University of Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study program / Specialization: Petroleum Geoscience Engineering	Spring semester, 2021 Open
Writer: Syeda Vania Mansoor	…… …. (Writer's signature)
Supervisor: Arild Buland	
Thesis title: Multi-Attribute Seismic Analysis Using Unsupervised Machine Learning Method: Self-Organizing Maps	
Credits (ECTS): 30	
Keywords: Unsupervised Machine Learning Seismic attributes analysis Self-Organizing Maps Statfjord Field	Pages : 44 + enclosures : 8 Stavanger, 30 June 2021

Multi-Attribute Seismic Analysis Using Unsupervised Machine Learning Method: Self-Organizing Maps

By

Syeda Vania Mansoor

MSc Thesis

Presented to the Faculty of Science and Technology

University of Stavanger

June 2021

Acknowledgements

I would like to thank my mother and siblings for their constant support and love. Who have always believed in my abilities and have been by my side to guide and help me throughout my academic endeavours. Thank you for your valuable advices, words of encouragement and prayers to keep me pushing forward to achieve my goals.

Next, I would like to thank my supervisor, Arild Buland, for giving me an opportunity to work on such an interesting topic that has developed my interest in Machine Learning even more. Thank you for guidance, advices and support.

Thank you Equinor ASA for providing the dataset.

Last but not least, I would like to thank all my friends here in Stavanger that have always been fun, kind and supportive. I have made many precious memories on this journey that I will cherish forever.

Abstract

Seismic attributes are a fundamental part of seismic interpretation and are routinely used by geoscientists to extract key information and visualize geological features. By combining different findings from each attribute, they can provide a good insight of the area and help overcome many geological challenges. However, individually analyzing multiple attributes to find relevant information can be time-consuming and inefficient, especially when working with large datasets. It can lead to miscalculations, errors in judgement and human bias. This is where Machine Learning (ML) methods can be implemented to improve existing interpretations or find additional information. ML can help by handling large volumes of multi-dimensional data and interrelating them. Methods such as Self Organizing Maps (SOM) allow multi-attribute analysis and help extract more information as compared to quantitative interpretation. SOM is an unsupervised neural network that can find meaningful and reliable patterns corresponding to a specific geological feature (Roden and Chen, 2017).

The purpose of this thesis was to understand how SOM can help make interpretations of direct hydrocarbon indicators (DHI) in the Statfjord Field area easier. Several AVO attributes were generated to detect DHIs and were then used as input for multi-attribute SOM analysis. SOMPY package in Python was used to train the model and generate SOM classification results. Data samples were classified based on BMU hits and clusters in the data. The classification was then applied to the whole dataset and converted to seismic sections for comparison and interpretation.

SOM classified seismic lines were compared with the results of the AVO attributes. Since DHIs are anomalous data, they were expected to be represented by small data clusters and BMUs with low hits. While SOM reproduced the seismic reflectors well, it did not define the DHI features clearly for them to be easily interpreted. Use of fewer seismic attributes and computational limitations of the machine could be some of the reasons behind not achieving desired results.

However, the study has room for improvement and the potential to produce meaningful results. Improvements in model design and training, and also the selection of input attributes are some of the areas that need to be addressed. Furthermore, testing other Python libraries and better handling of large datasets can allow better performance and more accurate results.

Contents

List of Figures

List of Tables

Chapter 1: Introduction

Machine Learning (ML) has been applied in geosciences since the 1970s (Dramsch, 2020), but over the last few years, there has been a boost in the application and development of ML by geoscientists in more practical settings. Most of the techniques are being adopted from other fields of research, where ML is rapidly developing. Workflows currently used require time-consuming manual labour and experienced-based decisions that often lead to miscalculations, errors in judgment and human bias. Due to lack of technology, time, and high costs, large datasets are inefficiently used that leaves a void in the important information that can be filled if all the relevant data is used. However, now with the availability of more open source libraries such as sklearn, scipy, tensorflow, as well as geoscience-specific libraries such as Segyio, Fatiando e Terra and PetroPy, and GPU-enabled high-performance computing, geoscientists are able to better handle 'big data' and make the most of it. ML models allow for more accurate and precise interpretations, which saves time and eliminates human bias. Furthermore, the emergence of data analytics platforms with readily available ML workflows allows geoscientists to focus on solving geoscience problems rather than developing algorithms and codes for specific cases (Larsen et al., 2018).

Attribute analysis allows extraction of concealed information from prestack or stacked seismic data that can help in delineating prospects, determining facies distribution, enhancing fractures and faults, and even highlight direct hydrocarbon indicators (DHIs) (Burnett et al. 2003; Castagna et al. 2003; Chopra and Marfurt 2007; Farfour et al. 2012; Hossain, 2020). Seismic attributes are categorized as geometric, instantaneous, spectral decomposition, seismic inversion, and AVO, and collectively they make up hundreds of individual attributes (Brown, 2004; Chen and Sidney, 1997; Chopra and Marfurt, 2007; Roden and Chen, 2017).

DHIs are frequently used in petroleum companies to evaluate prospect risks and determine precise well locations (Roden et al., 2005; Fahmy and Reilly, 2006; Forrest et al., 2010; Roden et al., 2012; Rudolph and Goulding, 2017). DHIs are seismic amplitude

anomalies caused by changes in rock physics properties (P and S wave velocities and density) due to the presence of hydrocarbon as compared to the reservoir rock holding the hydrocarbon or the brine solution present in the reservoir. Several types of DHIs include flat spots, bright spots, phase change at the fluid contacts, amplitude variation with offset, and amplitude conformance to structure (Roden et al., 2012). Seismic attributes generally applied for identifying DHIs are AVO, instantaneous, and inversion attributes. AVO attributes such as intercept times gradient, far offset-minus near offset-times the far offset, Lamda/Mu/Rho and fluid factor can highlight the hydrocarbon-bearing reservoirs. Whereas, sweetness, average energy, and amplitude envelope can make the amplitude anomalies stand out against the background, in turn identifying potential hydrocarbon accumulation. However, even with abundant seismic attributes that can be applied, the interpretation of DHI characteristics is not straightforward. Using multiple attributes can be intricate and complicated (Roden and Chen, 2017).

Machine Learning can help by handling large volumes of multi-dimensional data and interrelating them. Using appropriate algorithms, computers can find meaningful and reliable patterns that correspond with the presence of hydrocarbons (Roden and Chen, 2017). Methods such as Self Organizing Maps (SOM) allow multi-attribute analysis and help extract more information as compared to quantitative interpretations. It is an unsupervised, robust classification method that reduces the dimensionality of multi-dimensional data. It better represents the seismic characters and detects geologic trends in the area, while allowing ‘muting’ irrelevant data (i.e. seismic noise) (Manouchehri et al., 2020).

SOM has been successfully applied by many geoscientists, such as Rocky Roden, for facies classification, reservoir characterization, delineating existing and new prospects, and understanding structural trends in the area. Furthermore, recognizing the advantages of multi-attribute analysis, Paradise software by Geophysical Insight offers a built-in ML module for SOM analysis. This allows geoscientists who are not familiar with programming to easily handle and combine several different attributes and gain new information.

1.1 Aim and Objectives

1.1.1 Aim

Geoscientists can benefit tremendously by utilising all of the useful information provided by different seismic attributes. One way to combine and collectively understand the trends in the data from the attributes is through SOM analysis. The aim of this study is to understand how Self-Organizing Maps can help to identify DHI anomalies in the Statfjord area or reinstate present interpretation to increase confidence.

1.1.2 Objectives

- Determine and calculate different seismic attributes for identifying DHIs
- Identify and interpret hydrocarbon presence and related DHIs on the seismic attributes using Petrel.
- Using SOMPY library in Python, design and train SOM model specific to the data set.
- Interpret results from SOM analysis and compare with the results of seismic attributes.
- Determine whether SOM helps to refine interpretation and identify new anomalies.

Chapter 2: The Statfjord Field

The Statfjord Field is one of the largest and oldest producing fields in the North Sea. It is located on the border between the Norwegian and UK sectors in the northern part of the North Sea, geologically known as Tampen Spur (Figure 2.1) (Gibbons et al., 2003; Norsk Petroleum, 2021). It was discovered by Mobil Exploration Norway in 1974 and started production in 1979. Today, it is jointly owned by Equinor ASA (44.34%), Spirit Energy (34.29%) and Var Energi (21.37%). The hydrocarbon-bearing reservoir of the field covers approximately 24km by 4km area, making Statfjord the largest oil field in the Northern North Sea (Roberts et al., 1987; Gibbons et al., 2003).

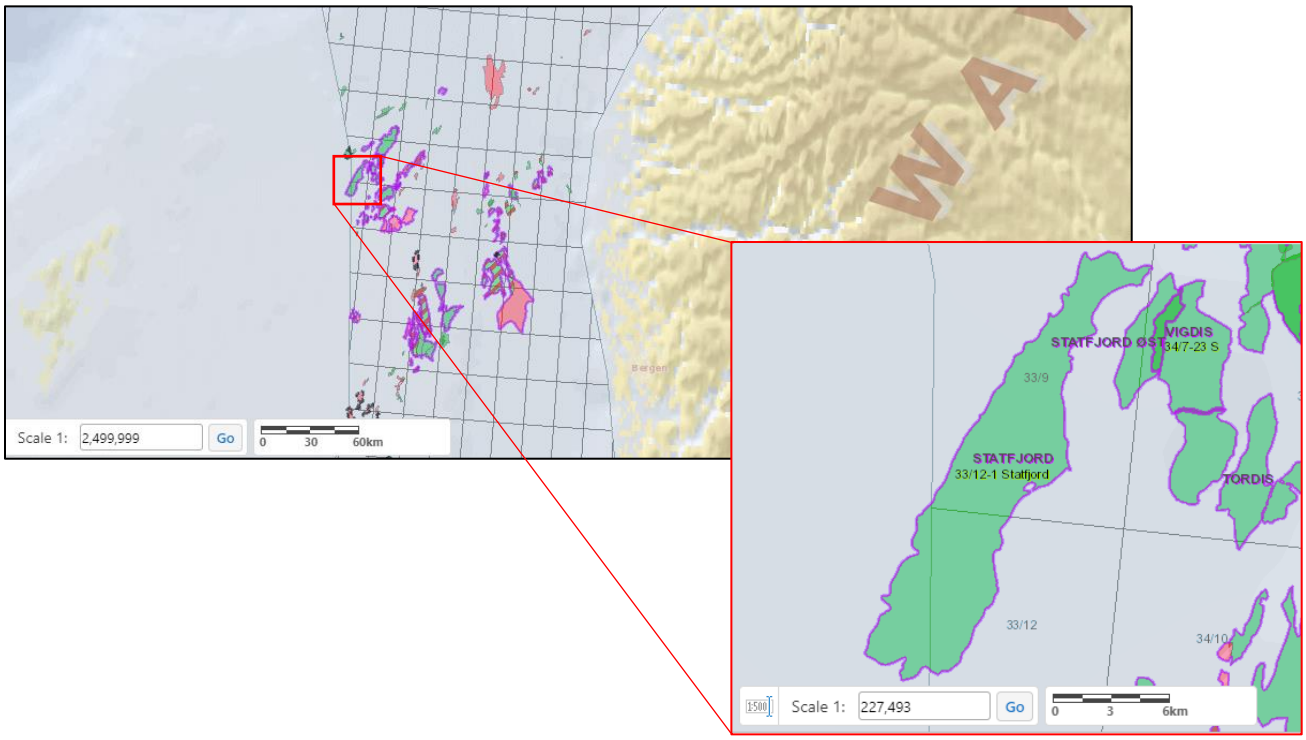

Figure 2.1: Location of the Statfjord Field (Modified after www.npd.no)

2.1 Regional Setting

The Statfjord Field (Kirk, 1980) lies within the East Shetland Basin, along the western margin of the North Sea Rift System (Figure 2.2a). The field is situated along the crest of

a NE-SW trending fault block that is slightly dipping towards the northwest (Figure 2.2c). Along the east flank, the field comprises of smaller, faulted, and rotated compartments that are referred to as gravitational collapse structures. It is bounded in the south west by the major Brent Fault, and in the north east by a horst structure (Hesthammer et al., 1999).

