $R_{\text{emp}}(f)$, which we would like to optimize in order to obtain the learning function, can be described as:

$$R_{\text{emp}}(f) = \frac{1}{N} \sum_{i=1}^n |f(\mathbf{x}_i) - y_i|, \quad (2)$$

where $|f(\mathbf{x}_i) - y_i|$ is, in theory, the so-called 0-1 loss function used to evaluate if the learning function $f(\mathbf{x}_i)$ correctly identifies the expected value of y_i and N is the total number of training datasets. The function $f(\mathbf{x}_i)$ that minimizes equation (2) over the training dataset is the learning function we eventually obtain. However, solving equation (2) often ends up with an overfitting of the problem because the number of the training datasets N is not infinite. Here we can try to avoid this issue by controlling the VC-dimension. The VC-dimension gives bounds on the expected risk $R_{\text{exp}}(f)$, which we cannot directly obtain due to the lack of information about the probabilistic distribution of the whole dataset, as a function

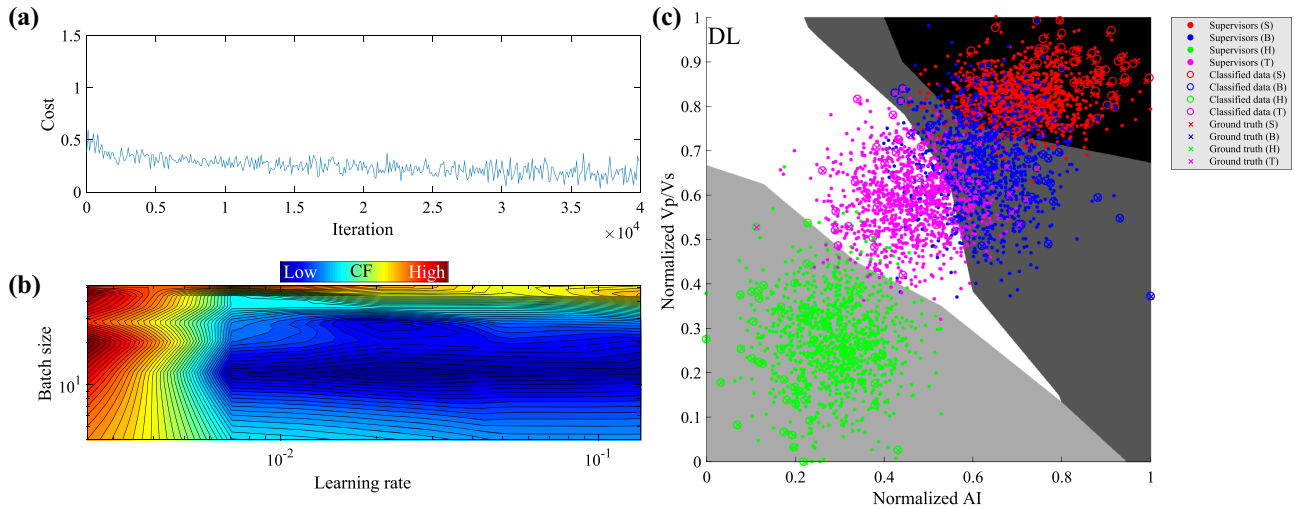


Figure 5 Synthetic elastic impedance cross-plot results from the DL approach where (a) is the cost function when the optimized hyperparameters are used, (b) is the lowest costs between the learning and modelled outputs when different hyperparameter values are used and (c) is the classification. The legend in (c) is the same as is described in Figure 4 with omission of the support vectors, which are not applicable to DL. Equivalent confusion matrices of training and test accuracies are shown in Figure 6(b) and (d).

of both $R_{emp}(f)$ and the number of data samples. Following Vapnik (1995), this can be written as:

$$R_{exp}(f) \leq R_{emp}(f) + \sqrt{\frac{b \left(\log \left(\frac{2N}{b} + 1 \right) - \log \left(\frac{\delta}{4} \right) \right)}{N}}, \quad (3)$$

where b is the VC-dimension of the model used to solve the problem, and $(1 - \delta)$ is a probability that equation (3) can satisfy. The second term of the right-hand side of equation (3) is called the VC-confidence. From equation (3), one can expect that the VC-confidence gets smaller (thus N increases and b decreases) when $R_{exp}(f)$ and $R_{emp}(f)$ converge. In other words, the best model occurs when the sum of $R_{emp}(f)$ and the VC-confidence is minimized and should therefore be automatically chosen by SRM.

Optimization problem

Support-vector-machine (SVM) theory, with respect to the optimal hyperplanes that classify the dataset non-linearly in the feature dimension via the so-called kernel-trick, has been presented by Li and Castagna (2004) and Li *et al.* (2015). Let us here directly start with the Lagrangian dual-problem regarding the maximal-margin of the classifiers as shown in the following:

$$\begin{aligned} \text{maximize : } L(\alpha_1, \alpha_2, \dots, \alpha_n) &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \\ &\times K(\mathbf{x}_i, \mathbf{x}_j), \end{aligned} \quad (4)$$

with the constraints:

$$\text{s.t. } \begin{cases} 0 \leq \alpha_i \leq c \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{cases}, \quad (5)$$

where α_i are the Lagrange multipliers (weights), $K(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel function that represents a dot (inner) product of feature vectors and c is the cost (penalization) parameter of the hinge-loss function that indicates how many misclassifications during the optimization run we can allow. c originally appeared in the primal problem (not shown here) of the dual-problem equation (4). Together with the Kuhn–Tucker conditions, the learning (identification) function can be obtained by solving equations (4) and (5) as follows:

$$f(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^n \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \right), \quad (6)$$

where \mathbf{x}_i are the support vectors for $\alpha_i \neq 0$ and b is the bias parameter that shifts the classifiers from the origin of the hyperplane solution. In Figure 2, a sketch of neural networks (NN) architecture for SVM is depicted. Since our actual problem is a multi-labelled classification, rather than the binary one, we used the one-against-one classification strategy.

