Preliminary Report
Variational Autoencoders (VAEs)
for Well Log Facies Prediction

Yujie Zhou

Department of Earth Science & Engineering, Imperial College London, United Kingdom

Supervisors: Ehsan Naeini (Ikon), Olivier Dubrule & Lukas Mosser (Imperial College London)

1. Introduction

Well log present a concise, detailed record of formation parameters versus depth [1]. The sedimentary facies and lithologies can be roughly inferred through interpretation of well logging data which reflects the properties of underground geological features. The traditional way of well log interpretations is manual identification based on the shape of well logging curves. This approach may be time-consuming and laborious, even implies instability in some cases because manual facies classification must often be combined with seismic lithostratigraphic interpretation of seismic data as well as the interpretation of well core data. In other words, it may lead to inaccurate results without the support of other data information. Thus, traditional well log interpretation may be inefficient for large data sets of hundreds of wells.

In this project, we attempt to explore some other approaches for well log classification with the help of machine learning. We will focus on the implementation and analysis of Variational Autoencoders (VAEs) for improved training facies prediction. New well-logs created by VAEs will be added to the original training set to enhance the Neural Networks.

The project will be done via an internship at Ikon Science based in London. Ikon Science is a leading software and services company in the field of reservoir characterisation, pore pressure prediction and geomechanics analysis. All the algorithm will be implemented in Python file, integrated with RokDoc software by an XML Document, which produces the user interface. RokDoc is Ikon's flagship software used by many operators around the world [2]. As part of this project, Deep QI which is RokDoc's module for deep learning has been used for preliminary study (see case study on the Forties field below) and as a reference to the what will be developed during this internship (see Proposed Approach section).

In this project, the well log dataset used for training and performance evaluation comes from 29 wells from the Forties Field, located 177 km offshore Aberdeen in the North Sea (Fig 1) within the UK production block 21/10 [3].

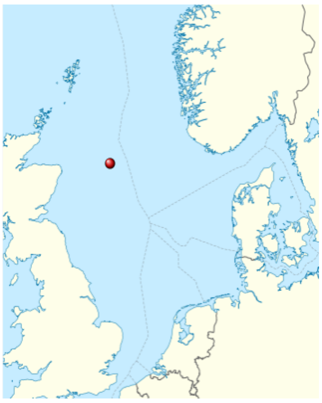
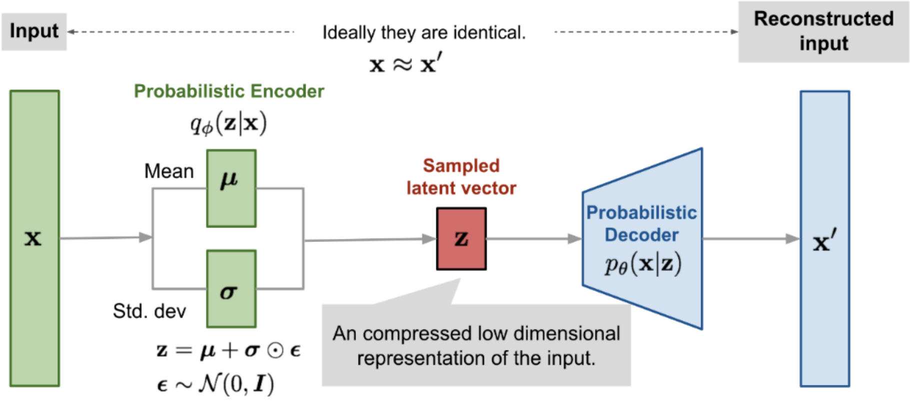


Fig. 1. Location of Forties Oil Field [4]

2. Literature Review

Machine-learning-based data interpretation methods for underground characterization has been in applied in practice [5], where well logs get interpreted through neural networks (NNs) [6]. One of the examples is classifying a formation into different facies and then use genetic neural network to estimate the porosity and permeability by P. M. Wong [7]. H.-C. Chang implement fuzzy memory neural network to predict lithofacies from well logs in Ordovician rock [8]. An inception convolutional network also has been employed for supervised facies classification by V. Tschannen [9]. In addition to some classic neural networks for supervised learning, Generative Adversarial Neural Networks (GAN) [10] and Variational Autoencoder (VAEs) [6] also have been taken into geological application.

