A Self-Organizing Map, Machine Learning Approach to Lithofacies Classification

Lan Mai-Cao
Department of Drilling and Production Engineering
Ho Chi Minh City University of Technology (HCMUT)
Ho Chi Minh City, Vietnam
maicaolan@hcmut.edu.vn
Corresponding author

Chau Le
Department of Drilling and Production Engineering
Ho Chi Minh City University of Technology (HCMUT)
Ho Chi Minh City, Vietnam
1450191@hcmut.edu.vn

***Abstract* - Nowadays, there are two main problems in data analysis, especially in lithofacies classification which are the big data and the fact that human cannot fully understand relationship between seismic attribute.[1] With our machine learning approach, we can not only solve these two problems but also reduce its time-consuming aspect and give an accuracy result even with non-experiences user. Typically, an exploration well is required to build a facies. However, only well log data is given and cores are not sampled. Given these circumstances and the conventional method like regression is unsolvable, our approach is taken by the use of 3 methods: (1) Using Principal Component Analysis (PCA) to select the most meaningful attributes, (2) grouping depth intervals which have similar facies into clusters by training Self-Organizing Map (SOM) and (3) Clustering to separate different facies into individual zones. Our case study focus on 2 well. The first one is Well Stuart, Brendon Hall, Enthought. 2016[2]. The second one is Well 1-X located in Oilfield Y, Vietnam. Our model is mathematically done by programming using Python language and then compared to Interactive Petrophysics(IP) software.**

Keywords - Machine learning; Principal Component Analysis (PCA); Self-Organizing Map (SOM); Clustering; facies classification.

I. INTRODUCTION

In the second half of the twentieth century, machine learning evolved as a subfield of Artificial Intelligence (AI) that involved self-learning algorithms that derived knowledge from data in order to make predictions. Instead of requiring humans to manually derive rules and build models from analyzing large amounts of data, machine learning offers a more efficient alternative for capturing the knowledge in data to gradually improve the performance of predictive models and make data-driven decisions. [5]There are three types of machine learning which are supervised learning, unsupervised learning, and reinforcement learning. Each type has its own application and unique algorithm but in this paper we only focus on unsupervised learning due to the lacking of outcome information in our case study. Furthermore, it is also taken into account that unsupervised learning can be automatically mined hidden pattern without human guidance so it is more closely to machine learning than other types. There is an unsupervised learning model for lithofacies classification which is SVM (Brendon Hall, Enthought .2016)[2]. In this paper, they introduced a machine learning model to classify the facies for Hugoton and Panoma gas fields and test the results with the actual facies data. Our model also use data based on this field but we introduce a new model to solve the problem which is Self-Organizing Map (SOM). The algorithm is mathematically calculated and visualized by Python language that based on some package like somoclu[6] and scikit-learn[7] with some modifications and then check with Interactive Petrophysics (IP) software. Our model would be best fit in case of lacking facies data or geological inexperience users. First unsupervised learning algorithm of our model is Principal Component Analysis (PCA). This is a linear mathematical technique to reduce a large set of variables (seismic attributes) to a smaller set that still contains most of the variation of independent information in the larger data set. It is traditionally relied on experience to choose which seismic attributes to input but now, with PCA, computer will automatically select the most valuable contribution attributes for the model. [8] Next is the Self-Organizing Map (SOM). Basically, this method will find a relationship between the input and the target by an equation: y=f(x) where y is denoted as the target and x is denoted as the input. Whereas SOM just need x to calculate y without finding the relationship. Finally, we use clustering to organize subgroups (facies) base on its dissimilarity.

In this paper, we would like to systematize the fundamental background of Self-organizing Map (SOM) and then apply this workflow to facies classification for two real cases – well Stuart from a University of Kansas class exercise on the Hugoton and Panoma gas fields.[2] and Well 1-X located at Oilfield Y in Vietnam. Finally, some discussions are presented based on the final results, which are compared with Interactive Petrophysics (IP) software.

II. LITERATURE REVIEW AND LIMITATIONS OF CURRENT TECHNIQUES.

One of the essential steps in reservoir characterization is facies classification which is the study of rock distribution along the interest domain of depth. Facies classification can be done by many methods but in general, there are two scenarios: perfect information and incomplete information.