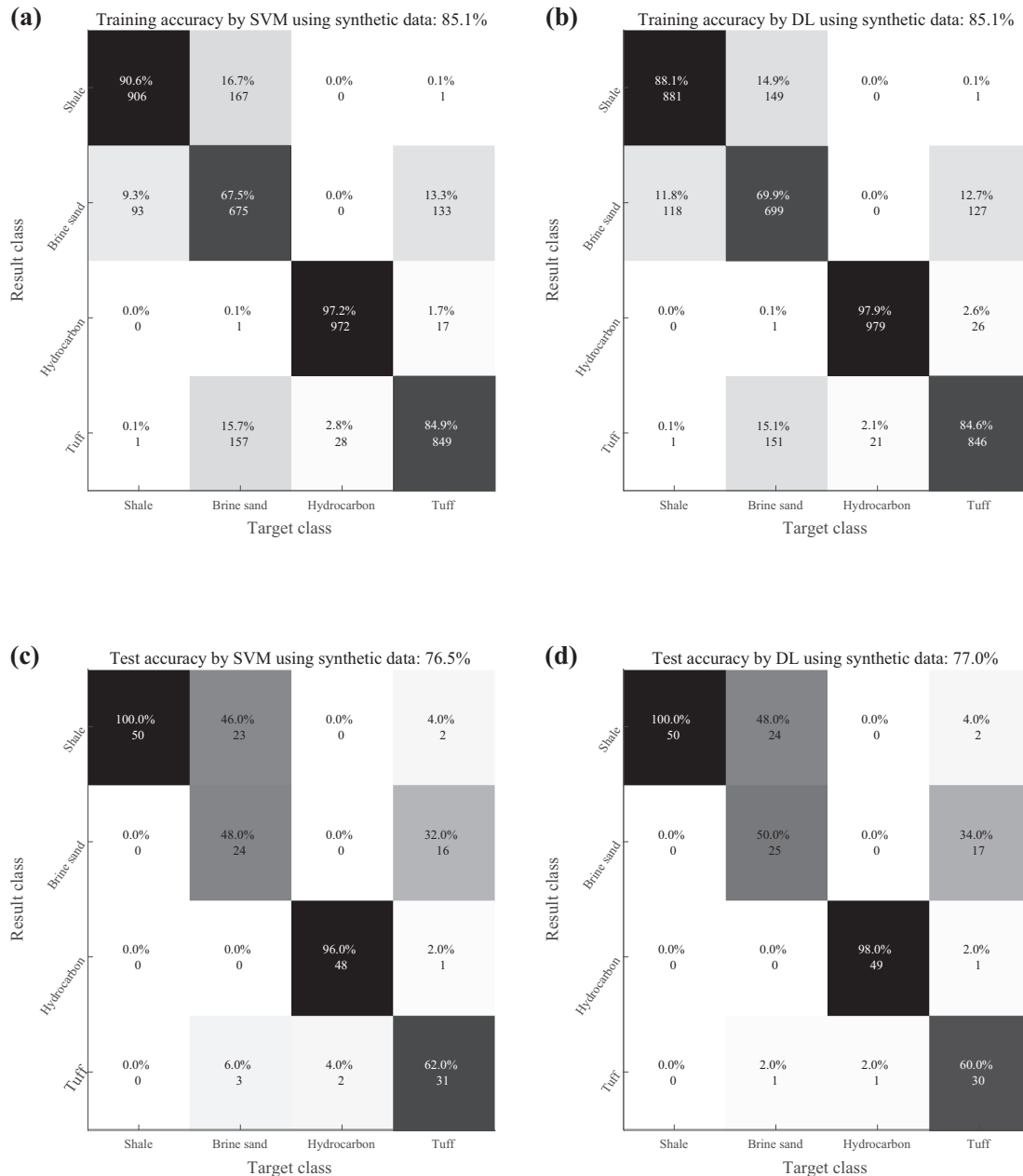


Figure 6 Confusion matrices from the synthetic data classification where (a) is the SVM and (b) is the DL training accuracies, whilst (c) is the SVM and (d) is the DL test accuracies. The percentile in each bin is the prediction accuracy between the target and result classes. The integer in each bin is the number of counts/samples used.

Kernel and hyperparameters

There are some parameters and functions that are pre-conceived in order to train the support vector machine (SVM). The key parameters are those associated with the kernel function and the hyperparameters. For the kernel function, following Schölkopf *et al.* (1997) who concluded that the radial

basis function (RBF or Gaussian) generally performed better than other kernels, RBF is also selected in this study. The RBF can be expressed as:

$$K(\mathbf{x}_i, \mathbf{x}) = \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}\|^2}{2\sigma^2}\right), \quad (7)$$

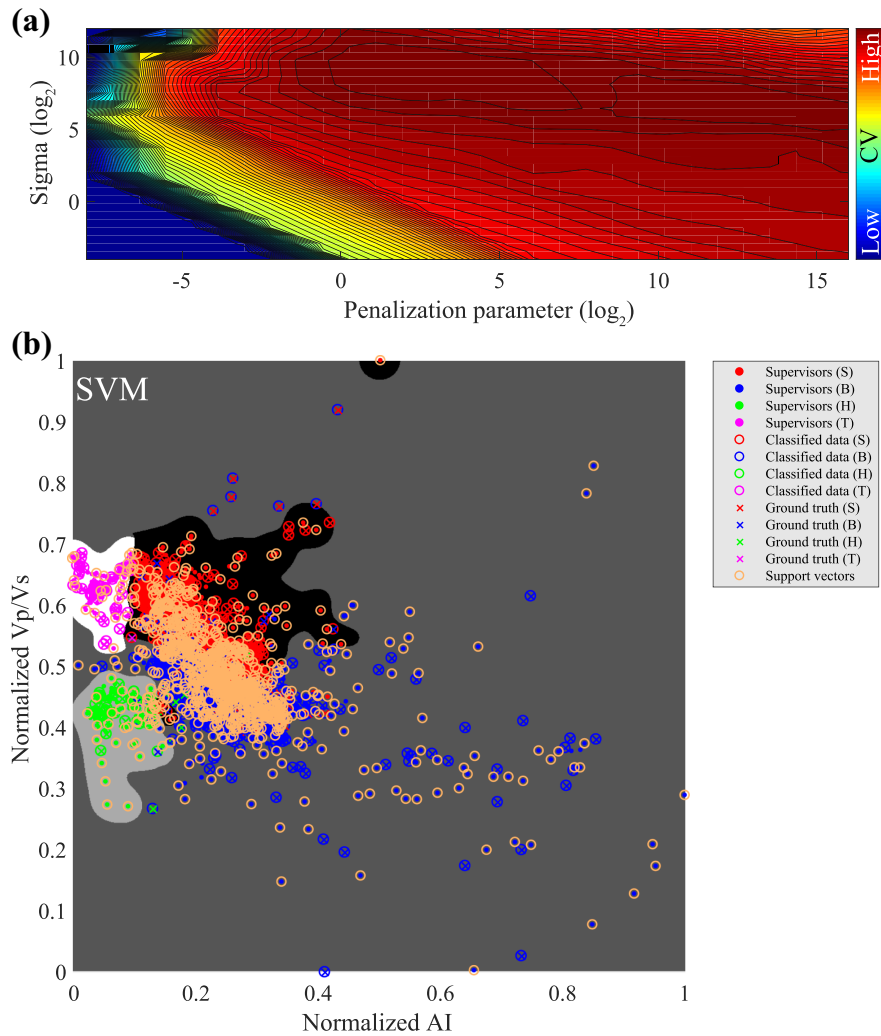


Figure 7 Elastic impedance cross-plot results from the SVM approach using the field data where (a) is the CV of the hyperparameters and (b) is the classification. The legend in (c) is the same as is described in Figure 4. Confusion matrices of the training and test accuracies are shown in Figure 10(a) and (d).

where σ is the kernel width (variance) that determines how far the influence should be reached. Intuitively speaking, we get smoother (linear) classifiers when smaller values of σ are used and vice versa. σ and the c parameters in equation (5) are often referred as the hyperparameters of SVM.