The very first idea of Variational Autoencoders (VAEs) came out in 2013 [11] and have been applied in many areas such as medical image processing. Variational Autoencoders (VAEs) can be described as a special kind of neural network, which produces new samples that are statistically compatible with the probability density function of data from the training set.



Input
Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Reconstructed input
Probabilistic Encoder
$q_\phi(\mathbf{z}|\mathbf{x})$
Mean
$\boldsymbol{\mu}$
Std. dev
$\boldsymbol{\sigma}$
$\mathbf{x}$
$\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \epsilon$
$\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$
Sampled latent vector
$\mathbf{z}$
Probabilistic Decoder
$p_\theta(\mathbf{x}|\mathbf{z})$
An compressed low dimensional representation of the input.
$\mathbf{x}'$

Fig.2. Illustration of variational autoencoder model with the multivariate Gaussian assumption. [12]

The VAE architecture (Fig.2) encodes training data input X into the latent vector z, which is constrained to follow a Gaussian distribution [12, 13]. VAEs are not totally the same thing as autoencoder model and the difference is that the input data is mapped based on probabilistic latent coordinates instead of deterministic ones. to constrain latent vectors into a Gaussian distribution [11]. VAEs are not totally the same thing as autoencoder model and the difference is that the input data is mapped based on probabilistic latent coordinates instead of deterministic ones. Let us write the data distribution is as p_θ (controlled by parameter vector θ). To compute samples from p_θ, a neural network calculates the parameters of the conditional distribution $q_\phi(z|x)$ and of the conditional distribution $p_\theta(z|x)$ [12].

One example of VAEs employed for subsurface characterization is that H. Li constructed a variational Autoencoder-Based Neural Network [6] to extract and reproduce the NMR T2 distributions. In his paper, he constructed a VAEs model with three hidden layers which have 16, 2 and 16 neurons, respectively. His trained VAEs performs well with overall R^2 of 0.75. It was mentioned that the noise in well logs, uncertainty in inversion-derived logs, and insufficient training dataset can lead to partial low R^2, whereas VAEs can enlarge the training data. Therefore, there is reason to believe appropriate implementation of VAEs has great potential to enhance the subsurface characterization using limited well logs.

3. Work Accomplished in the First Two Weeks

As an initialization for algorithmic part of the project to get familiar with the software environment, a case study was carried out on the performance of different machine learning methods on well-log from Forties field. Several methods were implemented through a menu of the RokDoc user interface, which is associated with python file (*.py) consisting of existing background algorithms. The machine learning methods available in RokDoc are:

- Supervised Learning – deep neural network (DNN)
- Unsupervised Learning
  - Dimensionality Reduction (DR): PCA, ICA, KPCA, Factor Analysis, Sparse Random Projection (SRP)
  - Clustering (KMeans, GMM, Mean Shift, DBSCAN and Hierarchical)

These methods can be employed through the menu module of Deep QI or Unsupervised corresponding to DNN and Unsupervised Learning. Deep QI or Unsupervised menu modules both have Training and Application submenus for training and testing. Training submenus consists of input and target logs selection (single or multi wells), user defined experiment settings and saving the trained model. For supervised training, we can set up parameters such as the proportion of validation data and enter the structure of deep learning network. For unsupervised learning, we can pick one preferred clustering method with or without one dimensionality reduction methods. Then in Application submenu, the saved model can be applied to wells that were not in the training set to get a test accuracy and visualise the predicted facies in a new well log panel. It should be noted that the facies type is indeed an integer, corresponding to a specific type of lithofacies (table.1). Therefore for unsupervised learning, there is a problem is how to match the predicted output integer with the right type of