The first one is to take rock sample from the target well, using lab analyses to define rock component (M.Flores et al., 1989) [3]. This method gives a very precison result, low risk but also not economically viable approach. The second one is using discriminant analyses which consists of getting rock samples from certain areas, collecting indirect test data from all target areas (including the rock sample locations) and trying to infer the mapping function between input (indirect measurements) and output (facies) (Inhaúma N. Ferraz et al., 2004) [4]. In addition, due to the low accuracy rates and the complexity of well log data, it is difficult and time consuming to manually analyze a huge amount of these digitized data. Thus, another alternative approach has come up which helps this interpretation work becomes easier and more accurate. It is machine learning approach with respect to Principal Component Analysis (PCA); Self-Organizing Map (SOM) and Clustering.

III. NEW PROPOSED METHODOLOGY

The whole process consists of three main stages: 1) PCA, 2) Training SOM and (3) Clustering. This workflow can be viewed in Figure 1.[9].

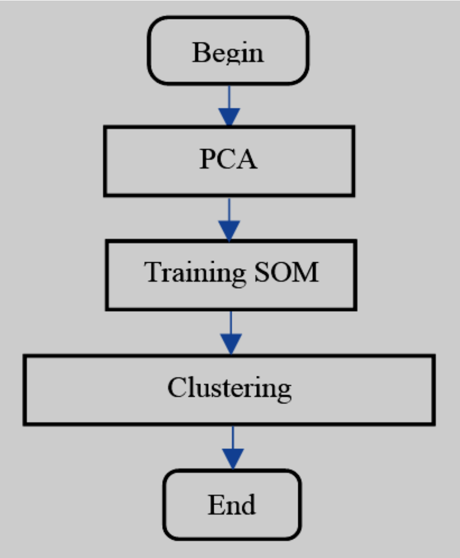

Figure 1. General workflow for facies classification.

A. Begin (Data Preparation)

1) For Well Stuart in Hugoton and Panoma Gas Fields

Given well log data for Well Stuart, input data consists of available log curves such as (GR, PHIND, RELPOS, ILD_LOG10, PE, DELTAPHI, NM_M) are selected with up to 474 data points. The input data matrix has, therefore, 474 rows and 7 columns (corresponding to the number of log curves). This matrix is an input for next step.

2) For Well 1-X located in Oilfield Y, Vietnam

Given well log data for Well 1-X, input data consists of available log curves such as (DT, NPHI, RHOZ, GR) are selected with up to 15600 data points. The input data matrix has, therefore, 15600 rows and 4 columns (corresponding to the number of log curves). This matrix is an input for next step.

B. Principal Components Analysis (PCA):

The input data matrix from previous step is normalized before starting to analyze principal components (PCA) so that each data point has the same range in [0,1]. The normalization is done by calculating the mean and standard deviation of each log and then normalizing the data by subtracting the mean and dividing by the standard deviation.

$$\frac{X - \mu}{\sigma} \qquad (1)$$

where: X is original data, μ is mean and σ is standard deviation

After normalizing, we calculate covariance matrix and from that, find eigenvectors and eigenvalues. These processes are called Principal Component Analysis (PCA). Generally, PCA transformed input data from n curves into p curves (p<n) but still contain enough information that original curves had. Because the SOM is presented in a 2-dimensional map, thus PCA is applied to reduce 4 curves into 2 new curves.

C. Training Self-Organizing Map (SOM):

At first, a square map grid in 2 dimensions is constructed with Total nodes (N) = mapwidth2 [8] and coordinates of map grid will be calculated in order to position each map unit in space. Each node has a weight vector with dimension equal to the number of input curves and denoted as formula (1) [6].

$$\overrightarrow{W_j} = \{W_{jk}: j = 1, \dots N; k = 1,2,3,4\} \qquad (2)$$

where: j is the index number of node, k is the number of curves, N is the total of nodes

Once the map grid is created, we take out two main principal components (PCs) which have largest eigenvalues to calculate initial weight values for map grid. Because we want to present the original data in 4D into 2D map grid, thus two first eigenvectors are chosen. Now 2 largest eigenvectors are taken out, we can denote them as:

$$\overrightarrow{Eig1} = \begin{bmatrix}\alpha 1\\ \alpha 2\\ \alpha 3\\ \alpha 4\end{bmatrix}; \overrightarrow{Eig2} = \begin{bmatrix}\beta 1\\ \beta 2\\ \beta 3\\ \beta 4\end{bmatrix} \qquad (3)$$

Consider that nodes has their coordinate is (x,y); eig1 is presented as x-axis and eig2 is presented as y-axis. Weight value of nodes is a linear combination between its coordinate and two eigenvectors. Finally, map grid is constructed that contain weight values in each node.