In order to determine specific values of these hyperparameters, an exhaustive grid search was carried out followed by a twofold cross-validation (Hsu, Chang and Lin 2003) over the training dataset. Since the number of major hyperparameters is two (σ and c), we were not encouraged to apply a random grid search (Bergstra and Bengio 2012) or Bayesian optimization (Mockus, Tiesis and Zilinskas 1978). After finding the optimized hyperparameters, via the exhaustive grid search, they were then applied to the rest of the data.

The deep learning approach

TensorFlow

For the deep learning (DL) approach in this study, we used TensorFlow developed and provided by Google (<https://www.tensorflow.org>). In TensorFlow, the architecture for DL is expressed as a graph that is basically a much simpler representation of a conventional neural networks (NN). Figure 3 shows the graph specifically designed for our problem. The architecture of Figure 3 can be translated into an algebraic equation as:

$$\mathbf{q} = \mathbf{a}(\mathbf{p} \cdot \mathbf{w} + \mathbf{b}), \tag{8}$$

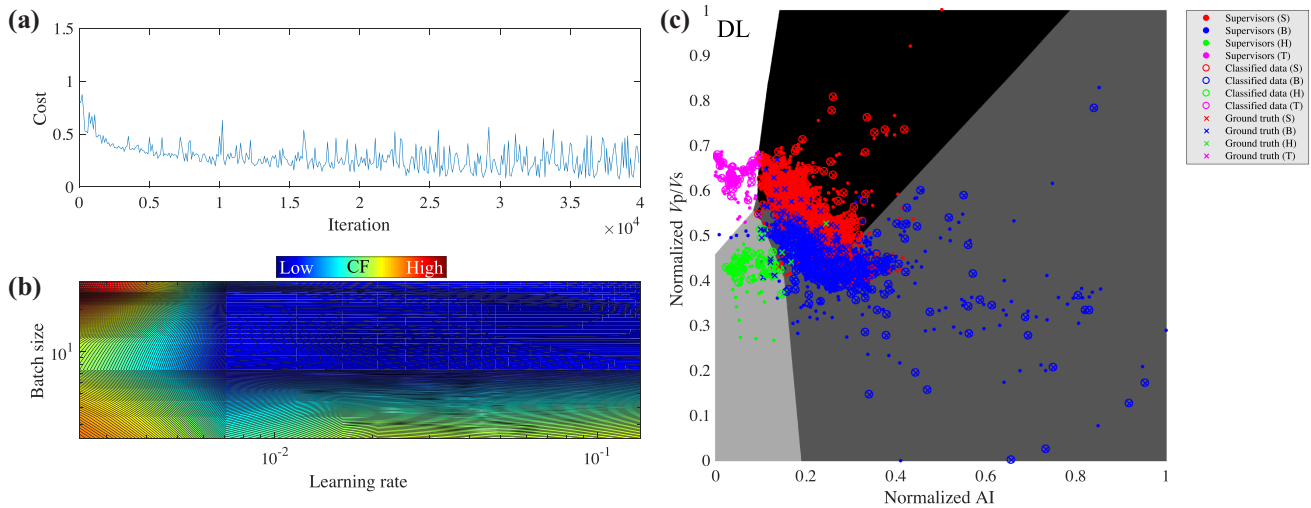


Figure 8 Elastic impedance cross-plot results from the DL approach using the field data where (a) is the cost function when the optimized hyperparameters are used, (b) is the lowest costs between the learning and modelled outputs when different hyperparameter values are used and (c) is the classification. The legend is the same as described in Figure 5. Equivalent confusion matrices of the training and test accuracies are shown in Figure 10(c) and (f).

where \mathbf{q} denotes the vector of outputs (the output layer in Figure 3), \mathbf{a} denotes the activation function (activator) that introduces non-linearity to the output, \mathbf{p} denotes the vector of inputs (the input layer in Figure 3), \mathbf{w} denotes the matrix of weights (w_i) whose values can be varied depending on the strength of the synapse (similar analogy to the biological brains neuron connectors) and \mathbf{b} denotes the vector of bias (b_i) that acts as a threshold of the neurons excitation. The number of neurons and hidden layers are parts of DL's hyperparameters. Note that, in the nomenclature of TensorFlow these are all different dimensions such that vectors and matrices are all termed tensors with different ranks. Following Heaton (2008), the number of neurons is set to be roughly two-thirds the size of the input layer plus the size of the output layer and the number of hidden layers are set to be two in order to handle any non-linear decision boundaries.

Optimization problem and hyperparameters

Similar to other learning algorithms, deep learning (DL) solves maximization/minimization problems using cost (objective, error and loss) functions. During the training process of the supervised classification in this study, the cost function C , between the expected labels and known labels, is back-propagated (Rumelhart, Hinton and Williams 1986) to the nearest and shallower layer to be updated. Our cost function is therefore a cross entropy between two probability functions;

however, this process does not happen without the hidden layers. The updating procedure is commonly carried out by minimizing the cost function using gradient-descent (Svetlana and Solov'ev 1997), which can be written as:

$$\text{minimize: } \nabla C^T \mathbf{d}, \quad (9)$$

with the constraint:

$$\text{s.t. } \|\mathbf{d}\| = 1, \quad (10)$$

where \mathbf{d} is the direction of the gradient-descent solver. Generally speaking, the gradient-descent (the first derivatives of the objective function) is easily trapped by local minima. An alternative approach, to avoid being trapped by local minima, is based on the calculation of Hessian matrices (the second derivatives of the objective function) that directly indicate a minimum, maximum or saddle point, thereby providing an escape route and more accurate solutions. However, their computational costs are much more expensive compared to the gradient descent approach alone. Therefore, we used a mini-batch stochastic gradient descent algorithm (Metel 2017) to increase computation speed. The algorithm is a combination of both batch gradient decent and stochastic gradient decent. Batch gradient decent tries to solve the cost function using the whole training dataset, which leads to local minima for non-convex surfaces of the cost function. The stochastic approach, on the other hand, provides improvement in computational cost and more chances to escape from local minima due to the method's ability to search out