Table. 1. Four facies type and their corresponding integer value in the well log data

Legend	Facies Type	Integer
🟨	Brine Sand	1
🟩	Oil Sand	2
▤	Shale	3
🟪	Soft Shale	4

facies. For example, we can get a reliable clustering result that the distribution of predicted facies looks similar to the interpreted facies, but the predicted facies (clusters) are corresponding with the wrong integer. To solve this problem and we can enter a mapping array to map the output integers into the correct ones in the Unsupervised-Application submenu. RokDoc software will calculate and display an appropriate mapping array for users when doing testing, and users can enter this one into the software to get a test accuracy. Below shows the predicted facies before and after mapping (Fig.3).

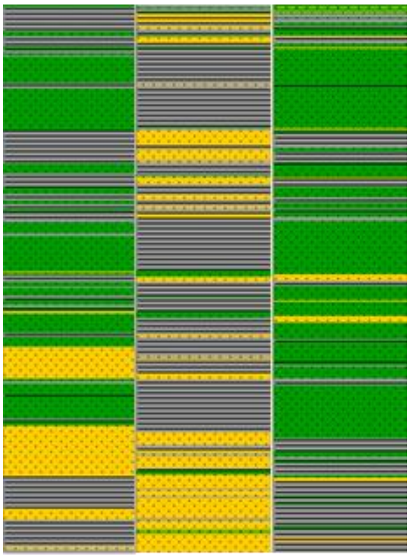


Fig. 3. Comparison of predicted facies by clustering before and after mapping. From left to right, three facies columns represent interpreted facies, facies before mapping and facies after mapping, respectively.

To get a better prediction of lithofacies, I determine to pick six well-logs as the input, which is Vp, Vs, Density, GR, Porosity and Resistivity. Of all wells in Forties Fields, twenty-four wells are available for these six kinds of well log data, in which twenty-two wells were selected for training and two blind wells for application: Well 21/9-6A, Well 22/6A-2 (Fig.4).

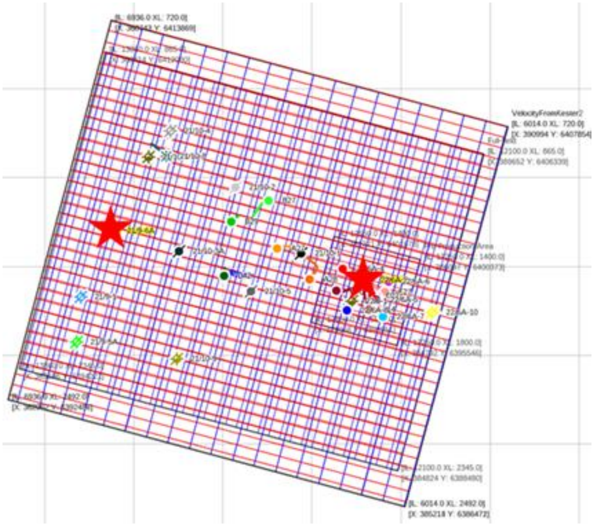
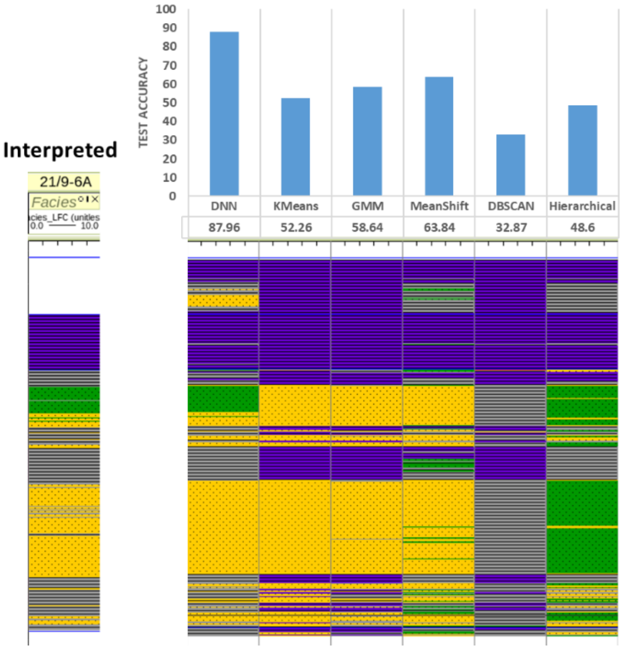