Once the weight values are initialized, the "Best Matching Unit" (BMU) is calculated, that is the node which most closely matches the input data given (similarity). This is calculated by the Euclidean distance [8] between each node's weight vector and the current input vector which is shown in a formular (4) below. The node with a weight vector closest to the input vector is tagged as the BMU.[10],

$$D_j = \sqrt{\Sigma_{k=1}^{4}(V_{nk} - W_{jk})}^2 \text{ (j: j = 1, 2…, N)} \quad (4)$$

Each iteration, after the BMU has been determined, the next step is to calculate which of the other nodes are within the BMU's neighbourhood or we can call it as an "effective radius" of BMU. This radius is initialized as a half of map grid (Ro) [8]. Then it will shrink over time until it becomes one node (BMU). To express the decrease in radius, exponential decay function is applied [6] as formula (4):

$$R(t) = Ro \times \exp\left(\frac{-t * \log(Ro)}{T}\right) \quad (5)$$

where : t is current iteration, Ro is initial radius, T is maximum iteration.

After calculating the region that was effected by the BMU, all nodes within this "effective radius" will be adjusted their weight values to become more similar to the input vector [6]. This new weight value is calculated in formula (5).

$$Wji\ (t+1) = Wji\ (t) + \theta(t) \times alpha(t) \times (Vi(t) - Wji(t)) \quad (6)$$

Alpha(t) : Learning rate at current iteration t.

$$alpha(t) = alpha\_ini \times \exp(-t/\lambda) \quad (7)$$

where: t is current iteration, λ is time constant, θ(t) is the amount of influence for a node based on its distance to the BMU.

$$\theta(t) = \exp(-Ud2/2R2(t)) \quad (8)$$

where: Ud is the Distance between each map unit computed by Euclidean distance.

Upon completion, the map is created and be ready for next step which is building U-matrix. The purpose of the U-matrix is to give a visual representation of the topology of the network. It depicts the average Euclidean distance between the code book vectors of neighboring neurons. Let N(j) denote the immediate neighbors of a node j. Then the height value of the U-matrix for a node j is calculated as: [6].

$$U(j) = \frac{1}{N(j)} \times \sum_{i \in N(j)} d(wi; wj) \quad (9)$$

D. Clustering:

The trained map is taken as an input for Clustering, including two main steps: the 1st step, total nodes are grouped into k clusters by K-means technique. The 2nd step, k clusters are merge into 9 groups using Hierarchical clustering.

➢ K-means algorithm :

- After training, total nodes are grouped into 15 clusters (k=15). k is initialized based on the total rows of the input matrix.
- Choose randomly 15 nodes to be « Centroids ».
- Calculate distances from each node to 15 Centroids by Euclidean distance.
- Group nodes to Centroids based on minimum distance.
- Update new centroids.
- Re-calculate distances from each node to new centroids.
- If centroids do not change values then stop iteration, if no back to step 5.

➢ Hierarchical Clustering :

After K-means is finished, N nodes were grouped into 15 clusters. These clusters, then, were arranged in a suitable position. Two closest groups were merged into a new cluster. Then update distance matrix between new cluster to the old nodes based on Linkage criteria. Continuously until there are just one cluster remaining. There are five types of Linkage :

- Single linkage (minimum distance between old nodes and new cluster).
- Complete linkage (maximum distance between old nodes and new cluster).
- Simple average (average distance between old nodes and new cluster).
- Group average (average distance between old nodes and each member of new cluster).
- Ward's linkage (minimization of the increase of sum of squares).