Tectonic history of the area can be defined by at least two major rifting events that followed the Devonian thinning and regional stretching of the Caledonian crust (Hesthammer et al., 1999). The first rift phase in the Permo-Triassic led to the opening of the Viking Graben (Badley et al. 1984, 1988; Beach et al. 1987; Roberts et al. 1995). The second rift phase that occurred in the latest middle Jurassic to earliest Cretaceous, led to an extension in generally NW-SE direction (Figure 2.2b) (Roberts et al., 1990). This phase was followed by a rise in sea level that resulted in the burial of the Triassic and Jurassic aged reservoir formations. The burial continued in the Cretaceous and Paleocene ages, during the thermal subsidence of the whole North Sea Basin in the post-rift stage (Gibbons et al., 2003).

Structurally, the Statfjord Field can be divided into two domains: a heavily faulted east flank characterized by rotated slide blocks and associated erosional debris, and a relatively undeformed section, covering most of the field with W-NW dipping strata and several NW-SE oriented cross faults that offset the base Cretaceous. The two domains are separated by the base of slope failure (BSF) surface, which can be mapped seismically (Hesthammer et al., 1999). The faulted blocks of the east flank cut through the reservoir formations (Gibbons et al., 2003). Most of the traps for petroleum are found in the rotated fault blocks along both margins of the Viking graben, formed as a result of Late Jurassic rifting (Faleide et al., 2010).

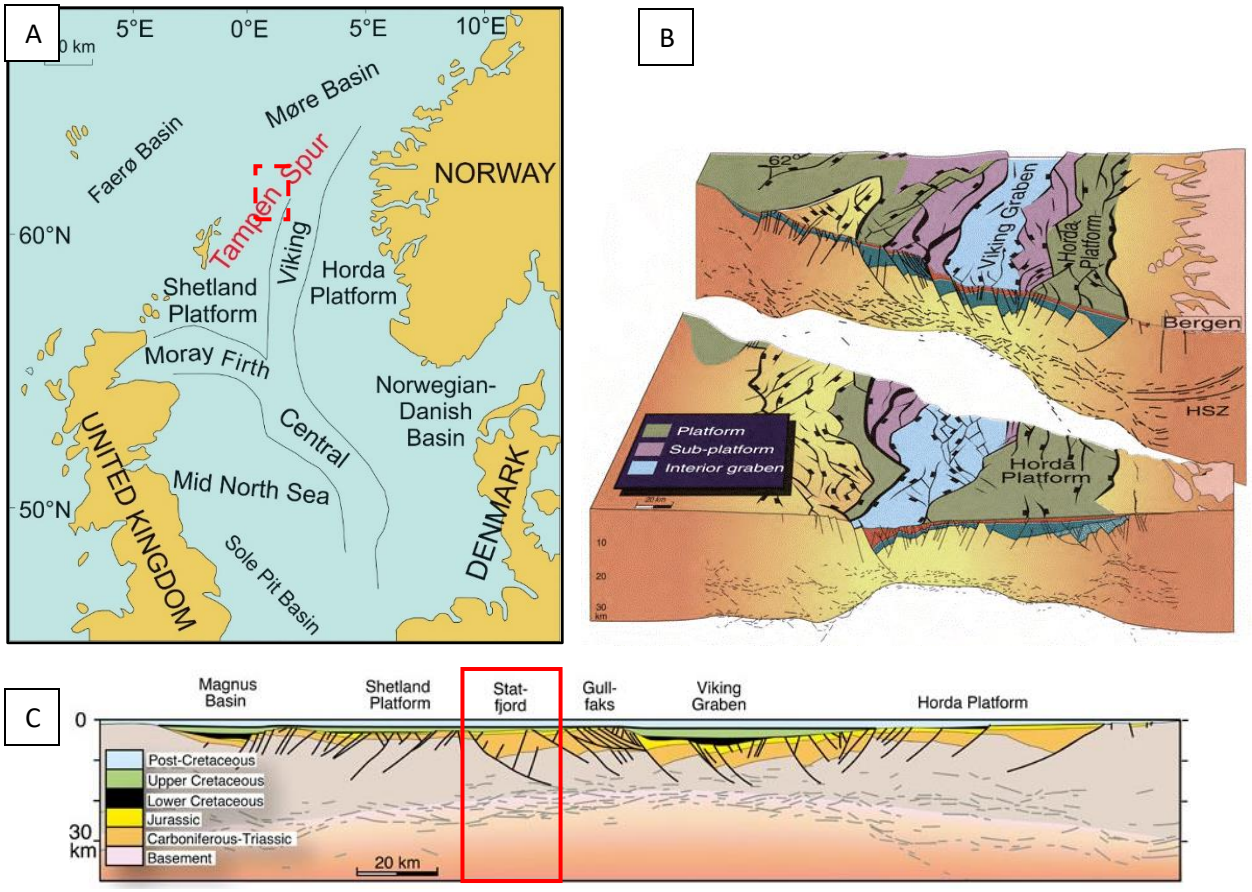

Figure 2.2: a) Map of the northern North Sea and regional setting of the Tampen Spur area. (Modified after Ketzer, 1999). b) Tectonics overview of the northern North Sea. (Modified after Fossen et al., 2000. c) Regional profile across the norther North Sea and the location of different fields, including the Statfjord Field. (Modified after Gawthorpe et al., 2019).

2.2 Stratigraphy of the Statfjord Field Reservoirs

The principal reservoirs of the Statfjord Field are Late Triassic–Early Jurrasic age Statfjord Formation and Mid-Jurassic age Dunlin and Brent Groups. They vary in thickness from 20 to 200m in the Statfjord Field area, and their average net-to-gross (N/G) is 0.75 for Brent Group, 0.05-0.45 for Dunlin and 0.6 for Statfjord reservoirs. Average porosity and permeability go as high as 27% (Brent Group) and 470mD (Statfjord Formation), respectively. Trap types for these reservoirs can be structural or stratigraphic, and in some places, a combination of both (Gibbons et al., 2003).

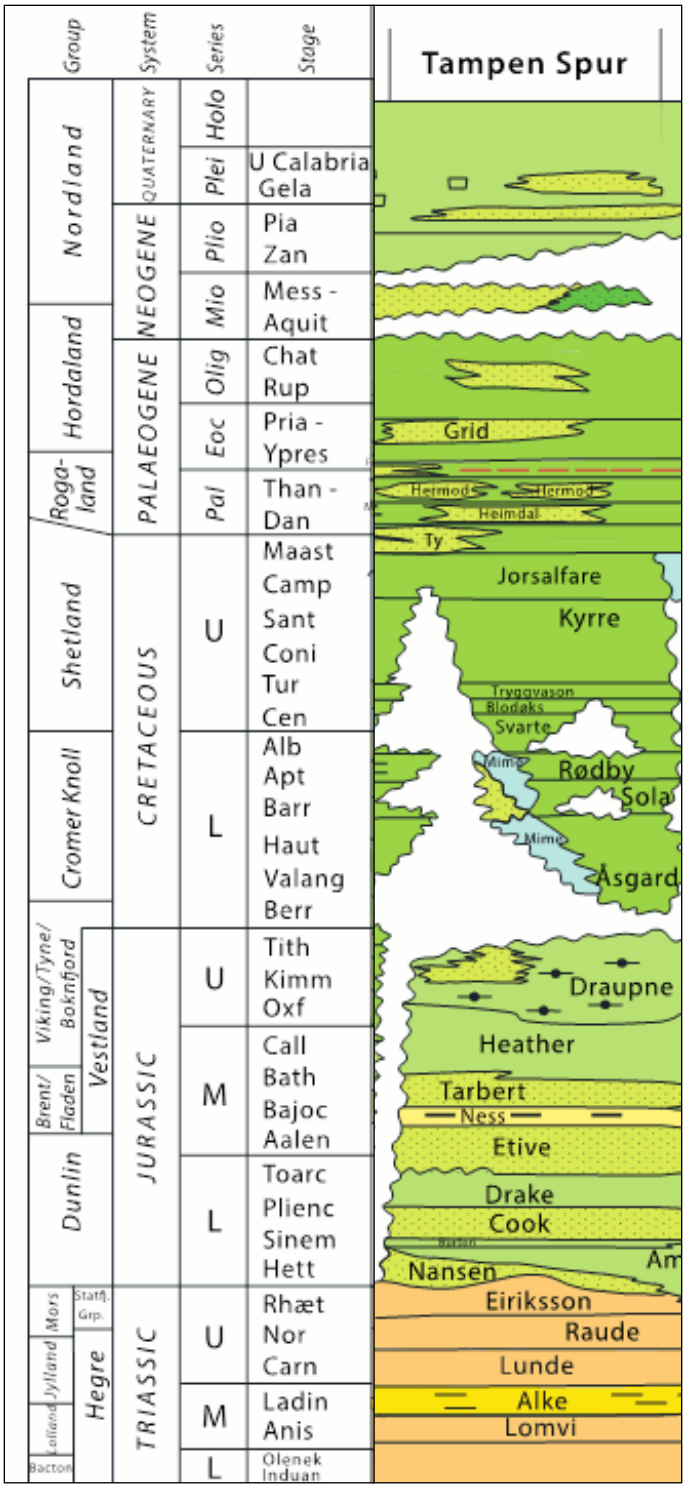

Figure 2.3: Stratigraphic column for the Tampen Spur area that contains the Statfjord Field. (Millenium Atlas, 2003).

2.2.1 Statfjord Formation

Statfjord Formation was deposited during latest Rhaetian to latest Sinemurian period and conformably overlies the Hegre Group (Figure 2.3). It is composed of interlayered sandstone/siltstone and shale with the total thickness of the formation ranging from 150 to 300m on the Statfjord Field. Towards the NNE, a thinning trend can be observed on the Statfjord Field as well as on a regional scale in the Tampen area (Hesthammer et al., 1999). In the northern most parts of the field, some localized thickening trend could be seen on the isochore maps of the formation. This indicates a change in depositional environment possibly related to the contemporaneous down throw of the hanging wall of the Alwyn-Ninian-Hutton fault zone in the southwest (Johnson & Eyssautier, 1987; Richards et al., 1993). No movement along the Statfjord Field boundary fault is exhibited during the deposition of the Statfjord Formation (Hesthammer et al., 1999; Gibbons et al., 2003).

The paleoenvironment of the Statfjord Formation can be described as alluvial plain deposits dissected by northward flowing axial rivers with local lateral fans along the Viking Graben margins (Gibbons et al., 2003). Statfjord Formation along with the Hegre Group forms the thick continental basin-fill rift and post-rift sequences found within the graben. Base of the Statfjord Formation is identified by a coarsening-upwards stratum which marks the change from Lunde Formation's (Hegre Group) shaly fluvial and fluviolacustrine deposits to Statfjord Formation's massive deposits of the alluvial plain/braided stream environment. This represents a regional basinward shift of the facies (Gibbons et al., 2003).

The Statfjord Formation is divided into three members: the Raude, Eiriksson and Nansen members. The Raude Member, which overlies the Lunde Formation of the Hegre Group, is mainly composed of fluvial channel sandstones embedded in a mudstone matrix. It is underlain by the Eiriksson Member comprising of more amalgamated fluvial channel sandstones and mudstones. The thin, transgressive sandstones of the Nansen Member was deposited as a result of marine transgression due to subsidence and a regional rise in sea-level. This sandstone unit has good reservoir properties (Gibbons et al., 2003).

2.2.2 Dunlin Group

The shallow marine mudstone, siltstone and sandstone deposits of the Dunlin Group range from the latest Sinemurian to earliest Bajocian age. The lower part of the Dunlin Group corresponds to the basin subsidence and a rise in sea-level initiated during deposition of the Nansen Member (Statfjord Formation). The transition from sandstones of the upper Statfjord Formation to silty mudstones of the lower Dunlin Group can be clearly seen on the gamma ray log. The log character for the Dunlin Group is more regular as compared to the overlying Brent Group and underlying Statfjord Group (Gibbons et al., 2003).