Fig. 4. Locations of Wells in Forties Field. Two blind wells for application are highlighted with stars.

In the supervised training, Stratified sampling was used instead of random sampling for splitting the original training dataset into training and validation set, to ensure these two datasets have approximately the same percentage of samples of each class as the complete set. The validation data was picked from training data by a proportion of 80%. A deep neural network was constructed, with 8 hidden layers of size 256, 128, 128, 64, 128, 128, 128 and 64 respectively. The network was trained for 100 epochs with learning rate as 0.01 and batch size as 50. The test accuracy was 87.96% for Well 21/9-6A and 89.56% for Well 22/6A-2 which are good.

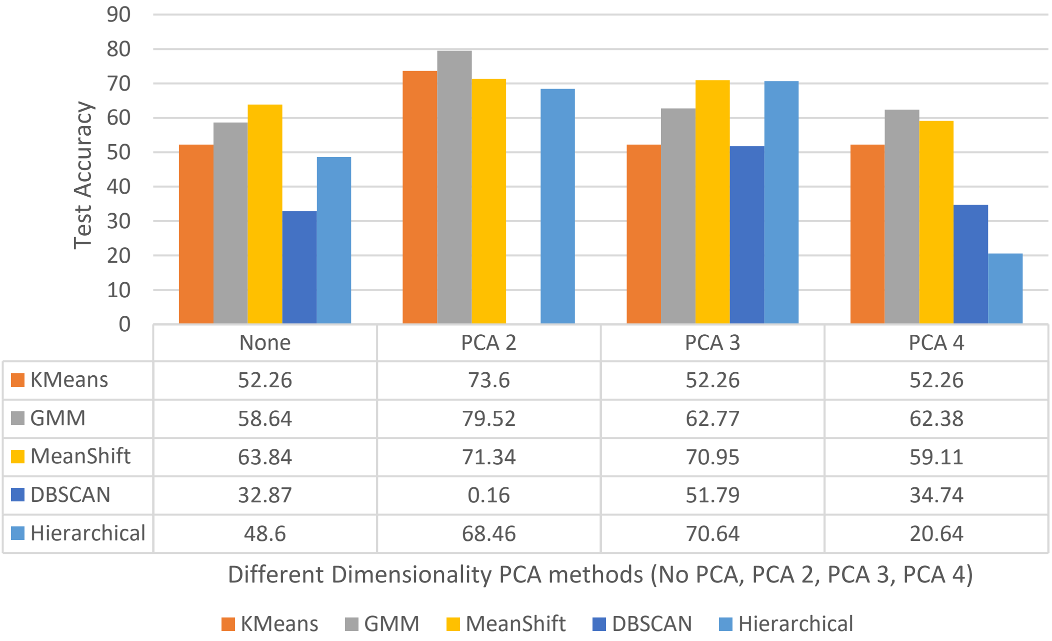
In unsupervised learning (UL), different clustering methods were carried out with and without dimensionality reduction (DR). PCA was used to reduce the dimensionality of input well log data from six to two, three and four respectively (PCA 2, PCA 3, PCA 4). That is, there are two sets of variates: different clustering methods; not using any dimensionality reduction methods and using PCA to get different dimensionality. Firstly, different clustering methods with no DR. If summarizing the predicted facies of these clustering methods and DNN in panels (Fig.5), we can find that Supervised algorithm (DNN) achieves better results than unsupervised algorithm. GMM also performs better than other unsupervised learning. Based on this experiment, a further experiment was conducted to implement different clustering methods with different dimensionality PCA (PCA 2, PCA 3, PCA 4). From the result of this experiment (Fig.6), it can be known that PCA 2 gives higher test accuracy than PCA 3, PCA 4 and no Dimensionality Reduction. For well 21/9-6A, GMM with PCA 2 gets the highest accuracy of 79.52%, compared with other unsupervised learning approaches.