Ward's linkage is used in this study because of its good results and appropriate separation [8]. Once Hierarchical clustering is done, we will have groups of rocks. Each cluster contains the number of nodes that belong to it. In trained map, based on the coordinate of total nodes in a cluster, we can determine the boundary for each cluster. With each node, a certain depth is assigned so that we can draw a facie log with the value is the cluster index). To build this log, three main tasks are done as follows :

- At a certain depth, we have 1 vector V with its components includes the value of seven curves.

- Determine its BMU then locate the BMU in cluster zones by compare its coordinate to the boundary of each cluster.
- Therefore, we know what facies that V belongs to.

E. End (Output)

Upon completion, facies of rocks are classified. A solution applied for well 1-X and well Stuart is completed.

IV. RESULTS AND DISCUSSION

Workflow is applied to 2 case study of well Stuart and Well 1-X. It is required that facies classification be performed along the well's depth.

In order to solve this problem, regression method and SOM are considered. With regression method, two parameters from well log data and cores information are needed in order to create a relationship between depth and relevant facies. Hence, this method is unsolvable due to the inhomogenity of facies along the depth and missing information from cores. As a result, SOM is applied alternatively.

A. For Well Stuart in Hugoton and Panoma Gas Fields

In this case, we use the data of Well Stuart in Hugoton and Panoma gas fields to compare our method with SVM method used by Brendon Hall, Enthought 2016[2].

1) Training SOM :

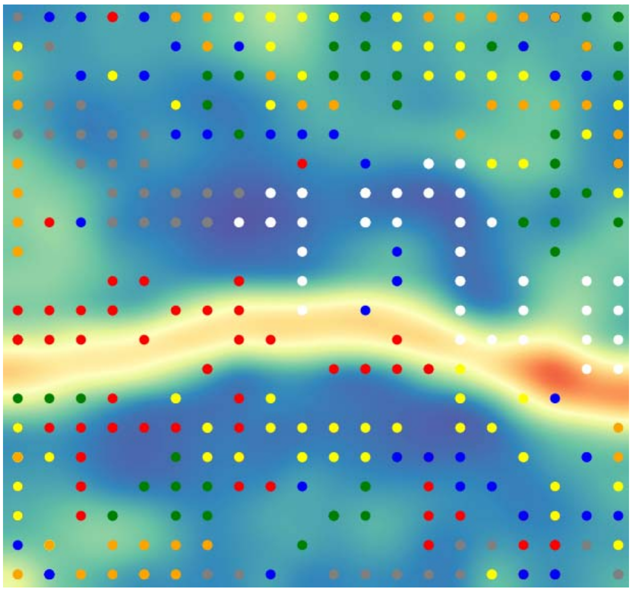

Figure 2. U-matrix after training by using Python.

In figure 2, by using Python laguage we created the U matrix which show the topology of the network. Those color dots inside represent the best matching units (BMU).

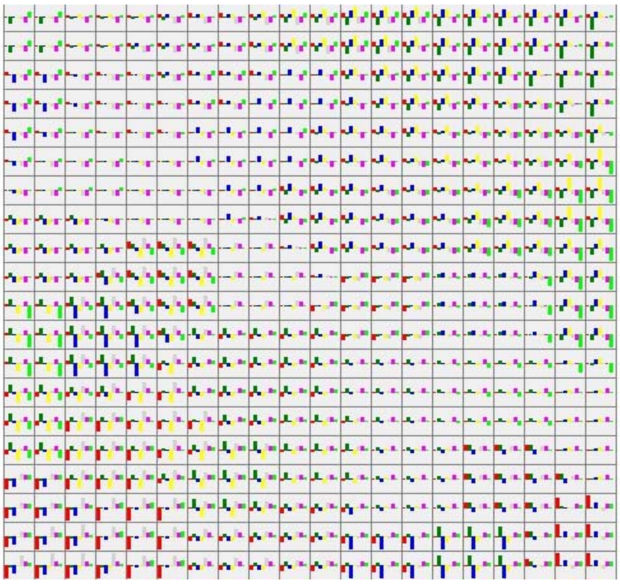

Figure 3 : SOM map result after training by using IP

Figure 3 shows the SOM map after training by using IP. Each node in the map has a weighting value for each of the different input vectors i.e. we are using four input curves then the node will store it's weighting for each of those four curves. These weightings are visually presented on the map as bar graphs, which are colored by the color of input vectors [8].