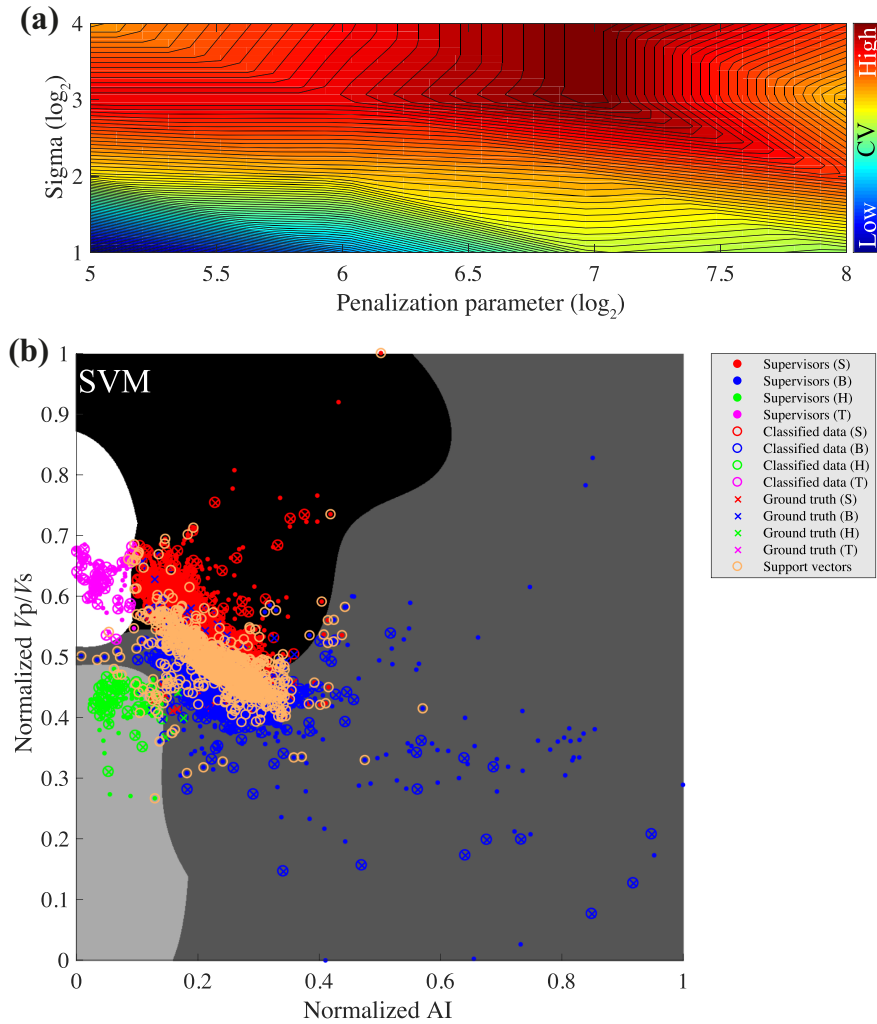


Figure 9 Elastic impedance cross-plot results from the SVM approach using the field data where (a) is the CV of the hyperparameters and (b) is the classification. The legend is the same as is described in Figure 4. The range of the hyperparameters are constrained when compared to Figure 7. Confusion matrices of the training and test accuracies are shown in Figure 10(b) and (c).

more solutions from multiple directions. In order to solve the cost function in equation (9), the weights and bias in equation (8) are updated as follows:

$$w_{i+1} = w_i - \eta \frac{\partial C}{\partial w_i} \tag{11}$$

and

$$b_{i+1} = b_i - \eta \frac{\partial C}{\partial b_i}, \tag{12}$$

where η is the learning rate (step size). However, one of the drawbacks of stochastic gradient descent is that the average trend of data redundancy is not measured because the algorithm is always based on one random data point. Therefore, mini-batch stochastic gradient descent combines the

advantages from both algorithms providing both an average trend of data redundancy and the ability to escape from local minima. The mini-batch size can therefore be regarded as one of key influencing hyperparameters for DL.

It is very difficult to optimize all of the hyperparameters since there are many (e.g. the depth of layers, the optimizer, the activator, the learning rate, the iteration number, the mini-batch size, the regularization and so on) as well as accommodating the trade-off effects among them, which can be an added complication (Li *et al.* 2015). This issue explains why this optimization is one of the most pertinent research topics in DL. However, there seems to be consensus that the learning rate is one of the most critical hyperparameters (Bergstra and Bengio 2012).