DNN	KMeans	GMM	MeanShift	DBSCAN	Hierarchical
87.96	52.26	58.64	63.84	32.87	48.6

Fig. 5. Well-log panels summarizing the predicted facies of DNN and several clustering methods for Well 21/9-6A. The left column is the interpreted facies and the other columns display predicted facies of different methods corresponding to the x-axis of the histogram. The y-axis represents the test accuracy, displayed numerically in the last row of the table.

This study case is a preparatory work for following algorithmic implementation of VAEs since the performance of VAEs need to be evaluated on the bases of comparing against these basic classifiers. In this way, the results of VAEs can be analysed and evaluated relatively if there is room for improvement.



	None	PCA 2	PCA 3	PCA 4
KMeans	52.26	73.6	52.26	52.26
GMM	58.64	79.52	62.77	62.38
MeanShift	63.84	71.34	70.95	59.11
DBSCAN	32.87	0.16	51.79	34.74
Hierarchical	48.6	68.46	70.64	20.64

Fig. 6. Summary and Comparison for different clustering classifiers with and w/o DR for Well 21/9-6A. The y-axis is the test accuracy while the x-axis corresponds to no DR, PCA 2, PCA 3 and PCA 4. As the histogram is viewed from different colours, the bars each colour with respect to each clustering methods get test accuracies display in each row of the table. As the histogram is viewed from different clusters, each bar with respect to each DR approach get test accuracies display in each column of the table.

4. Proposed Approach

All the data provided by Ikon Science can be saved as csv files. Torch, Numpy and Pandas need be installed in Python.

Before loading the data, we need to know what is the structure of our dataset that each csv file with respect to each well. Each csv file consists of six columns of well logs（Vp, Vs, Density, GR, Porosity and Resistivity）. Each row corresponds to a specific depth. It should be noted that the well log data in these csv files has been scaled by the existing algorithm of RokDoc so there is no need to do data scaling in the new algorithm. I will take twenty-two wells as training set and two blind wells for test as what I have done in the study case part.

To load the data, I will use numpy.loadtxt to transform csv files into 'numpy.ndarray'.After loading these twenty-two csv files in Python, make sure there is no data loss. Then numpy.column_stack will be used to integrate twenty-two individual arrays into one array that used as the training set later. Similarly, test set can be gained from two csv files.

To train the model through Pytorch, the training set need to be transformed into Pytorch Tensors. A custom Pytorch Dataset will be implemented for training and test sets, but there is no need to do any transformation for them.

Next, the neural networks for VAEs need to be set. Here the PyTorch implementation of VAEs [14] by Lukas Mosser and Olivier Dubrule was taken as reference for networks construction. Here a simple fully connected DNN will be implemented for a variational autoencoder. VAEs workflow is:

High-Dimension Encoder -> *Low-Dimension Latent Space* -> *High-Dimension Decoder*

The DNN structure in the Lukas's github repository will be taken as a starting point [14]. The DNN has four torch.nn.Linear layers for the encoder and three for the decoder. In the four layers for encoder, first two are hidden layers and the each of other two layers works for mean a log-variance of the latent space variables, respectively. ReLU activation functions is needed for all the layers except the layers for mean and log-variance as well as the final output layer. No activation is required on the layers for mean and log-variance and a sigmoid should be employed on the final output layer for reconstruction.