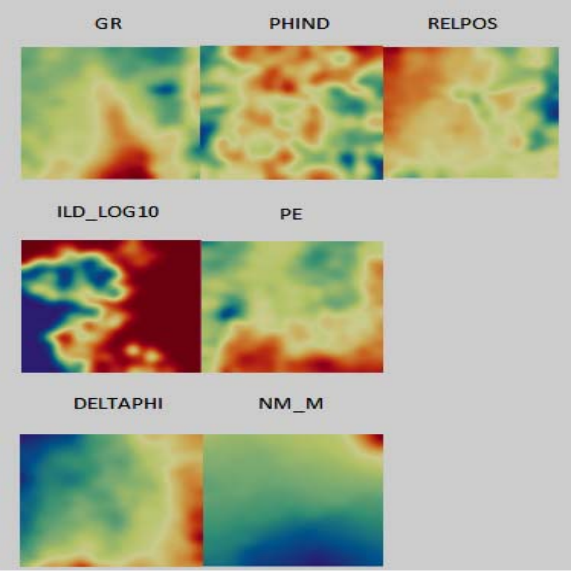

Figure 4: 7 curves distribution by using Python

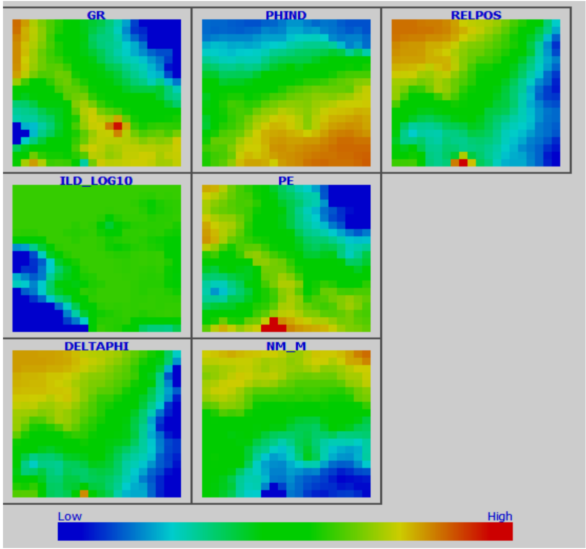

Figure 5 : 7 curves distribution using IP

In figure 4 and 5, there are seven small graphs that represented seven input curves (GR, PHIND, RELPOS, ILD_LOG10, PE, DELTAPHI, NM_M) with their values are distributed from low to high. In both figure, we can observe the similar pattern.

2) *Clustering :*

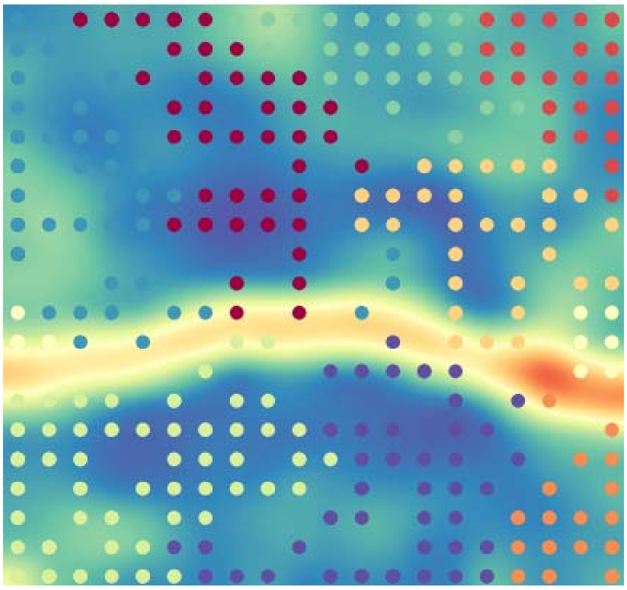

Figure 6: Ward linkage using Python in Well Stuart

Figure 6 shows the U-matrix after using Ward linkage. The best matching units now are just 9 colors that represent 9 clusters like in IP, These 9 clusters also represent 9 facies of rock just like the SVM method.