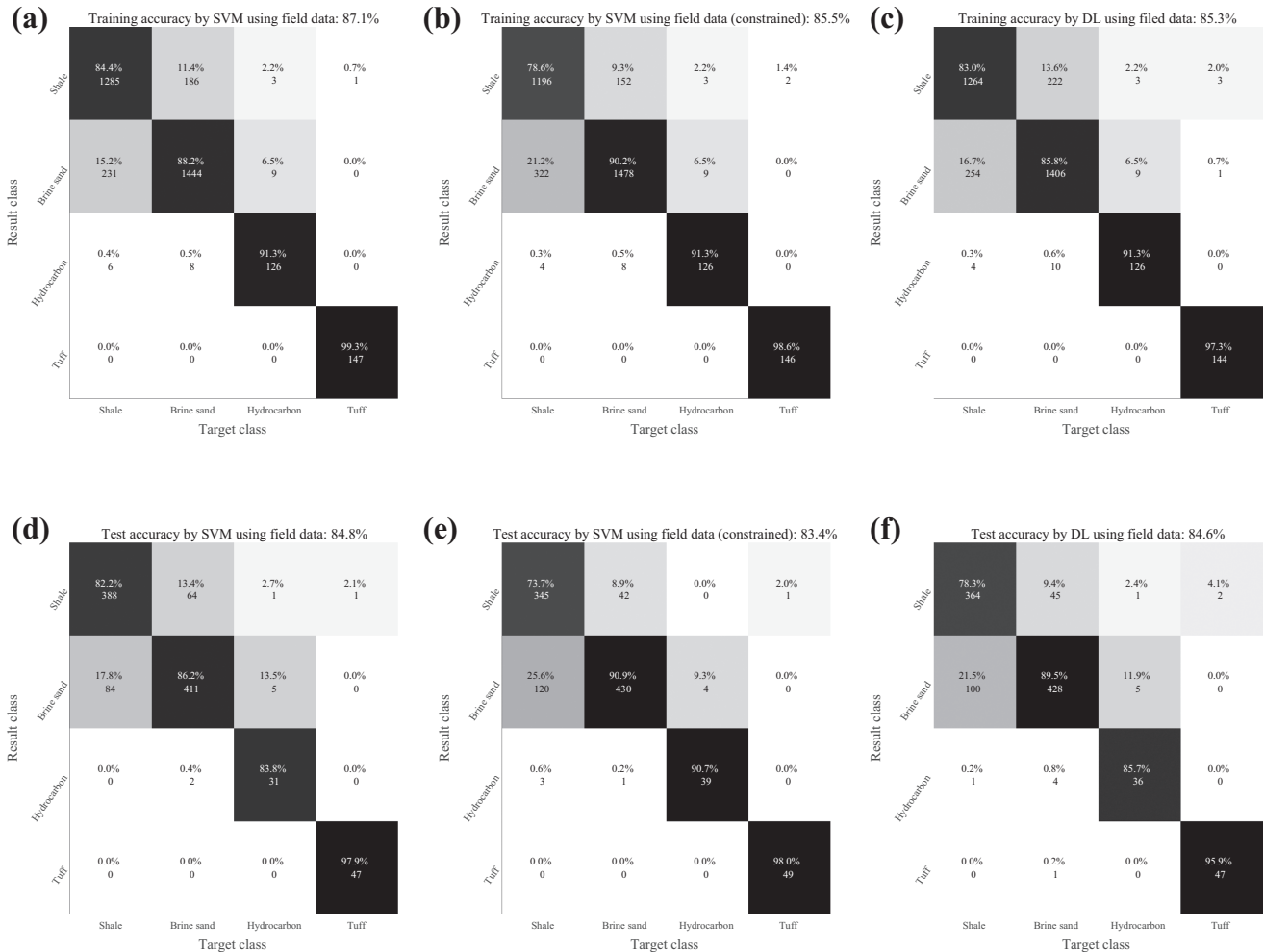


Figure 10 Confusion matrices from the field data classification, where (a) is the SVM training accuracy as in Figure 7, (b) is the SVM training accuracy with the constrained hyperparameters as in Figure 9, (c) is the DL training accuracy as in Figure 8, (d) is the SVM test accuracy as in Figure 7, (e) is the SVM test accuracy with the constrained hyperparameters as in Figure 9 and (f) is the DL test accuracy as in Figure 8.

The hyperparameters used in the DL workflow were determined as follows. The activators for the hidden layers and the output layers are the rectified linear unit (ReLU, also called the ramp-function) and the softmax. The ReLU in the hidden layers is known to avoid the gradient vanishing problem, whereas the softmax in the output layer provides the probability of each output making the DL approach probabilistic in application. There is no substantial difference between either the sigmoid or the softmax in the output layer if the classification is a binary problem. The iteration number is set as 40,000. The choice of the solver and the depth of the neural networks (NN) are already described above. The remaining two main hyperparameters (the learning rate and the mini-batch size) are optimized by the exhaustive grid search, just as we applied to the support vector machine (SVM),

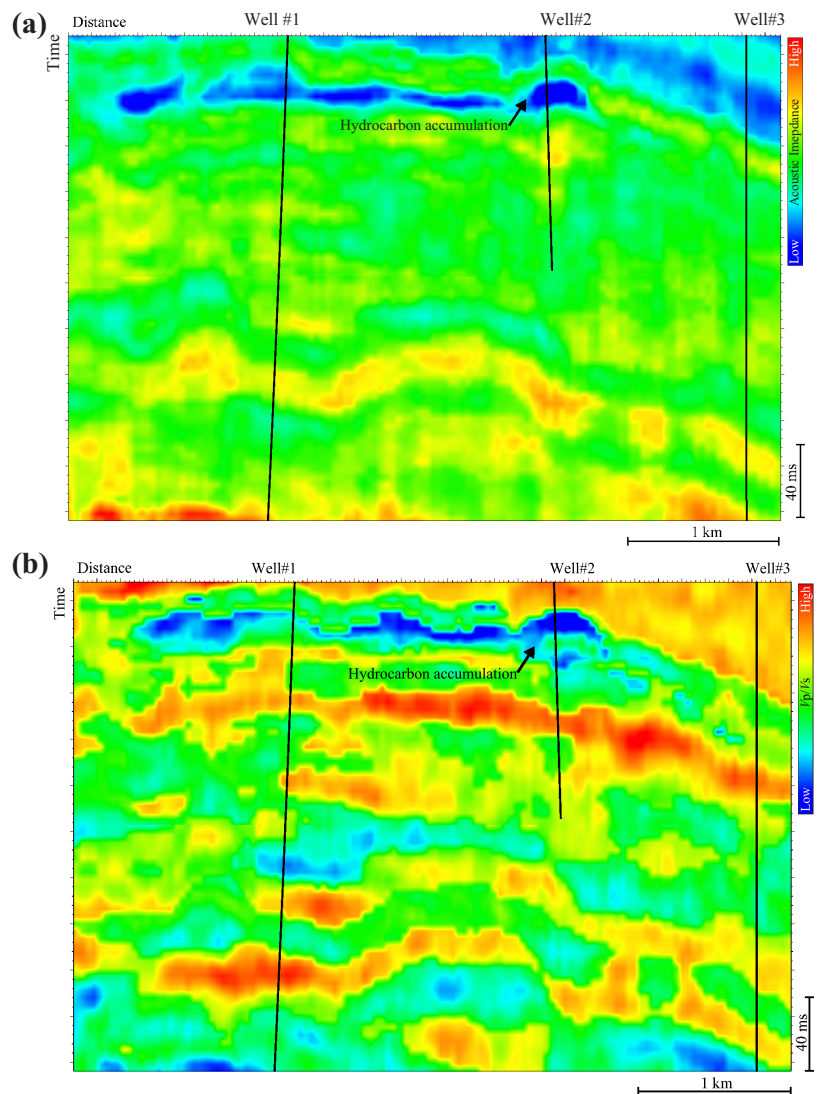
thereby providing a fair comparison. Note that there are many reports about the optimization of the hyperparameters. For instance, while Bergstra and Bengio (2012) suggested that the random grid search is a good choice, Snoek *et al.* (2015) argued the Bayesian optimization is better. Such comparisons are beyond the scope of this study.

RESULTS

Synthetic test results

The field dataset we intend to classify has four facies (shale, brine sand, hydrocarbon-bearing sand and tuff) that are determined by elastic cross-plot products principally using AI and V_P/V_S . Before performing the actual application of support vector machine (SVM) with LIBSVM

Figure 11 Cross-sections (location shown in Figure 1(b)) depicting the output seismically derived elastic impedances, where (a) AI and (b) V_p/V_s . These data were derived from a joint-impedance facies inversion and used as the input to the various classification methods as shown in Figure 12. An anomalous response, in terms of decreased AI and V_p/V_s , is visible at the top of the section and corresponds to a hydrocarbon accumulation encountered within Wells #1 and #2.



(<http://www.csie.ntu.edu.tw/~cjlin/libsvm>), we tested characterization feasibility with a synthetic dataset in order to check whether the elastic parameters provided sufficient input to identify the four facies. The synthetic dataset has been generated by a randomizer to realize the four-labelled facies. The number of training data (supervisor) and test (classified) data per label is 1000 and 50, respectively. Figure 4(a) shows the cross-validation value obtained between σ and c using the synthetic dataset. The cross-validation was done by the exhaustive grid search with the global-maximum value being found successfully (Figure 4(a)). The test accuracy of the classification was found to be 76.5%. With these given hyperparameters, SVM was performed as shown in Figure 4(b). The classifiers in Figure 4(b) appeared to not be too hard

(linear wise) or be too soft (non-linear wise), which gave us some confidence that the SVM worked in terms of chosen hyperparameters.