Four formations that make up the Dunlin group are the Amundsen, Burton, Cook and Drake Formations. All four formations can be characterized as heterolithic with alternating beds of sandstone and mudstone. The Amundsen and Burton Formations consist of shallow marine siltstones and mudstones. The Burton Formation comprises mostly of mudstones from offshore open marine environment and thus tends to be shalier. The transition to the Amundsen Formation's marine shales from the Nansen Member's marine sandstones is represented by the presence of calcareous sandstone near the base of the Amundsen Formation in the southwestern part of the Statfjord Field. The mudstones and sandstones of the Cook Formation overlie the Burton Formation. Deposition of the Cook Formation can be corresponded with a regional regression and a rapid sea-level fall during the late Pleinsbachian-early Toarcian, verified by sequence stratigraphic studies (Parkinson and Hines 1995; Dreyer and Wiig, 1995). Two large scale coarsening upwards sequences can be identified in the formation that are composed of heterolithic mudstones and sandstones deposited in wave-influenced lower shoreface and offshore environments (Gibbons et al., 2003). In the east of the Gullfaks area, deposits of a more tide-dominated deltaic environment can be found in the upper part of the Cook Formation (Dreyer and Wiig, 1995). Pre-rift doming and tectonic uplift along the eastern flank of the Viking Graben can be linked to the regional regression. Sea-level began to rise leading to a regional transgression towards the end of the early Toarcian, and this resulted in deposition of the Drake Formation's marine mudstones (Gibbons et al., 2003).

2.2.3 Brent Group

180 – 250m thick (Statfjord Field) Brent Group is made up of regressive and transgressive delta system's deposits consisting of sandstone, siltstone, shale, and coal. It was deposited during lower Bajocian to middle Bathonian ages. It is the principal reservoir of the Statfjord Field and has been divided into five formations, the Broom, Rannoch, Etive, Ness and Tarbert formations (Hesthammer et al., 1999).

The three oldest formations of the Lower Brent are deposits of coastal to shallow marine environment and can be interpreted as progradation deposits of the deltaic complex. The Broom Formation is part of a shallow marine platform that comprises of storm deposits and small distal bar build-ups. The depositional environment changes to storm wave dominated pro-delta, delta front and ebb-tidal for the Rannoch Formation, and then to tidal inlet/ebb-tidal, upper shoreface foreshore and lagoon barrier for the Etive Formation (Gibbons et al., 2003).

Upper Brent's upper two formations exhibit maximum progradation of the delta followed by onset of regression. This is represented by fluvio-deltaic deposits of the Ness formation and overlying Tarbert Formation's shallow marine deposits. The inter-fingering sandstones with shales of the Tarbert Formation in the southern Statfjord Field are determined to be fluvio-deltaic; however, in the northern part of the field that sandstones are interpreted as middle to lower shoreface (Johannessen et al., 1995). This makes transgressional nature of the Tarbert Formation highly debatable (Gibbons et al., 2003).

Chapter 3: Theoretical Background

3.1 Seismic Attribute Analysis

Since their introduction in the 1970s, seismic attributes have advanced considerably and are now an essential analytical tool for seismic interpretation (Taner, 2001). A seismic attribute measures qualitative and quantitative information of seismically driven subsurface parameters, such as frequency, velocity, amplitude, and their rate of change in terms of time and space. They are abundantly used for reservoir characterization to improve hydrocarbon exploration and development by reducing uncertainties and risks. Some of the oldest attributes include bright spot analysis that led to gas discoveries, along with some failures. As an improvement, AVO and seismic inversion were introduced combined with colour display (Taner, 2001; Chopra and Marfurt, 2005). Further development in the 1990s led to the introduction of coherence technology, spectral decomposition, and neural network applications, which further enhanced pattern recognition and visualization. With continuous advancements in computing power, geoscientists are able to combine different attributes and perform multi-attribute analysis to better understand the subsurface data (Chopra and Marfurt, 2005).

With the growing variety and number of attributes, many authors have tried to classify them into different groups depending on their computation and/or applications. For example, Taner et al. (1994) broadly categorizes attributes into physical and geometrical. While geometrical attributes enhance visualisation of the geometrical characteristics of the seismic data such as dip, azimuth, and continuity; physical attributes are associated with lithology of the subsurface and therefore, frequency, amplitude, and phase. These categories can be further divided into prestack and poststack attributes. A more recent classification by Liner et al. (2004) consists of specific and general categories. Attributes in the general category measure seismic features such as dynamic, kinematic, geometric, or statistical. These can be reflector dip and azimuth, reflector time and reflector amplitude, complex amplitude and frequency, edge detection/coherence, generalized Hilbert attributes, AVO, and spectral decomposition. These attributes are related to

general lithology or geology and can be applied to different basins in different locations. Whereas, specific attributes are exclusive to each basin in question. They are correlated to a geological feature or reservoir property of a basin. Chopra and Marfurt (2005) suggest a third category to Liner et al.'s classification that is 'composite' attributes. These can be of two types: those that display more than one attribute at a time and those combined with the help of geostatistical methods, and ML algorithms such as neural networks (Chopra and Marfurt, 2005).

3.1.1 Attributes to Identify Direct Hydrocarbon Indicators

Some of the most commonly used attributes used to detect DHI features are instantaneous, AVO and inversion attributes. Usually, DHI characteristics are associated with anomalous seismic data within the reservoir interval. Seismic attributes help compare these anomalous events to other features such as background trends, models, similar events, and geologic features (Roden and Chen, 2017).

Instantaneous Attributes:

Instantaneous attributes are computed sample-wise, and display variations of different parameters. These can be determined from complex traces. Some of the examples of instantaneous attributes are trace envelope, instantaneous phase and instantaneous frequency (Taner, 2001).

AVO Attributes:

The two fundamental attributes of AVO are intercept and gradient. After calculating variations in amplitudes with offset from the common-midpoint (CMP) gathers in each offset cubes, intercept (I) and the gradient (G) is computed in a cross-plot of amplitude versus $sin^2\theta$, with the help of linear regression. I is the cut-off on the amplitude axis: R_0 and G is the slope of the regression line. By cross-plotting, the data is converted to the 'amplitude versus angle-of-incidence' (AVA) domain from the offset domain. This is achieved by Snell's law application at the interfaces and applying interval velocities from the smoothed normal moveout (NMO) velocities.

For AVO attribute analysis, it is common to calculate the product of intercept and gradient, I*G attribute. The results are often displayed in 'product stack' sections so that the behaviour can be inspected. Another attribute that is computed is the fluid factor (FF), which can be calculated in several ways, such as the I – G cross-plot method and the V_p-V_s cross-plot method. The generated cubes from these attributes are closely inspected for anomalies that indicate the presence of hydrocarbon in the reservoir (Veeken and Rauch-Davies, 2006).

Reservoirs can be AVO classified based on the amplitude characteristics of the top reflection as a function of offset (Rutherford and Williams, 1989; Castagna and Swan, 1998). The widely used classification consists of 4 classes:

Class 1: Large positive R_0 amplitude that remains positive (dimming of reflection on stack).

Class 2: Small positive R_0 that converts to negative reflections with offset (polarity reversal and dimming or brightening of reflection on stack).

Class 3: Negative R_0 amplitude that becomes more negative (brightening of reflection on stack).

Class 4: Negative amplitude becomes less negative with offset (Veeken and Rauch-Davies, 2006).

3.2 Machine Learning

Machine learning is a branch of artificial intelligence (AI) that uses statistical computational methods and experience to make accurate predictions or improve performance. It consists of designing accurate and efficient algorithms that can provide more insight to the data and help with decision making. A learning problem is a problem of improving performance of a program through some form of data training (Jordan and Mitchell, 2015). Training the algorithm is done using a set of samples extracted from the dataset, known as 'training data'. Typically, a supervised machine learning algorithm learns through three processes: a decision process, an error function, and an optimization

process. The unique and important quality of machine learning algorithms is that they update independent of human intervention, and with each update the analytical accuracy improves (UC Berkeley, 2021). Practical applications of ML include classifications such as document classification, natural language processing (NLP), speech processing or recognition, fraud detection, learning to play games, computational biology and so on (Mohri at al., 2018). ML is now being applied actively in a range of industries, especially those concerned with data-intensive issues, to optimize business operations (Jordan and Mitchell, 2015; UC Berkeley, 2021).

Machine learning methods are usually classified into these categories:

Supervised Learning: the learning algorithm uses pre-labelled data to train and predict the outcome. With this type of learning, performance can be assessed for how accurate it is as the intended output is provided. Common methods of supervised machine learning are linear regression, support vector machine (SVM), logistic regression, naïve bayes and random forest (IBM Cloud Education, 2020).

Unsupervised Learning: for unsupervised learning, humans do not need to supervise the model. The algorithm works on its own to analyse clusters and patterns to find hidden information from unlabelled data. It allows for processing of more complicated tasks as compared to supervised learning, but there is greater unpredictability in the outcome. Nonetheless, unsupervised learning can benefit the users by finding all kinds of unknown patterns and features that can be useful for categorization. It can also be used for dimensionality reduction by reducing the number of features; singular value decomposition (SVD) and principal component analysis (PCA) are an example of this. Some methods of unsupervised learning are neural networks, k-mean clustering and self-organizing maps (SOM) (IBM Cloud Education, 2020).

Semi-supervised Learning: as the name suggests, it is partially supervised and partially unsupervised. During the training process, it uses a smaller, labelled dataset to perform classification and feature extraction from a larger, unstructured dataset. This improves learning accuracy (Expert.ai, 2020).

Reinforcement Learning: this learning model is similar to supervised or semi-supervised learning, but instead of using labels, the algorithm learns through trial and error. During the training stage, reward signals are assigned for correct sequences, and errors for incorrect ones. This way the program maximises its performance by determining the best behaviour (Jordan and Mitchell, 2015).

3.2.1 Self-Organizing Maps

Self-organizing maps (SOM), or sometimes also known as Kohonen maps, are a type of artificial neural networks that was first introduced by Professor Teuvo Kohonen in the 1980s (Miljković, 2017). SOM networks are inspired by the operations of the brain. Observations show that many sensory impressions are mapped into the brain spatially and the neurons are organized into a two-dimensional map (Kohonen, 1984, 1995, 2012). SOM is a form of unsupervised, non-linear, competitive learning algorithm that produces low-dimensional (1-D or 2-D), visually interpretable clusters from a multi-dimensional, complex data. Therefore, it is a powerful data clustering and visualizing tool. The data points on the low-dimensional map are positioned such that they maintain original topological relations from the multi-dimensional space and show relative similarity between the points (Kiang, 2001; Miljković, 2017; Cottrell et al., 2018).

Self-organizing systems can adapt their internal functions and/or structure in response to external stimuli and circumstances. Within the system, elements are able to organize each other, that results in a more stable structure or function against external fluctuations. The process involves enhancing space-time complexity of the self-organizing system that leads to an emergence of new phenomena and positive and negative feedback loops of internal regulation. This process can be observed in several natural phenomena, such as from arrangements of nanoparticles to stars and galaxies, and in living ecosystems (Banzhaf, 2009).

A SOM consists of a single, hidden layer of neurons set along a planar grid, that is connected to an input layer containing input or codebook vectors (Figure 3.1). Each neuron in the hidden layer has *n*-components and neighbouring neurons. Number of neighbouring neurons depends on the grid geometry. Most applications use rectangular

grids, but hexagonal grids are also common. SOM operate on three common principles, which are:

1. Competitive process: discriminant function for classifying the data points is calculated for each input vector connected to the map. The neuron that matches best or with the most similarity to the input pattern vector is the best matching unit (BMU) or a winner neuron.
2. Cooperative process: the BMU finds its spatial location among the neurons in the topological neighbourhood, that can then cooperate with each other.
3. Synaptic Adaptation: through the process of weight adjustments, neurons are able to change the values of their discriminant function associated with the input vectors (Miljković, 2017).