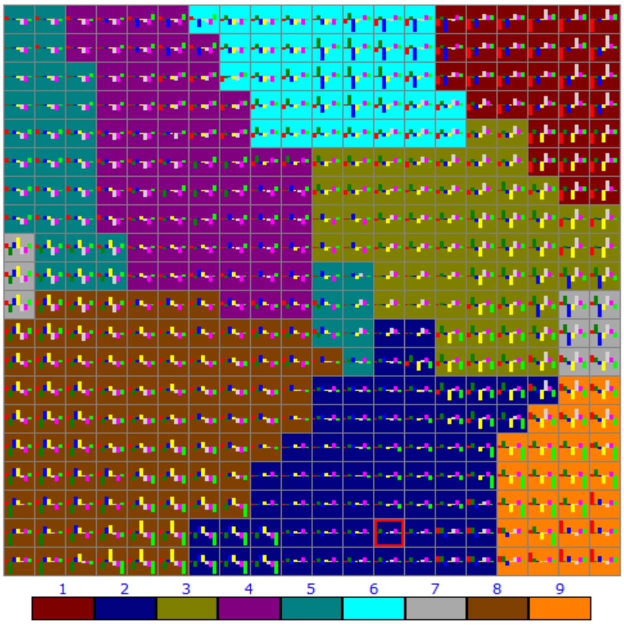

Figure 7: Ward linkage clustering using IP in well Stuart

Figure 6 and figure 7 show very similar pattern thus we can confirm our algorithm is correct. However, in IP we can observe better visualization likes boundary, colors,erc…

After having the clustering map we can predict the facies at certain depth by determining the BMU of a chosen input vector V and then compare its coordinate with the cluster's boundary. Therefore, we know what what facies that V belongs to.

In this paper, we use Ward's linkage because of good separation as mentioned above. In Ward's linkage, the sum of square error (SSE) is denoted as the error between old cluster and new cluster, hence, when the number of members in one cluster increases, it leads this error increases quickly, thus, this linkage can measures the increase in sum of square error (SSE) to be at least. Consequently, Ward's linkage is better than the others.

B. For Well 1-X located in Oilfield Y, Vietnam

We apply our workflow in case of well 1-X located at oilfield Y in Vietnam. Due to the similarity of training SOM step, we will alternatively show the last result of clustering step to compare whether the algorithm can be applied in other cases or not.