Using an identical dataset, as input to the SVM modelling in Figure 4, we employed the same synthetic test for the deep learning (DL) approach (Figure 5). Figure 5(a) shows the cost function per iterations when the optimized hyperparameters from the grid search are used. In Figure 5(b), we show the cost value associated with the learning rate and the mini-batch size. The results indicate that the global minima was found (Figure 5(b)). Finally, the classifier results are shown in Figure 5(c). The test accuracy of the classification is found to be 77.0%, which is nearly same as to what was achieved with the SVM approach. Both the results in Figures 4 (SVM) and

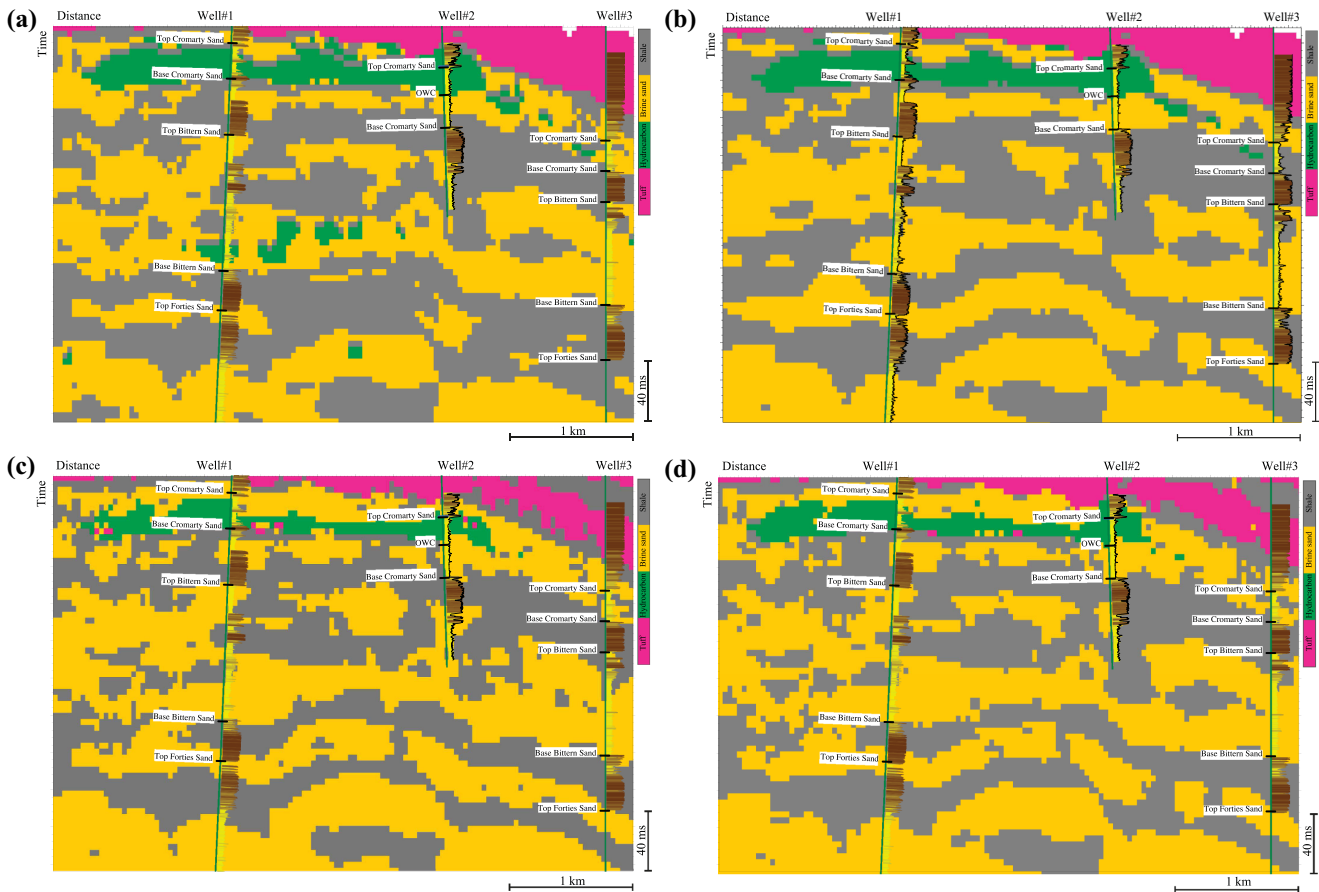


Figure 12 Cross-sections (location shown in Figure 1(b)) depicting the seismic upscaling output from the four main facies classification methods where (a) is the linear, (b) is the Bayes, (c) is the SVM and (d) is the DL approach. The Vshale logs are also plotted for each of the wells (yellow for sand and brown for shale). Note the Vshale logs are shown without being filtered back to the seismic's vertical resolution in order to illustrate the marked difference in vertical resolution between the seismic and well data.

5 (DL) appear to be visually similar. The computation time of the DL approach is, however, 1.6 times faster than that of the SVM approach when all the hyperparameters are fixed in our environment.

To objectively compare the performance between the SVM and DL approaches, when the above synthetic data are used, we plot confusion matrices for the training and test accuracies in Figure 6. The percentiles for each bin are also shown in Figure 6 and provide the prediction accuracy per target class when the sample numbers in the same bin are used.

Field data results

In Figures 7 and 8, we show the classification results of the elastic impedance cross-plot products applied to support vector machine (SVM) and deep learning (DL) using the Avalon discovery well (Well #1 in Figure 1). Wells #2 and #3 were not

used for the training process, but instead acted as blind tests, in order to objectively test the seismic upscaling of the SVM and DL facies classification outputs. The ratio of the training to test data was fixed at 0.7. Although the hyperparameters of the SVM approach seem to be sufficiently optimized via cross-validation (Figure 7(a)), the classifications in Figure 7(b) appeared to be over-fitted (exhibit strict decision boundaries) with a classification accuracy of 84.8%. As pointed out by Kuzma (2003) previously, higher penalization values can be numerically unstable and for this reason we therefore limited their range in Figure 9. This range limitation yielded softer decision boundaries (Figure 9(b)) with the trade-off that the hyperparameters were ultimately marginally less optimized (Figure 9(a)) compared to before (Figure 7(a)). The classification accuracy was still however 83.4%. On the other hand, the DL result (Figure 8(b)) appeared to provide simpler decision boundaries compared to the SVM result (Figures 7(b)