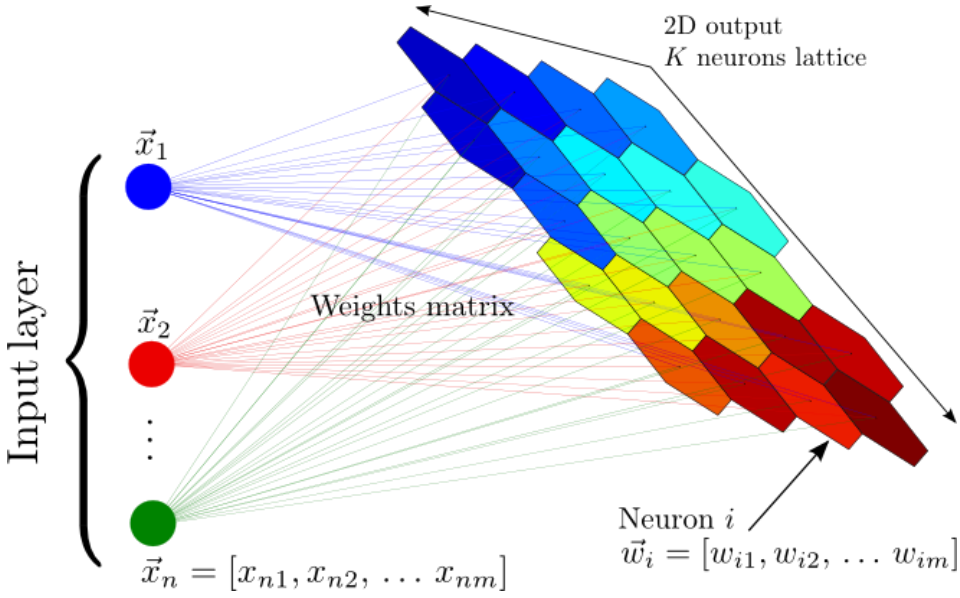

Figure 3.1: Overview of self-organizing map neural network. (Modified after Haihan Lan, 2018).

SOM algorithm:

1. Measure of distance and similarity:

Different measurement calculation methods can be used to determine the similarity between the input vector and the neurons on the map (Figure 3.1). Some of the commonly

used ones are Euclidian, direct cosine, correlation, and block distance. Squared Euclidean distance is most popular, found in most application and can be defined as:

$$dj = \sum_i (x_i - w_{ij})^2 \quad (1)$$

2. Neighbourhood functions:

Neighbourhood function is used by the neurons to interact with each other in the grid. These functions can take the form of the Mexican hat, Gaussian, cone or cylinder. The learning rate of the function can be linear, exponential, or inversely proportional, and it decreases with time.

3. Initialization:

SOM can be initialized in multiple ways before the training step. Common approaches are using random sample values from the input training data or using Principal Components that reflects the distribution of the data.

4. Training:

There are two methods to train SOM model: sequential and batch training. For sequential training, one vector at a time is presented to the map and then the neurons adjust their weights. Whereas, for batch training, all vectors are presented together before the adjustments to the neuron weights.

Steps for training are:

I. Initialization: initialising the neuron weights (iteration steps *n*=0)
II. Sampling: randomly sampling the input vectors $x(n)$.
III. Similarity matching: iterated through each neuron on the map and find the best matching unit (BMU), i, with weights $w_{BMU} = w_i$. (2)

$$c = arg_i min \parallel x(n) - w_i(n) \parallel \quad (3)$$

Where, x is the training vector from the observation and w_i is a single neuron in the matrix.

IV. Updating: each neuron is updated using the following rule:

$$w_i(n+1) = w_i(n) + \alpha(n)h(w_{BMU}(n), w_i(n), r(n)) \parallel c - w_i(n) \parallel \quad (4)$$

$w_i(n)$: the weight vector before the neuron is updated.

$w_i(n+1)$: the weight vector after the neuron is updated.

$x(n)$: the training vector from the observations.

$\alpha(n)$: the learning-factor, which can be linear, exponential or inversely proportional.

$h(w_{BMU}(n), w_i(n), r(n))$: the neighbourhood function (a smoothing kernel defined over the lattice points).

$r(n)$: neighbourhood radius (Miljković, 2017; Yuan, 2018).

V. Increment *n.* Repeat steps 2-4 until the map has reached a stable state (Figure 3.2). Stability and convergence can be confirmed when the learning-factor $\alpha(n)$ and neighbourhood radius $r(n)$ are decreasing towards zero with each iteration (Miljković, 2017).

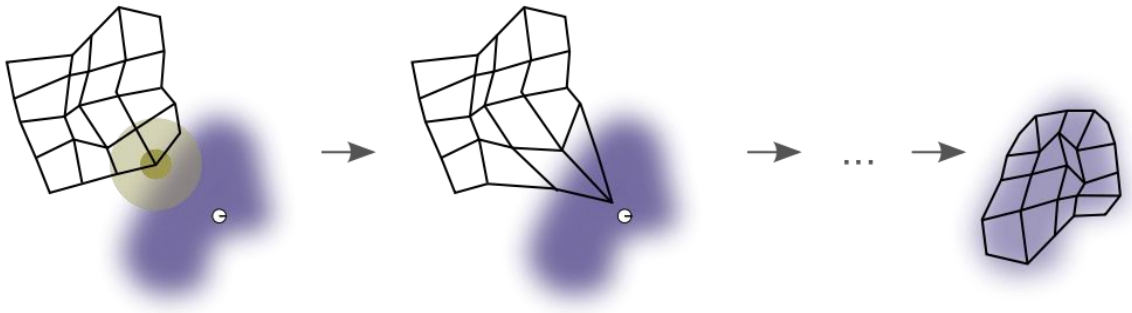

Figure 3.2: Updating the Best Matching Units (BMUs) and the neighbour radius of the data during the training process of the SOM until it reaches stability (Modified after Haihan Lan, 2018).

Chapter 4: Dataset and Methodology

4.1 Dataset

Dataset for this study was provided by Equinor ASA that consists of 3D seismic cubes and several wells from the Statfjord Field area.

4.1.1 Seismic Data

Seismic cube ST9703RZ16 is a 3D, depth-migrated seismic cube acquired by the WesternGeco in 1997 and was reprocessed in 2016. The cube covers the main Statfjord Field and the Northern Flank. Partial stacks, near, mid and far stacks, along with velocity cubes were provided. However, only partial angle stacks were utilised for this study. Some more information about the data and angle stacks is provided in Table 1.

Table 1: ST9703RZ16 seismic data summary.

Coordinate reference system	ST_ED50_UTM31N_P23031_T1133
Polarity	SEG Reverse Polarity
Near angle stack	13.5°
Mid angle stack	22.5°
Far angle stack	31.5°

4.1.2 Well Data

Several wells with their well logs were provided, but only a handful were used. Mainly, wells that were located within the cropped seismic volume and were used for assessment were: 33/9-1, 33/9-3, 33/9-4, 33/9-9, 33/12-1, 33/12-2, 33/12-4 and 33/12-5. Logs such as gamma-ray, neutron, density and resistivity were analysed for this research.

4.2 Methodology

4.2.1 Thesis Workflow

Thesis workflow is summarised in Figure 4.1. Schlumberger's Petrel software was used to carry out initial seismic and well data sorting and seismic attribute analysis. The results from Petrel were then transferred to Jupyter notebooks for the ML process using Python programming language.

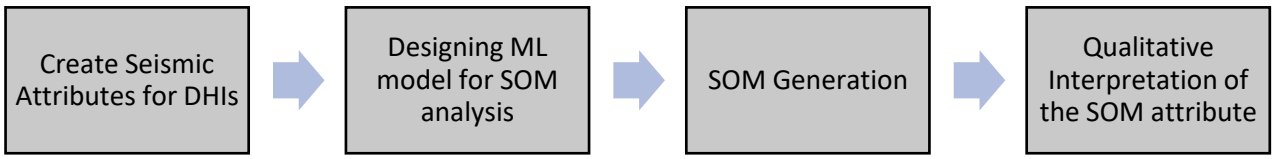

Figure 4.1: Thesis workflow.

4.2.2 Data Sorting

Purpose of data sorting was to identify main areas of interests. For this study, area of interests were the reservoir zones where hydrocarbon presence was proven or expected. With the help of the provided horizons, reservoirs for this seismic data were divided into three zones (Figure 4.2), which can be defined as:

Zone 1: reservoir interval between the BCU and the Top Cook Formation.

Zone 2: reservoir interval between the Top Cook Formation and the Top Statfjord Formation.

Zones 3: reservoir interval between the Top Statfjord Formation and to the end of the well.

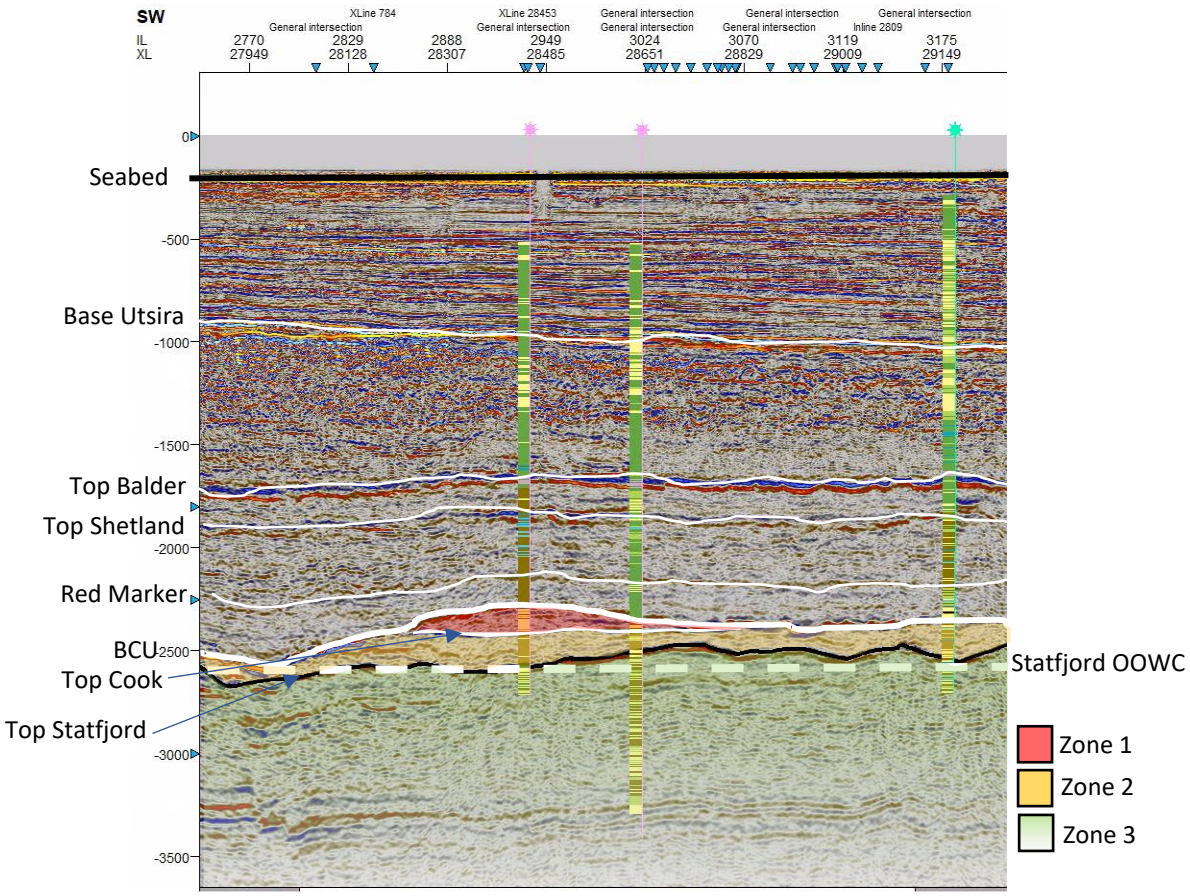

Figure 4.2: Seismic section with wells, key surfaces and zones.

4.2.3 Generating Seismic Attributes

Seismic attributes were generated for this study to detect and evaluate Direct Hydrocarbon Indicators (DHIs) with the help of Petrel software. Petrel offers a comprehensive package of attributes for seismic interpretation that can be classified as either surface or volume attributes (Sarhan, 2017). As previously discussed, common attributes for DHIs are instantaneous, AVO and inversion attributes. Volume attributes generated for this study are summarised in Table 2.

Table 2: Seismic attributes generated for analysis.