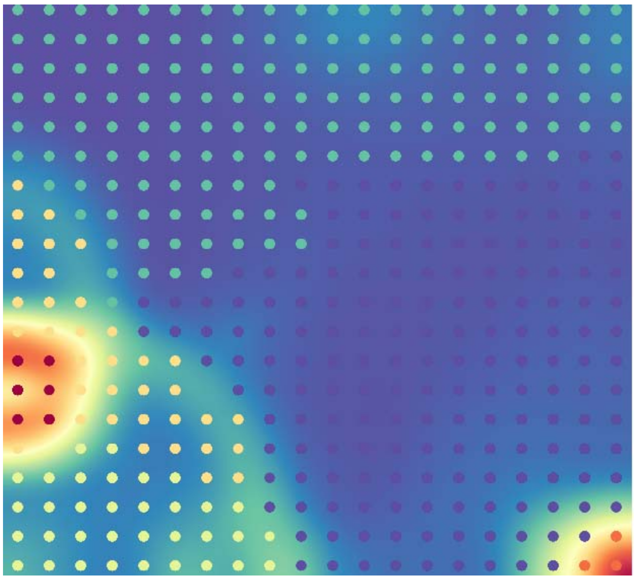

Figure 8 : Ward linkage using Python in Well 1-X

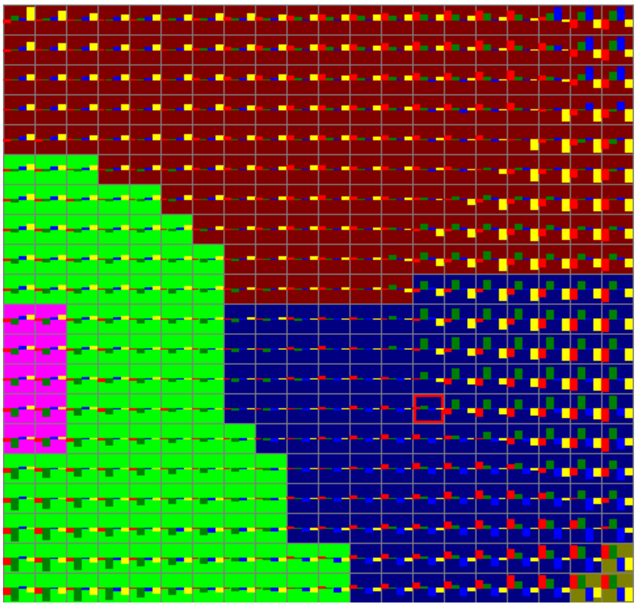

Figure 9 : Ward linkage using Python in Well Stuart

Figure 8 and Figure 9 illustrate the facies classification result in Well 1-X located in Oilfield Y Vietnam. Again, both figure show similar pattern which are 5 clusters or in other word, 5 facies. This result suggested that our algorithm can be applied in different area with big data problems.

V. CONCLUSIONS

In this study, we have systematized the fundamental background of Self-organizing Map. It is also visualized through the U-matrix and we can directly see BMUs. General workflow is presented from this work and then applied it for Well Stuart in order to compare with SVM approach by Brendon Hall, Enthought 2016.

Moreover, some new adjustments are made like normalization to standardlize the input data, elliminate the gap in scale value between curves. It helps giving the more accurate and realistic result.

In addition, we also introduce some mathematically calculation like Principal Component Analysis and Hierarchical Clustering to clarify the process. Moreover, properties of facies log are also indicated, including the shape, the measurement and its values relate to depth. Self-organizing Map construction is also detailly codified as the input of Clustering process. Although there are 5 types of linkage that can be used at which certain situations, we conclude that Ward linkage should be used more often than others because it measures the increase in SSE to be at least.

Comparison between using IP software and Python language are made. Same input data and same workflow are taken into account by these 2 approach which show similar pattern and some slightly diiferent due to randomly choosing input vector at the beginning of training SOM and randomly choosing centroids at step clustering.

ACKNOWLEDGMENT

We would like to thank Office for International Study Program (OISP) and Department of Drilling and Production Engineering, Faculty of Geology and Petroleum Engineering, Ho Chi Minh City Univesity of Technology (HCMUT), Vietnam for the support and encouragement.

REFERENCES

[1] Rocky Roden and Deborah Sacrey, "Seimic Interpretation with Machine Learning," GeoExpo 2016

[2] Brendon Hall, Enthought , "Facies classification with machine learning" , SEG: The Leading Edge, vol. 35, Dec. 2016, pp. 906-909, doi:10.1190/tle35100906.1

[3] Romeo M. Flores and C. William Keighin, "Petrology and Depositional Facies of Siliciclastic Rocks of the Middle Ordovician Simpson Group, Mazur Well, Southeastern Anadarko Basin, Oklahoma", USGS Publications 1866-E, 1989..

[4] Inhaúma N. Ferraz, Ana C. Bicharra Garcia, Cristiano L. Sombra, "Neural Networks and Fuzzy Logic for improving Facies recognition from well log data", ADDLabs Publications, 2004.

[5] Sebastian Raschka & Vahid Mirjalili, "Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow", 2nd Edition, September 20, 2017

[6] Peter Wittek, Shi Chao Gao, Ik Soo Lim, Li Zhao, "Somoclu: An Efficient Parallel Library for Self-Organizing Maps", Journal of Statistical Software, 78(9), pp.1–21, DOI:10.18637/jss.v078.i09.

[7] Pedregosa et al, "Scikit-learn: Machine Learning in Python", JMLR 12, pp. 2825-2830, 2011.

[8] Senergy Software Ltd, IP Help Manual version 4.2, 2013

[9] Lan Mai-Cao, Tram Bui, Chau Le, "Application of Self-Organizing Map (SOM) in lithofacies classification", unpublished

[10] Kohonen, T.: Self-Organizing Maps, vol. 30. Springer, Heidelberg (2001).