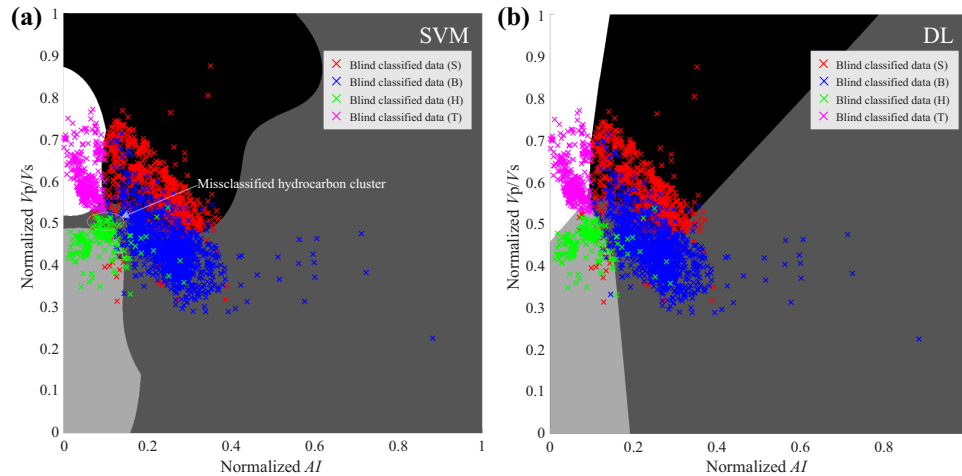


Figure 13 Blind test of elastic impedance cross-plot products from Wells #2 and #3, whose locations are given in Figure 1(b). The classification is based on Well #1 by (a) the SVM and (b) the DL approaches, which are identical to the results already shown in Figures 9 and 10. The arrow marks the area, corresponding to Well #2, where hydrocarbons should have been classified instead. This SVM misclassification resulted in a decreased hydrocarbon column around Well #2, Figure 12(c), compared to what was actually encountered.

and 9(b)). We observed that the overall trend of the DL cost function seems to decrease with increasing iterations (Figure 8(a)) but also becomes less efficient as it fine tunes itself to the optimum solution. In Figure 8(b), the combination between learning rate and batch size looks to be approximately optimized giving a classification accuracy of 84.6% (Figure 8(c)).

As we made confusion matrices with the synthetic data (Figure 6), we also show equivalent matrices in Figure 10 when SVM and DL are applied to the field data. However, based on the matrices, we do not observe any major differences between the approaches.

Seismic upscaling

For the purpose of demonstrating practical application and validation of the obtained results against the blind test wells (Wells #2 and #3), we apply the SVM and deep learning (DL) trained classifiers, as shown in Figures 9(b) and 8(c), to actual seismic data. Figure 11 shows the amplitude versus offset (AVO) inversion derived AI and V_p/V_s used as input to allow the upscaling away from Well #1. It is noted that areas with low AI and low V_p/V_s (labelled in Figure 11) correspond to proven hydrocarbon bearing facies. In order to determine any substantial differences between more conventional facies classification methods and the SVM or DL approaches, we compare four different classifications: linear projection, Bayes, SVM and the DL solution, as shown in Figure 12. We normalize the seismic input in the same fashion to the elastic

impedances from the well data to ensure a consistent scale. The linear-regression line ($\text{Projection} = (uAI + v) - V_p/V_s$ (where u denotes the gradient and v denotes the intercept of the regression line)) is calculated by gradient decent. The Bayes result is calculated by the facies-based seismic inversion (Zabihi Naeini and Exley 2017) and uses probability distribution functions to determine facies in elastic impedance cross-plot space. Note that the result of the Bayes approach (Figure 12(b)) is representatively taken from where we have strongest confidence from the corresponding probability density functions (the maximum a posteriori). The main formation tops around the reservoir and Vshale (the volume of shale) at the three wells are also plotted in Figure 12.

DISCUSSION

One of the most obvious observations from Figure 12 is that the four different methods generally exhibit similar results. In particular, the results of the linear projection (Figure 12(a)) and the deep learning (DL) approach (Figure 12(d)) appear to be most similar compared to other comparisons. However, it can be noted that the linear approach wrongly identified hydrocarbons within the Base Bittern Sand at Well #1 (Figure 12(a)); conversely this error does not present itself in the DL result (Figure 12(d)). The Bayes result (Figure 12(b)) has better signal-to-noise ratio compared to the other methods with good lateral continuity of potential sedimentary layers but had the poorest correlation with the wells particularly the Cromarty Sand interval in Well #3. The support vector

machine (SVM) result (Figure 12(c)) shows better correlation within the deeper section but a reduced hydrocarbon column compared to what was actually encountered in the wells, which was due to the harder decision boundaries in Figure 9(b) (SVM) compared to Figure 8(c) (DL).

To further check the performance of the SVM and DL approaches, we applied AI and V_p/V_s derived from blind Wells #2 and #3 to the predicted classification in Figure 13. Although the test accuracy in Figure 10 indicated similar performance, the classification of the brine sand in Figure 13(a) seems to be overestimated, around the normalized AI of 0.1 and V_p/V_s of 0.5, where hydrocarbons should have been classified (labelled in Figure 13). This would explain why the SVM result in Figure 12(c) shows a decreased hydrocarbon column compared to what was actually encountered in the wells or the DL result in Figure 12(d). In general, the SVM result suffers from an overly strict classification, with marginally more constrained decision boundaries, compared to the DL result. However, both the SVM and DL results could converge with additional training using a wider range of input training data, more representative of general rock physics trends.

It is natural to expect that better results can be obtained when more training wells, for instance Wells #2 and #3, are used. However, such improvements are not necessarily guaranteed for the linear approach because it cannot handle non-linear (or complex) decision boundaries. On the other hand, while the Bayes approach is more suited to more complex decision boundaries, that is where different facies overlap in elastic cross-plot space, it requires good prior information to determine appropriate probability distribution functions. Whilst the SVM and DL approaches do not require this prior knowledge, they do not take into account any rock physics trends that may be known outside of the immediate input data range provided by the learning data. This is in contrast to both the linear and Bayes based methods that are orientated to capture general compaction/porosity/depth trends.

One potential way to increase DL's performance further is to use data augmentation, especially when the available data are limited. In the case of convolutional neural networks (CNN), augmenting image samples by rotating, stretching, shrinking, flipping, adding noise and so on is common (Okafor *et al.* 2017). These processes can be applied either in the original or feature domain. However, since DL automatically and implicitly extracts relevant features from the input data, one might, for instance, try to use V_{shale} , density, V_p , V_s or other rock property trends such as the Castagna equation (Castagna, Batzle and Eastwood 1985). These inputs could be directly

used in the first layer of DL, or pre-training, to extract features via an auto-encoder (Baldi and Hornik 1989).