Instantaneous attributes:	
Stack:	Attribute:
Far	Instantaneous Phase
Near	Trace Envelope
AVO attributes:	
Intercept	
Gradient	
Intercept * Gradient	
Intercept + Gradient	
Intercept - Gradient	
Far - Near	
(Far – Near)*Far	

AVO attributes displayed the desired results and were proceeded further for ML. To further improve the interpretation of AVO attributes, AVO Class Cube was generated using Cegal's module in Petrel. Intercept and gradient are first calculated from the partial stacks to generate the cube. The cross-plot of intercept and gradient shows in which class the points lie (Figure 4.3). The muting function can be used to mute the background trend of the data. Only selected attributes were chosen for ML due to constraints on the computational power of the available computer. For also this reason, the attribute volumes were cropped to focus only on the reservoir zones and reduce the data size. The selected attribute volumes were then exported as SEG-Y files to be used for the ML model.

4.2.4 Data Preparation for ML Model Generation

SEG-Y files of the attribute volumes were imported in Python with the aid of Segyio Python library. Segyio is a Python library developed by Equinor ASA for handling 3D seismic data stored in a SEG-Y format. To start with, after importing the SEG-Y file for one of the attribute volumes, text file header or EBCDIC header was printed to check if the file was read correctly. EBCDIC header states information about acquisition and processing workflow of the data. It mentions information about the coordinates, number of inlines and crosslines, sampling interval, amplitude range, etc. that can be used to QC the geometry of the seismic. After sorting out the inline, crossline, trace and sample

numbers, a central seismic line from the cube was viewed using Python's Matplotlib library to ensure data was consistent (Figure 4.4). The process was then repeated for all the attribute volumes, one by one.

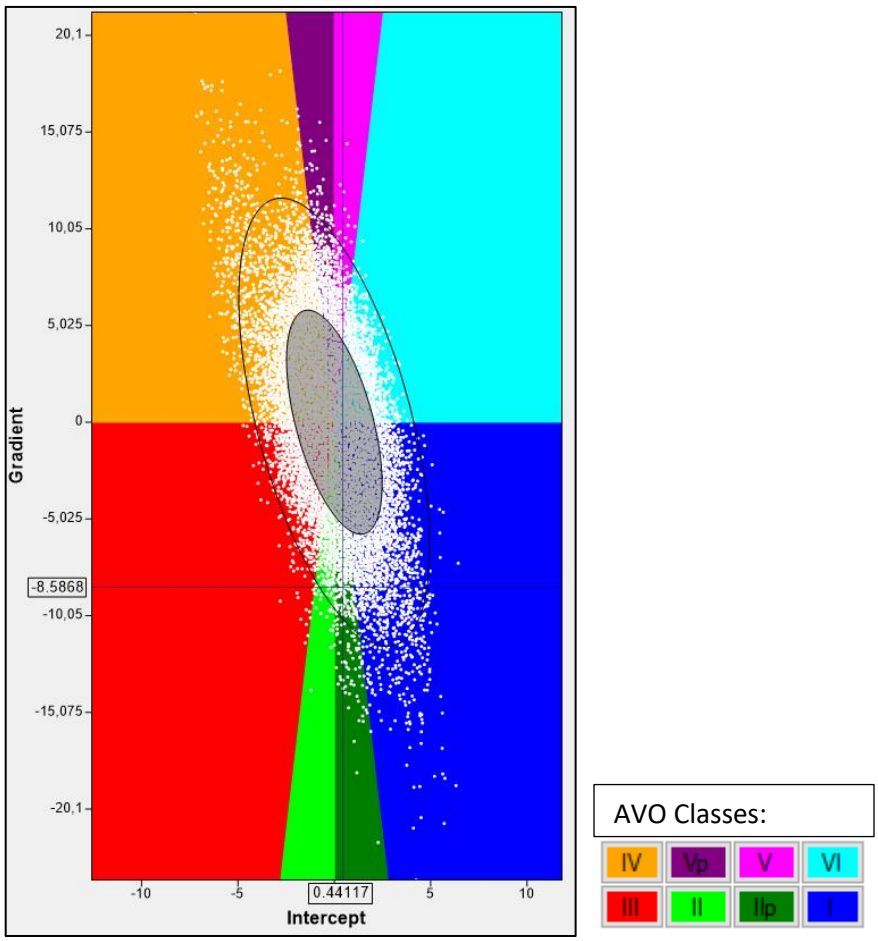

Figure 4.3: AVO classification of the seismic data.

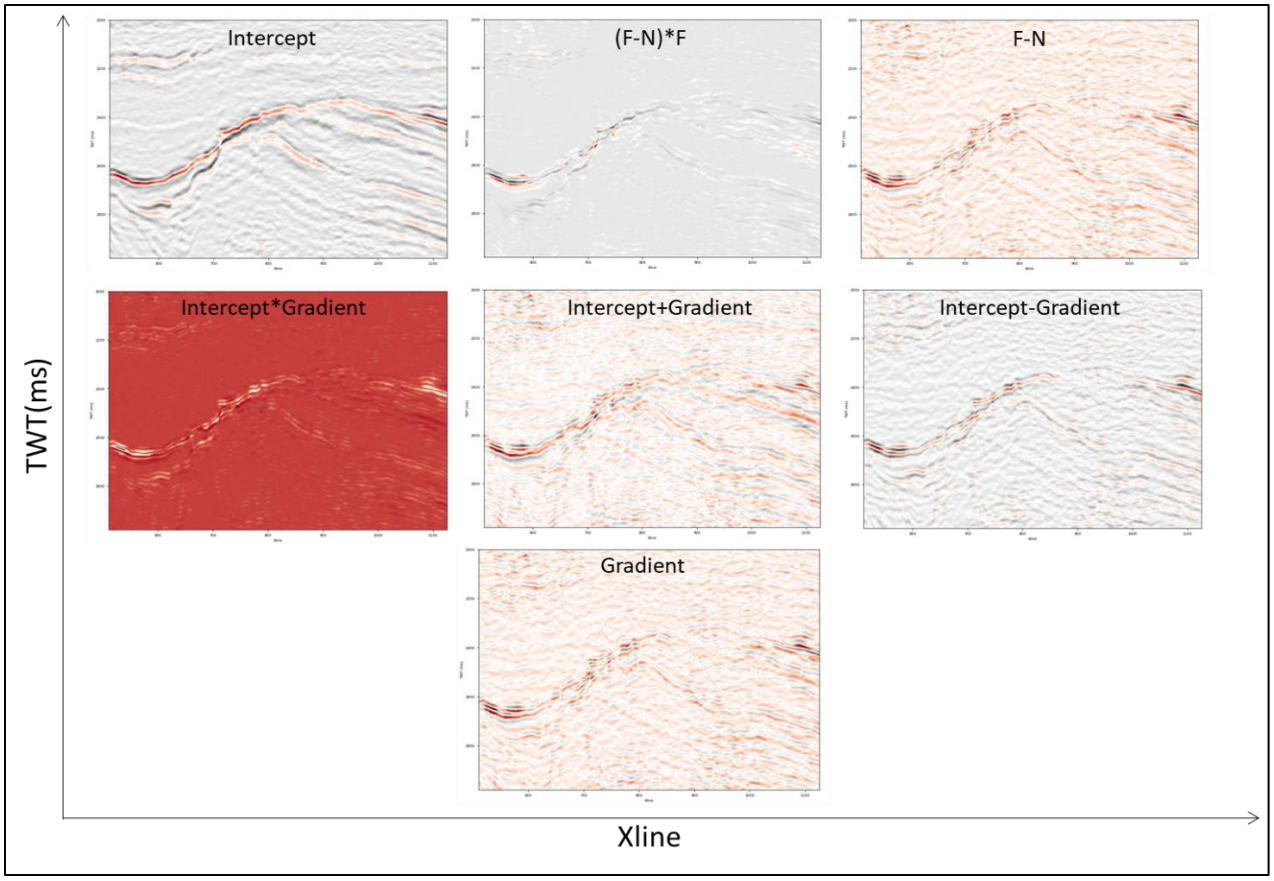

Figure 4.4: Seismic lines from each attribute imported in Python.

The seismic data was converted to a 3D array using Xarray. Xarray is a Python package that allows efficient and easy handling of multi-dimensional arrays. It stores information in the form of dimensions, coordinates and attributes in addition to raw NumPy arrays for better handling and manipulation of the data with less errors for developing purposes (Xarray Developer, 2021). Once the data for all attributes was converted to an array, it was stored into a dataset where each seismic attribute is a feature/variable (Figure 4.5). The dataset can then be converted to a matrix to be used as an input for the ML model.

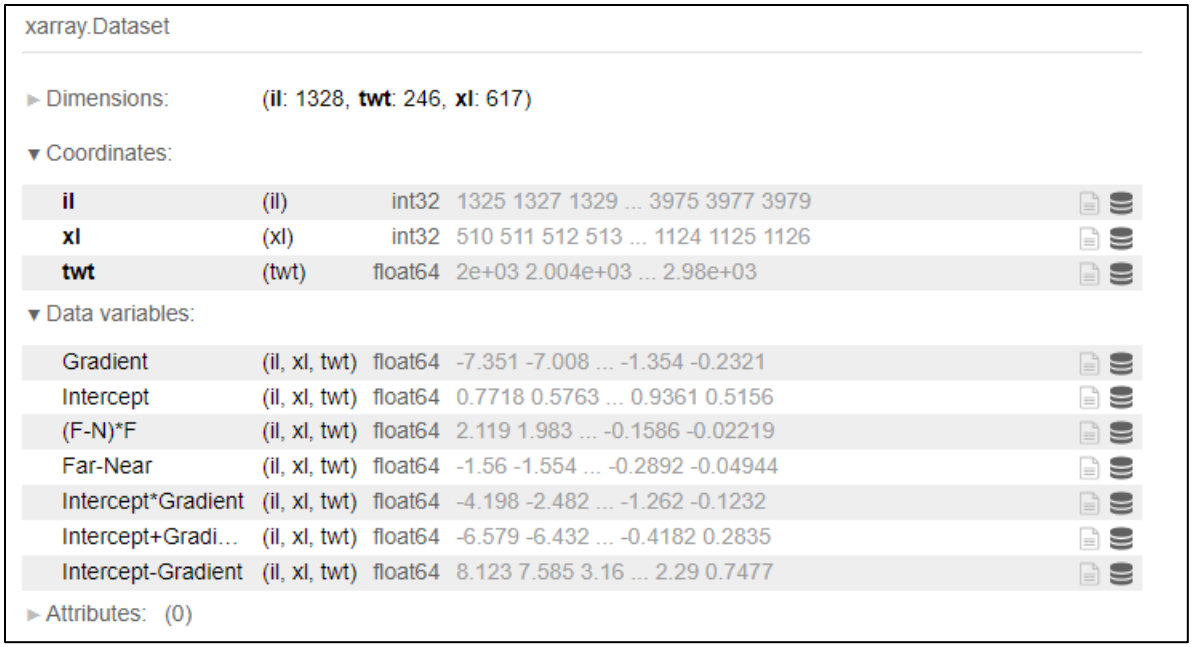

Figure 4.5: Xarray dataset with data arrays of different seismic attributes stored as variables.

4.2.5 SOM Model Generation and Training

For this study, SOMPY library for implementation and visualization of the Self-Organizing Maps in Python was used. This library is closely based on SOM Toolbox, the SOM library for MATLAB, developed by the Helsinki University of Technology. It depends on other Python packages such as NumPy, SciPy, Scikit-learn, Pandas and Matplotlib, and offers functionalities such as batch training, random and PCA initialization, rectangular and hexagonal shape for 1-D or 2-D SOM grid, BMU Hitmap and U-Matrix visualization.

Once the input data was converted to a multi-dimensional NumPy array, SOM model was constructed. Parameters for building a model and their description are summarised in Table 3:

Table 3: SOM model parameters and their brief description.

Parameter	Description
Data	Input data that is to be clustered. In a matrix format with n rows as data point and m columns as features.
Neighbourhood	Calculation of neighbourhood matrix by either Gaussian or bubble method
Normalization	Normalizing data using the variance method
Map size	Defined as dimensions of the SOM or number of nodes

Map shape	Shape of the SOM map. SOMPY only offers an option of a planar map
Lattice	Type of lattice for the SOM. Options for this are rectangular or hexagonal
Initialization	Method to select the samples from the input data to initialize the SOM. Samples can be chosen randomly or by PCA.
Component names	Names of the individual variable or attributes of the data.
Training	Either batch or sequential training algorithm to train the model.