Next, the VAE loss functions should be implemented to get the reconstruction loss and the KL-Divergence. Stochastic gradient descent will be used to for minimizing the KL-Divergence. The hyperparameters of the model also should be set.

After constructing the prototype networks, I will start to do training and visualize the reconstruction loss and the KL-Divergence. Different layer numbers, neuron numbers and activation function (e.g. Leaky ReLU) should to be tested by comparing the loss and KL-Divergence to get the best neural network architecture with appropriate hyper-parameters. The final step of VAEs is to apply the model to the test dataset and store the reconstructions and latent-variables. The reconstructions should be added into original dataset to enlarge the training set. The whole workflow is illustrated in Fig. 7.

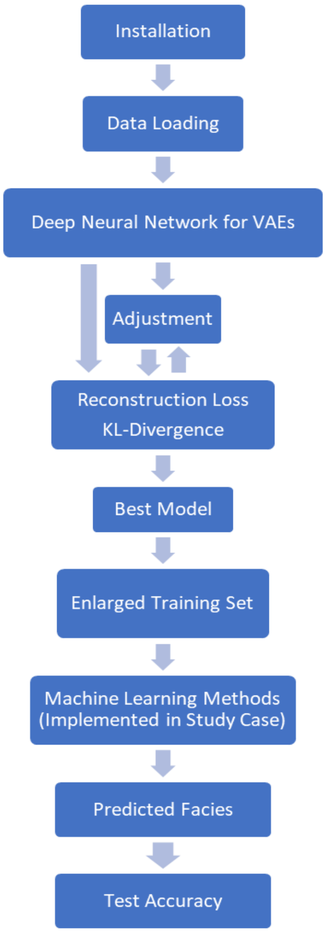
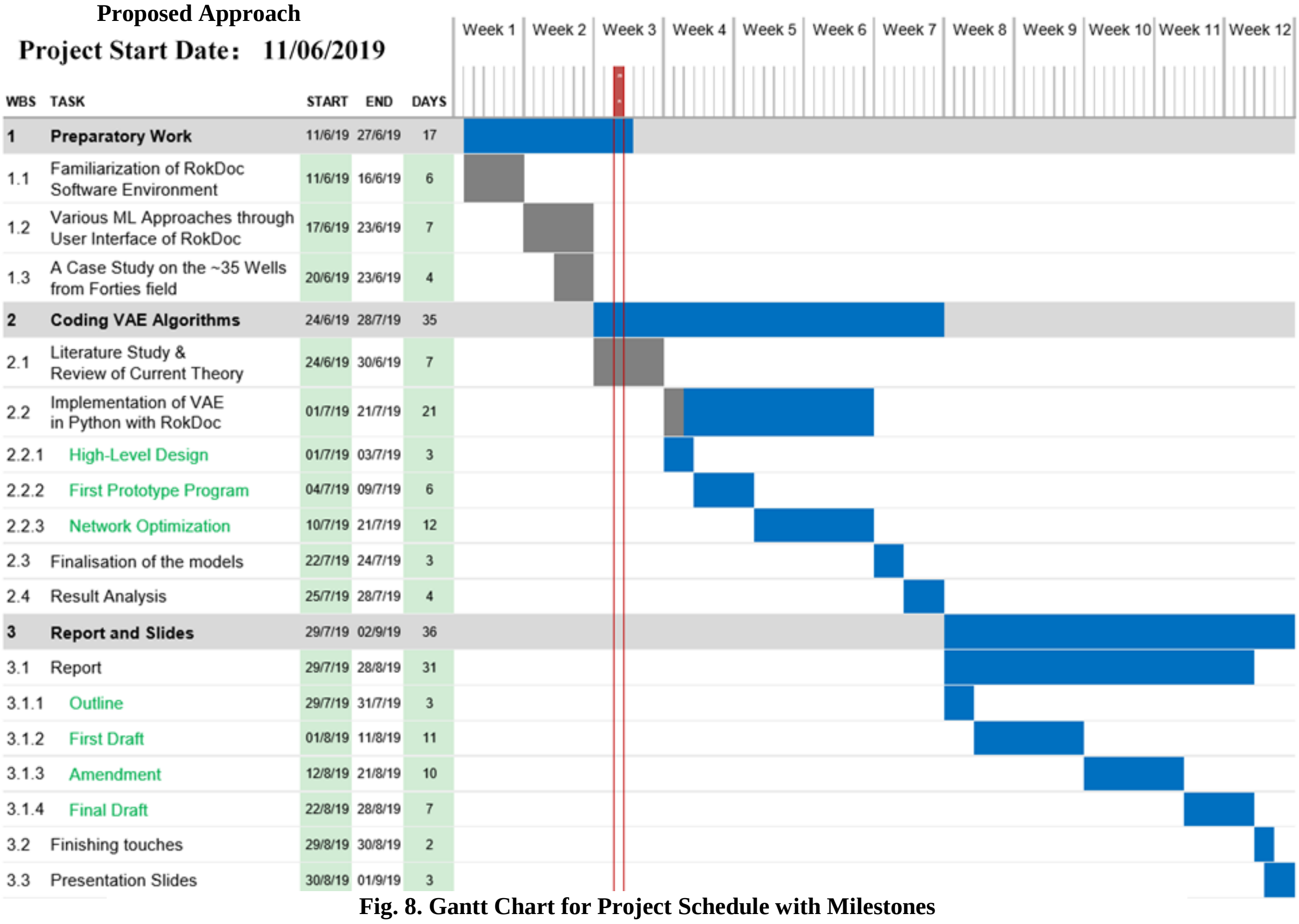