Although all of the details regarding the well correlations are not disclosed here, the overall correlation of sand, shale, tuff and oil sand facies compared with the seismically derived classifications is qualitatively reasonable, as shown in Figure 12 despite differences in vertical resolution. It is, however, difficult to decisively choose which one of the classification methods might be superior because each method is inherently subject to different weaknesses and ultimately classification errors. For example, as briefly described above, the linear regression can be too simplistic when there is significant overlap of different facies types in elastic cross-plot space. Whilst the Bayes-based method can treat such overlapping scenarios better it does, however, require good prior information that may not be known. Also, as discussed previously with the SVM and DL approaches, appropriate optimization of the hyperparameters is essential and also subject to error. Nonetheless, we find great potential in the application of the DL method with respect to elastic impedance cross-plot classification as given sufficient training one can expect to achieve a high level of automation within inversion workflows, whilst also reducing reliance on human interactions and rock physics assumptions.

CONCLUSIONS

This paper presents support vector machine (SVM) and deep learning (DL) facies classification examples using well-derived elastic impedances from the UK, North Sea. Additionally, the SVM and DL methods were also upscaled and applied to equivalent elastic outputs of an amplitude versus offset (AVO) inversion applied to a seismic field dataset covering the well data. Although the SVM and DL approaches provided similar results with a simple synthetic input, there were obvious differences when upscaled to the seismic data. Such differences seem to be connected to the variation in optimized hyperparameters between the SVM and the DL approaches. Even though the SVM approach provided similar training accuracy, compared to DL, the DL approach showed visually more realistic results and better correlation with the well data. The similarities of the 'automated' SVM and DL results when compared to established 'manual' classification methods such as linear projection or Bayes-based classification are encouraging and suggest that machine learning approaches such as DL have the future potential to guide us towards automated quantitative interpretation, whilst also mitigating subjective human interactions.

ACKNOWLEDGEMENTS

The authors would like to thank Summit Exploration and Production, Ping Petroleum, Petro Summit E&P Corporation and Sumitomo Corporation for their permission to publish. We also thank S. Masaya at TU Delft for his theoretical discussion and Ikon Science for their contribution to this work. We appreciate Behzad Alaei and one anonymous reviewer for their constructive comments that improved the quality of this manuscript. The seismic input to this study was CGG's, TomoML, multi-client data. For SVM and DL applications, LIBSVM and TensorFlow were used in this study, respectively. The maps and sections were drawn with ArcGIS, GMT (Wessel and Smith 1991) and IHS's Kingdom.

REFERENCES

- Baldi P. and Hornik K. 1989. Neural networks and principal component analysis: learning from examples without local minima. *Neural Networks* 2, 53–58.
- Bergstra J. and Bengio Y. 2012. Random search for hyperparameter optimization. *Journal of Machine Learning Research* 13, 281–305.
- Castagna J.P., Batzle M.L. and Eastwood R.L. 1985. Relationships between compressional-wave and shear-wave velocities in clastic silicate rocks. *Geophysics* 50, 571–581.
- Heaton J. 2008. *Introduction to Neural Networks for Java*. Heaton Research.
- Hinton G.E., Osindero S. and Teh Y.-W. 2006. A fast learning algorithm for deep belief nets. *Neural Computation* 18, 1527–1554.
- Hinton G.E. and Salakhutdinov R.R. 2006. Reducing the dimensionality of data with neural networks. *Science* 313, 504–507.
- Hsu C.-W., Chang C.-C. and Lin C.-J. 2003. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University.
- Kemper M. and Gunning J. 2014. Joint impedance and facies inversion: seismic inversion redefined. *First Break*, 32, 89–95.
- Kuzma H.A. 2003. A support vector machine for AVO interpretation. 73rd SEG meeting, Dallas, USA, Expanded Abstracts, 181–184.
- Li G., You J. and Liu X. 2015. Support vector machine (SVM) based prestack AVO inversion and its applications. *Journal of Applied Geophysics* 120, 60–68.
- Li J. and Castagna J. 2004. Support vector machine (SVM) pattern recognition to AVO classification. *Geophysical Research Letters* 31, L02609.
- Li Z., Jin L., Yang C. and Zhong Z. 2015. Hyperparameter search for deep convolutional neural network using effect factors. IEEE China Summit and International Conference on Signal and Information Processing, Chengdu, China, pp. 782–786. IEEE.
- Metel R.M. 2017. Mini-batch stochastic gradient descent with dynamic sample size. arXiv:1708.00555.
- Mockus J., Tiesis V. and Zilinskas A. 1978. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization* 2, 117–129.
- Mudge D.C. 2014. Regional controls on Lower Tertiary sandstone distribution in the North Sea and NE Atlantic margin basins. *Geological Society, London, Special Publications*, 403, 17–42.
- Okafor E., Smit R., Schomaker L. and Wiering M. 2017. Operational data augmentation in classifying single aerial images of animals. *IEEE International Conference on INnovations in Intelligent Systems and Applications (INISTA)*, pp. 354–360. IEEE.
- Rosenblatt F. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review* 65, 65–386.
- Rumelhart D.E., Hinton G.E. and Williams R.J. 1986. Learning representations by back-propagating errors. *Nature* 323, 533–536.
- Schölkopf B., Mika S., Burges C.J.C., Knirsch P., Müller K.-R., Ratsch G. and Smola A.J. 1999. Input space versus feature space in kernel-based methods. *IEEE Transactions on Neural Networks* 10, 1000–1017.
- Schölkopf B., Sung K.-K., Burges C.J.C., Girosi F., Niyogi P., Poggio T. and Vapnik V. 1997. Comparing support vector machines with Gaussian kernels to radial basis function classifiers. *IEEE Transactions on Signal Processing* 45, 2758–2765.
- Smola A.J. and Schölkopf B. 2004. A tutorial on support vector regression. *Statistics and Computing* 14, 199–222.
- Snoek J., Rippel O., Swersky K., Kiros R., Satish N., Sundaram N., Patwary M.M.A., Prabhat and Adams R.P. 2015. Scalable Bayesian optimization using deep neural networks. *Proceedings of the 32nd International Conference on Machine Learning*, pp. 2171–2180.
- Svetlana S.P. and Solov'ev A.D. 1997. The origin of the method of steepest descent. *Historia Mathematica* 24, 361–375.
- Vapnik V. 1995. *The Nature of Statistical Learning Theory*. Springer.
- Vapnik V. and Chervonenkis A. 1974. *Theory of Pattern Recognition* (in Russian). Nauka.
- Vapnik V. and Lerner A. 1963. Pattern recognition using generalized portrait method. *Automation and Remote Control* 24, 774–780.
- Wessel P. and Smith W.H.F. 1991. Free software helps map and display data. *EOS Transactions AGU* 72, 441–448.
- Zabihi Naeini E. and Exley R. 2017. Quantitative interpretation using facies-based seismic inversion. *Interpretation* 5, SL1–SL8.