When designing the model, it is essential to understand the function of each parameter and how it can affect the training process as well as the outcome. Finding the right parameter also requires several trials and then choosing the ones that give the best response. However, this can be a very time-consuming process, especially when working with a large dataset, such as the one used for this project, as training the model can take several hours. Therefore, the parameter for the model were selected based on theoretical knowledge (discussed in Chapter 3) to reduce the number of trials.

To efficiently train the model, batch training algorithm was selected to speed up the training as the model takes input data in one batch rather than vector-by-vector for sequential training. Data samples were chosen randomly and were standardized using the variance method, where variance is standardized to 1. Rectangular lattice with 225 neurons (map dimensions 15 x 15) displayed the best result when compared with other map sizes as it resulted in none to least amount of ‘dead neurons’, which are neurons with no BMUs.

To initiate training, the number of epochs were defined so that SOM can thoroughly iterate through all the nodes and update their weights until the map reaches a stable condition and finds BMU for all the sample points (Figure 4.6).

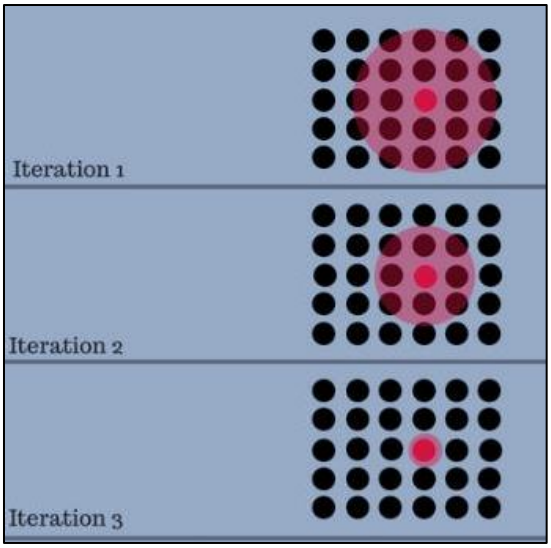

Figure 4.4: Neighbourhood radius decreasing with each iteration. (Modified after Amir Ali, 2019)

Training with SOMPY is divided into rough training and fine-tune training. Rough training is the organizing or the ordering phase during which the weight vectors are topologically ordered on the map. While, fine-tune training is the convergence phase that trains the input vectors and provides a statistical representation of the data (Miljković, 2017). Epochs for rough training was set as 2, and for fine-tune training was set as 3. Epochs for fine-tune training could have been set to a higher value to ensure model convergence, however, this training phase is computationally expensive and can take several hours to days with the given data size.

4.2.6 SOM Visualization and Clustering

SOM is versatile tool for data analysis by visualization and the results of a trained SOM can be viewed in various ways. Extracted features and their values can be projected on the grid for analysis. The aim of this project was to detect DHI characteristics in the seismic data, and these can be identified by anomalous data points.

Initially, to understand which part of the map best corresponds to the data, BMUs from the investigated samples are displayed on the map (Figure 4.7). BMU Hit map shows how the data responds to the map and how many times each neuron on the map was the BMU for the input data sample. This helps to visualise how many data values each neuron holds. The neurons with high values are representing more data points, therefore, neurons with low values could possibly be holding the anomalies. Furthermore, clustering

classifies each map unit into classes to represent the natural clusters present in the data (Figure 4.7). Natural clusters in a geological data such as seismic can represent subsurface facies distribution, seismic reflection characteristics, faults, and other geological features. Clustering in SOMPY utilises Scikit-learns cluster.KMeans method that groups data into n groups with equal variance after dimensionality reduction. This minimises a criterion known as 'inertia' or sum-of-squares within clusters (Equation 5) (Scikit-learn User Guide, 2020).

$$\sum_{i=0}^{n} \min_{\mu_j \in C} (\|x_i - \mu_j\|^2) \quad (5)$$

Where, x_i is the sample point and μ_j is the mean of the samples in the cluster.

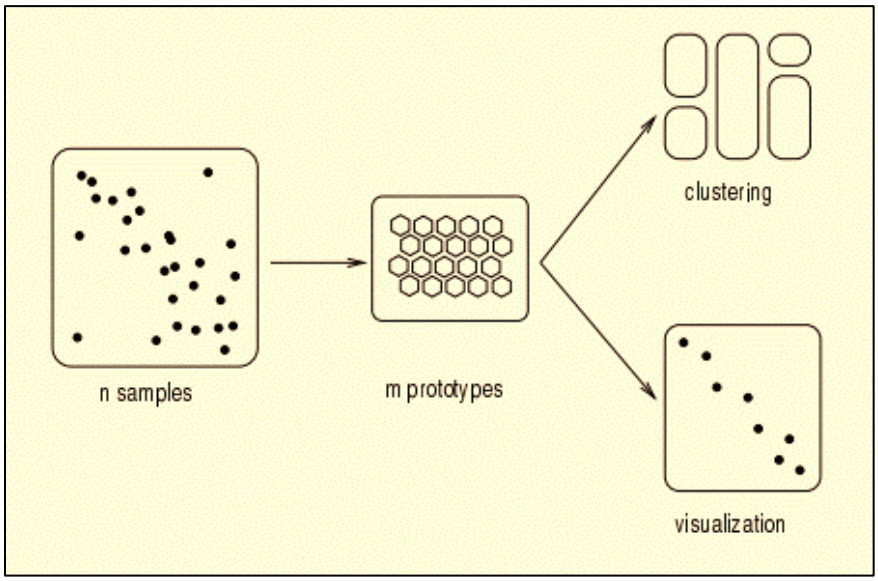

Figure 4.7: Data Analysis from SOM. (Modified after Vesanto, 2000).

Number of clusters needs to be specified and it is hard to know what the optimal number of clusters is to define the natural clustering in the data. One of the methods to overcome this is using the 'Elbow Curve'. The 'Elbow Curve' method takes into account inertia and distortion, which is the average of the squared distances from the cluster centres, and usually the Euclidean distance metric is used. The plot, such as the one in Figure 4.8, shows distortion values for each number of clusters. Cluster values beyond the 'elbow' of the curve do not contribute towards modelling the data better and may lead to 'over-

fitting'. Therefore, by identifying the 'elbow', the optimal number of clusters can be determined. After this, cluster in SOM can be evaluated and anomalies can be identified by isolated neurons that are away from their data class.

To further investigate, BMUs can be projected over the whole dataset and classified by the cluster classes. This process gives an array with same amount of values as the input data, which allows converting the array to original data geometry so the results can be viewed as a new seismic attribute cube and assessed for new information.

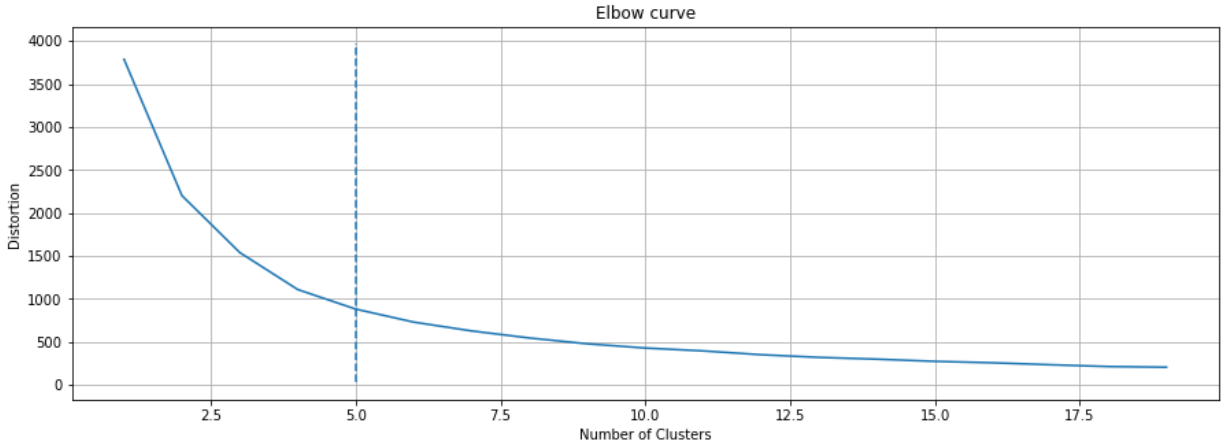

Figure 4.8: The 'elbow curve' to determine the optimal number of clusters.

4.2.7 SOM Quality Measures

It can be challenging to assess the outcome of SOM training since it is an unsupervised ML method and target values are not factored into the training. However, there some measures that can be calculated to check the accuracy of the model. The most popular and the ones evaluated in this study are Quantization error (QE) and Topographic error (TE). For a SOM model to be accurate, it has to maintain the neighbourhoods and the topology of the input data (Breard, 2017).

Quantization Error:

QE is a basic quality measure and is computed by calculating the distance between the data points and the map nodes. The error reaches towards zero with increasing iterations and/or map size. Smaller value of QE indicates a better fit; however, it is only a measure

of quality when compared with the maps of the same input data (Breard, 2017). For this study, QE values after each iteration were observed to evaluate the quality of the final map. The calculation can be defined as:

$$QE(M) = \frac{1}{n}\sum_{i=1}^{n}\|\phi(x_i) - x_i\| \quad (6)$$

Where n is the number of training data points and $\phi: D \mapsto M$ denotes mapping of the SOM M from the input space D.

Limitations of QE are that it only evaluates the local structure of the data, and not the interrelationship of the neurons (Breard, 2017).

Topographic Error:

TE measures SOM's ability to preserve topological features of the input space in a low dimensional space. It estimates the local discontinuities in the mapping by evaluating the positions of the best-matching and second best-matching neuron for each input (Breard, 2017). Topology is preserved if the neurons are besides each other, otherwise it accounts for an error. Topographic error for the map is calculated by dividing the total number of errors by the total number of data points (Equation 7).

$$TE(M) = \frac{1}{n}\sum_{i=1}^{n} t(x_i) \quad (7)$$

$$t(x) = f(x) = \begin{cases} 0, & if\ \mu(x)\ and\ \mu'(x)\ are\ neighbours \\ 1, & otherwise \end{cases}$$

Chapter 5: Results and Discussion

5.1 Results

5.1.1 Seismic Attributes

The focus of this study was on direct hydrocarbon indicators (DHIs), to understand the types of fluid present in the Statfjord Field area and their response on seismic. Both oil and gas can be found in the reservoirs of the Statfjord Field. Porosity and permeability of the reservoirs can range from 11% to 30% and 5 to 5000mD, respectively (Gibbons et al., 2003). These reservoirs can be classified into AVO classes by studying the effect of hydrocarbon on seismic amplitude with respect to offset. AVO classes are discussed in chapter 3, and by applying that knowledge we can observe the angle stacks to determine the AVO classes of hydrocarbon accumulations. In Figure 5.1, an inline from near, mid and far angle stacks located in the south-west of the study area is compared. The circled area corresponds to the top of Etive Formation of the Brent Group that is oil-bearing, as recorded in the well 33/9-1. A brightening effect, or a 'bright spot' which is a common DHI, is observed for the negative (red) reflector in the far and mid angle stacks when compared to the near angle stack. This behaviour corresponds to AVO Class 3 that states negative amplitude becomes more negative.

A quick and minimal AVO Class Cube was generated after calculating the intercept and gradient parameters from the partial stacks. The cube confirms that the anomaly is of Class 2 negative and Class 3 (Figure 5.2a). The whole extent of the anomaly can be seen on the time slice in Figure 5.3a. Figure 5.3b shows the intercept and gradient cross-plot where AVO Class 3 has a negative intercept and gradient. The grey-coloured ellipse indicates muting of the background trend and the outer ellipse is the strength scale, which is twice the radius of the mute ellipse. Gamma-ray log from well 33/9-1 overlying the anomaly on crossline (Figure 5.2b) indicates that the interval corresponds to good quality sands. Class 3 reservoirs are often porous, loose sands that have lower acoustic impedance than the overlying shales. Whereas, Class 2 reservoirs are moderately

consolidated with the same acoustic impedance as the overlying shales. It is possible that the shalier sands observed on the gamma-ray log corresponds with Class 2.