Fig. 7. Workflow of the Proposed Approach

One further task is to implement the same methods implemented in the study case on the enlarged dataset to take a comparison and result analysis. In this way, the superiority or disadvantages of VAEs can be illustrated. It is possible to enhance the training if I can combine the advantages of several methods and let them complement the deficiency of each other. If time permits, I will try other approaches to improve the accuracy for facies prediction. At first, I will investigate the Improved Deep Embedded Clustering (IDEC) which has been implemented on MISNT by Xifeng Guo in 2017 [15]. Furthermore, I will try to use Generative Adversarial Networks [10] to enlarge the training set and compare its performance with VAEs.

5. Project Milestone

The project plan is illustrated in the Gantt Chart below (Fig.8). The plan consists of three parts and each of them has a list of milestones with an approximate date for each deliverable.



Project Start Date: 11/06/2019

WBS	TASK	START	END	DAYS
1	**Preparatory Work**	11/6/19	27/6/19	17
1.1	Familiarization of RokDoc Software Environment	11/6/19	16/6/19	6
1.2	Various ML Approaches through User Interface of RokDoc	17/6/19	23/6/19	7
1.3	A Case Study on the ~35 Wells from Forties field	20/6/19	23/6/19	4
2	**Coding VAE Algorithms**	24/6/19	28/7/19	35
2.1	Literature Study & Review of Current Theory	24/6/19	30/6/19	7
2.2	Implementation of VAE in Python with RokDoc	01/7/19	21/7/19	21
2.2.1	High-Level Design	01/7/19	03/7/19	3
2.2.2	First Prototype Program	04/7/19	09/7/19	6
2.2.3	Network Optimization	10/7/19	21/7/19	12
2.3	Finalisation of the models	22/7/19	24/7/19	3
2.4	Result Analysis	25/7/19	28/7/19	4
3	**Report and Slides**	29/7/19	02/9/19	36
3.1	Report	29/7/19	28/8/19	31
3.1.1	Outline	29/7/19	31/7/19	3
3.1.2	First Draft	01/8/19	11/8/19	11
3.1.3	Amendment	12/8/19	21/8/19	10
3.1.4	Final Draft	22/8/19	28/8/19	7
3.2	Finishing touches	29/8/19	30/8/19	2
3.3	Presentation Slides	30/8/19	01/9/19	3