Multiple attributes were generated from the combination of intercept, gradient and angle stacks, with the assumption that the reservoir is of AVO Class 3. The effect of these attributes on the anomaly mentioned above is shown in Figure 5.4. As mentioned before, intercept and gradient are negative for Class 3, therefore, their product (I*G) is positive. Similarly, the result of (F-N)*F is also positive and clearly highlights the AVO effect. Results for I+G, I-G and F-N were all negative, indicated by the red colour of the reflector. These attributes may not emphasize Class 2 anomalies in this particular inline but can be good indicators for Class 2 anomalies, that may be present elsewhere in the field.

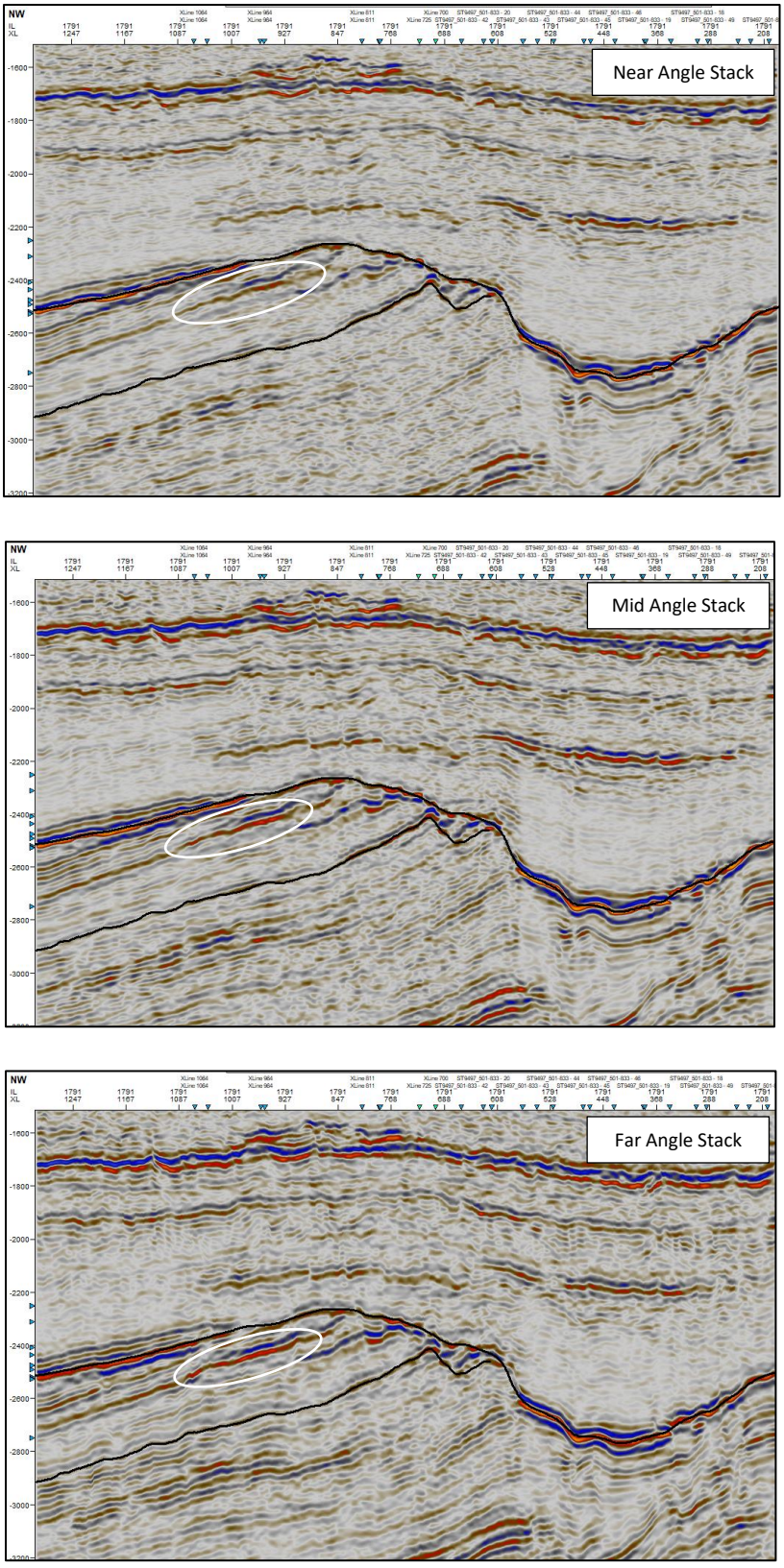

Figure 5.1: Inline 1791 from angle stacks (from top to bottom: near, mid and far) with AVO anomaly highlighted. The top interpreted horizon is BCU and the bottom is Top of Statfjord Formation.

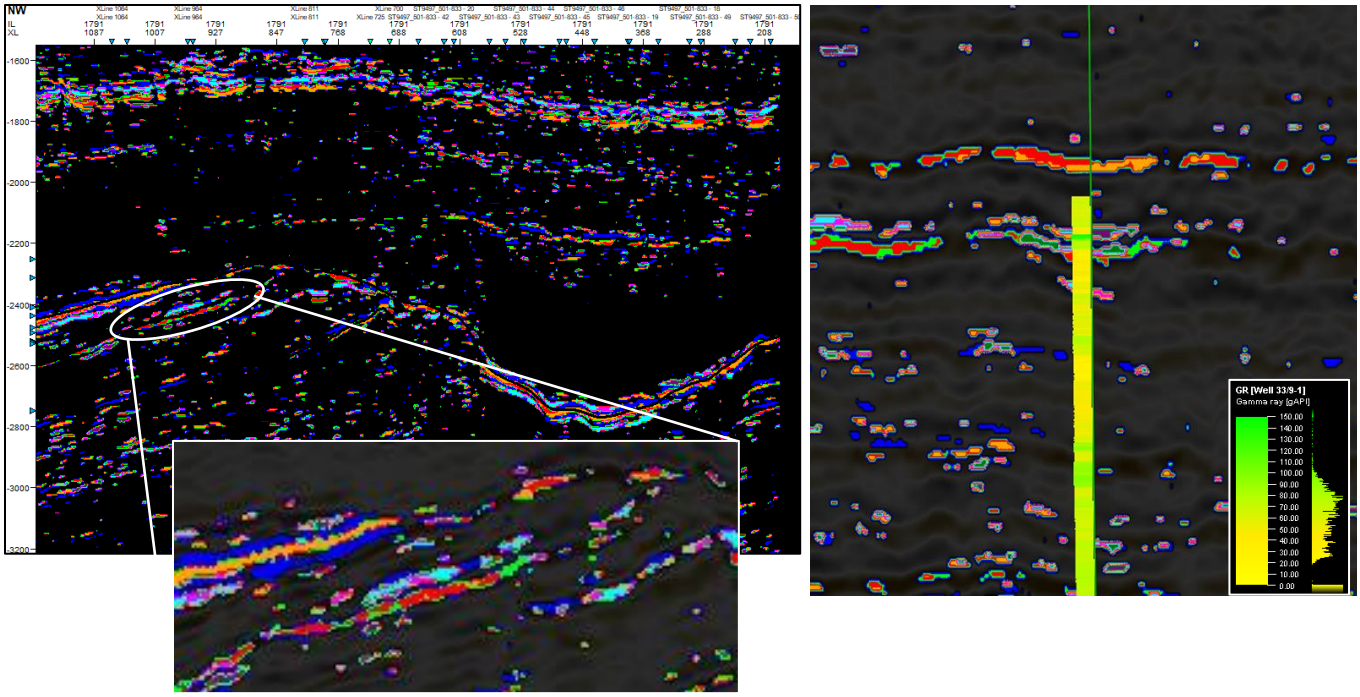

Figure 5.2: a) Inline 1791 from AVO cube. b) Crossline 964 with gamma-ray log from well 33/9-1 overlaid.

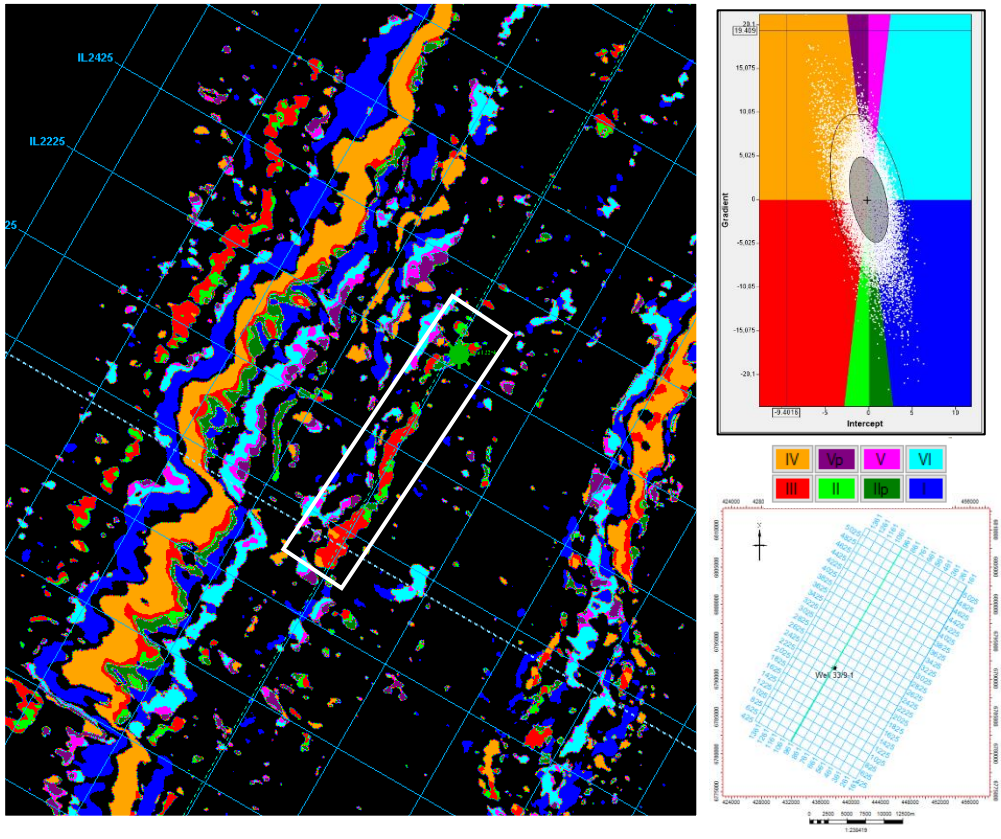

Figure 5.3: a) Time slice -2436ms from the AVO class cube showing the extent of the AVO anomaly. b) Intercept and gradient cross-plot with AVO classification as background. C) Map of seismic survey with crossline 964 and well 33/9-1.