Fig. 8. Gantt Chart for Project Schedule with Milestones

References

[1] Matt V. *The Defining Series: Basic Well Log Interpretation*. Available from: https://www.slb.com/-/media/Files/resources/oilfield_review/defining_series/Defining-Log-Interpretation.pdf?la=en&hash=3DD25483EEE6EBAA69320AF7612AB9E9C4066993 [Accessed 25th June 2019].

[2] Ikon Science. *RokDoc Overview*. Available from: https://www.ikonscience.com/software/rokdoc [Accessed 24th June 2019].

[3] Ehsan N, Kenton P. Machine learning and learning from machines. *The Leading Edge*. 2018;37(12): 886-893. Available from: doi: http://dx.doi.org/10.1190/tle37120886.1

[4] Wikipedia. *Forties Oil Field*. Available from: https://en.wikipedia.org/wiki/Forties_Oil_Field [Accessed 24th June 2019].

[5] Watheq J. Al-Mudhafar. Integrating well log interpretations for lithofacies classification and permeability modeling through advanced machine learning algorithms. *Journal of Petroleum Exploration and Production Technology*. 2017; 7(4): 1023-1033. Available from: doi: https://doi.org/10.1007/s13202-017-0360-0

[6] Hao Li, Siddharth Misra. An improved technique in porosity prediction: A neural network approach. *IEEE Trans. Geosci. Remote Sens*.2017; 14(12): 2395-2397. Available from: doi: 10.1109/LGRS.2017.2766130

[7] P. M. Wong, T. D. Gedeon, I. J. Taggart. An improved technique in porosity prediction: A neural network approach. *IEEE Trans. Geosci. Remote Sens*.1995; 33(4): 971-980. Available from: doi: 10.1109/36.406683

[8] H.-C. Chang, H.-C. Chen, J.-H. Fang. Lithology determination from well logs with fuzzy associative memory neural network. *IEEE Trans. Geosci. Remote Sens*.1997; 35(3): 773-780. Available from: doi: 10.1109/36.582000

[9] V Tschannen, M Delescluse, MRodriguez, J Keuper. *Facies classification from well logs using an inception convolutional network*. ArXiv; 2017.Available from: https://arxiv.org/abs/1706.00613 [Accessed 24th June 2019].

[10] L Mosser, O Dubrule & MJ Blunt. Reconstruction of three-dimensional porous media using generative adversarial neural networks. *Physical Preview E*.2017; 96 (4): 043309-1 – 043309-17. Available from: doi: https://dx.doi.org/10.1103/PhysRevE.96.043309

[11] D. P. Kingma, M. Welling. *Auto-Encoding Variational Bayes*. 2013.Available from: https://arxiv.org/abs/1312.6114[Accessed 24th June 2019].

[12] Lilian Weng. *From Autoencoder to Beta-VAE*. Available from: https://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html [Accessed 24th June 2019].

[13] C. Doersch. *Tutorial on variational autoencoders*. Available from: https://arxiv.org/abs/1606.05908 [Accessed 24th June 2019].

[14] L Mosser, O Dubrule. [code] GitHub. Available from: https://github.com/msc-acse/ACSE-8-2018-19/blob/master/practical_6/Morning_6_Variational_Autoencoders_Solutions.ipynb [Accessed 24th June 2019].

[15] Xiaofeng Guo, Long Gao, Xinwang Liu, Jianping Yin. *Improved Deep Embedded Clustering with Local Structure Preservation*. Available from: https://www.ijcai.org/proceedings/2017/243 [Accessed 24th June 2019].