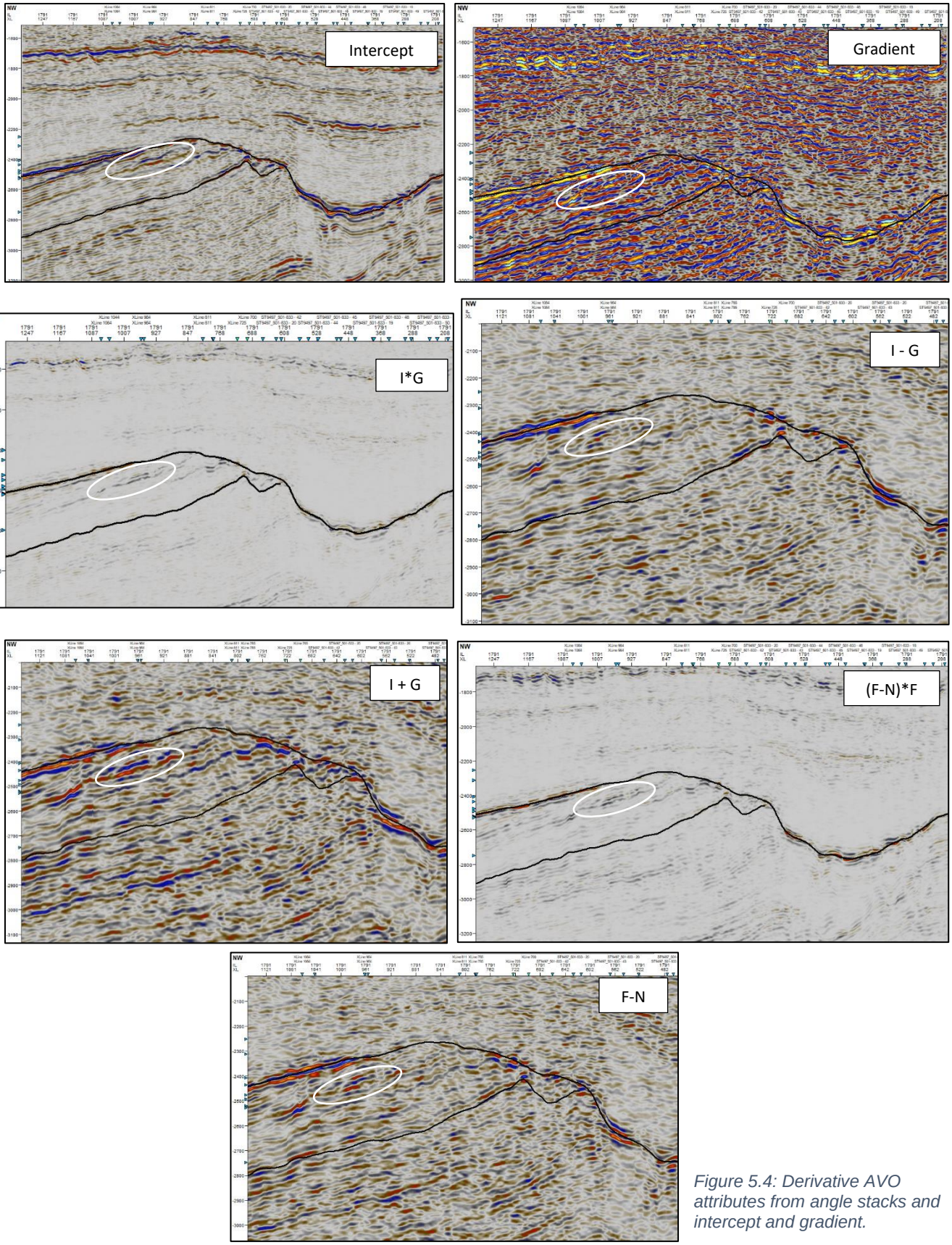

Figure 5.4: Derivative AVO attributes from angle stacks and intercept and gradient.

5.1.2 SOM Analysis

The AVO attributes identified the presence of hydrocarbon and their effect on seismic amplitudes and helped understand the reservoir characteristics. This information can be used to identify new accumulations and reduce risk and failures of future prospects in the field. However, analysing each attribute individually can be time-consuming and inefficient. The attributes can instead be combined to further enhance the results and highlight new, discrete anomalies and features that were previously missed or were not visible. This can be achieved by multi-attribute analysis using SOM as it takes into account variation in the data to provide new information or build confidence in previous interpretations.

Figure 5.5 shows the resultant BMU Hit map after training the SOM model. It consists of 225 neurons on a 15 by 15 lattice. The neurons are annotated by the number of data points that particular neuron is representing. For example, green neurons in the top, right corner and bottom, left corner have values 2977114 and 4305134, respectively. These values are the highest number of data any BMU is representing. Whereas, the dark red cells have comparatively lower values, that indicates fewer data points representation. The model was trained with 2 rough training epochs and 3 fine-tune training epochs. The final Quantization error (QE) was 0.449, that decreased from the initial error of 10.378, and the final Topographic error (TE) was 0.287.

Clustering was done to classify data by their characteristics (Figure 5.6). Number of clusters were determined using the ‘elbow curve’ (discussed in Chapter 4) that suggested the optimal number of clusters is 5 (Figure 4.8). Different clusters can represent different geological feature in the seismic data. Since DHIs are anomalous features, they were expected to belong to small data clusters. Therefore, cluster ‘4’ in the top write corner and the isolated cluster ‘3’ on the left were of particular interest.

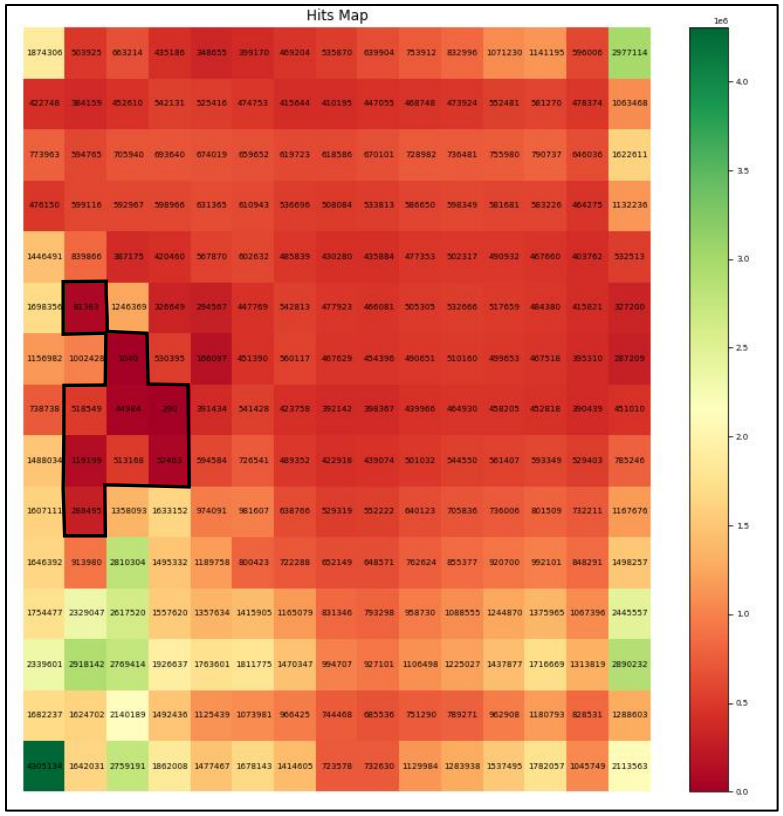

Figure 5.5: BMU Hit map of rectangular lattice and 225 neurons. Low value neurons are outlined in black.

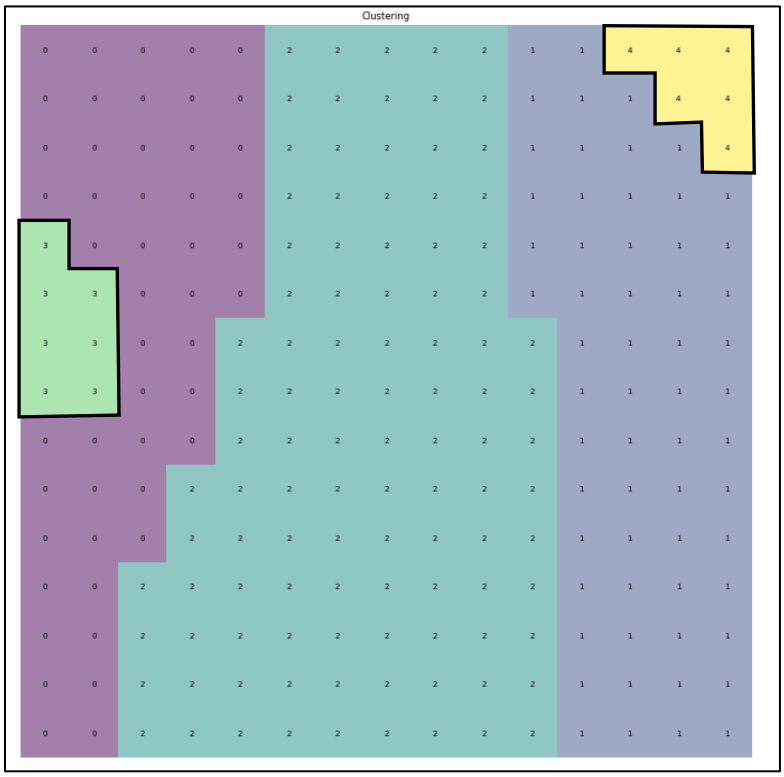

Figure 5.6: Result of data clustering with cluster number = 5. Clusters of interest are highlighted in black.

To visualise the results of the SOM classifications, the BMUs from the prototype vectors were project over the whole data. Dataset with assigned BMU was then viewed as seismic. Similarly, the BMU assigned data values were clustered using the cluster prototype vector and transformed into seismic. Results of the new ‘SOM attributes’ are presented in figures 5.7 and 5.8. Though the figures are of slightly lower resolution, both classifications show the geological structure and dipping of the horizons. The reflectors are continuous and separate horizons can be interpreted, even more easily with well control. Strong reflectors, such as the BCU and the Top of Statfjord Formation, are clearly evident in the BMU classification cube by the dark-green coloured reflectors, just like in the original seismic. They are a little difficult to delineate in cluster classification cube, but still interpretable by continuous reflectors, such as those in green and brown colour. However, it is difficult to interpret the DHI previously identified in the AVO attribute cubes. White circle on the figures infers the position of the anomaly. Apart from slight dimming of the reflector on the BMU classification cube, no distinct irregularities were found. On the plot of the time-slice that corresponds with the time slice in Figure 5.3a, the anomaly approximately lies within the white box in Figure 5.9. Therefore, it can be speculated that by SOM classification, the Class 2/3 anomaly is possibly the cluster in brown colour, which is clusters ‘3’.

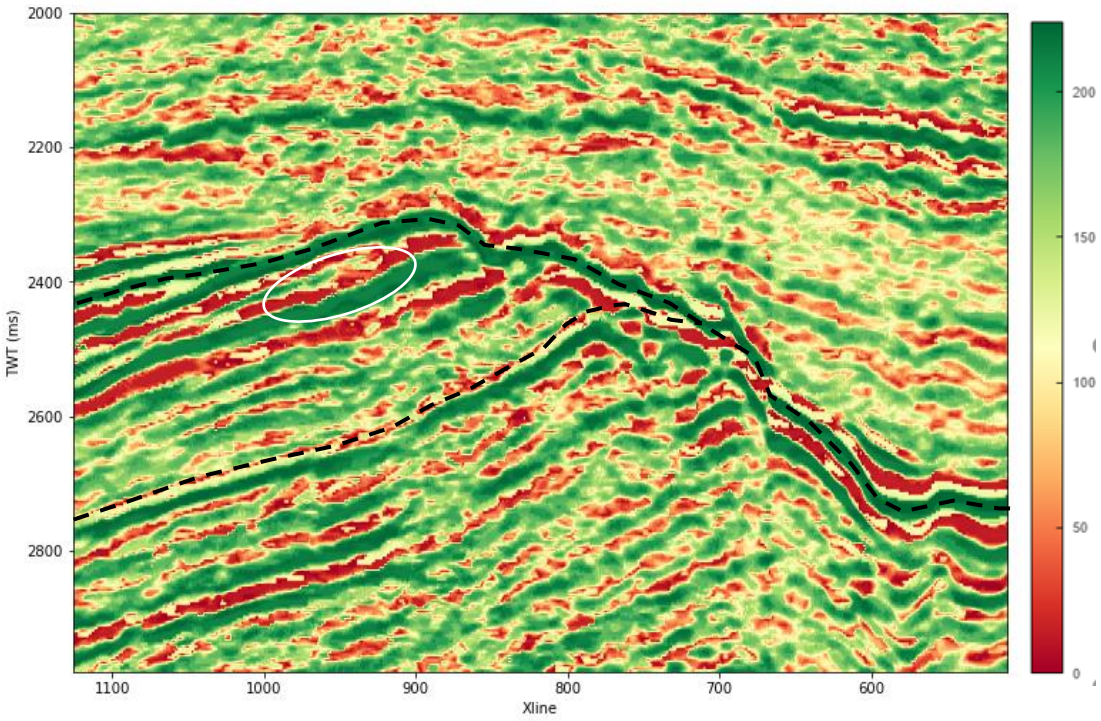

Figure 5.7: Seismic inline classified by BMUs with BCU and Top Statfjord Fm interpretations in black, dashed line.

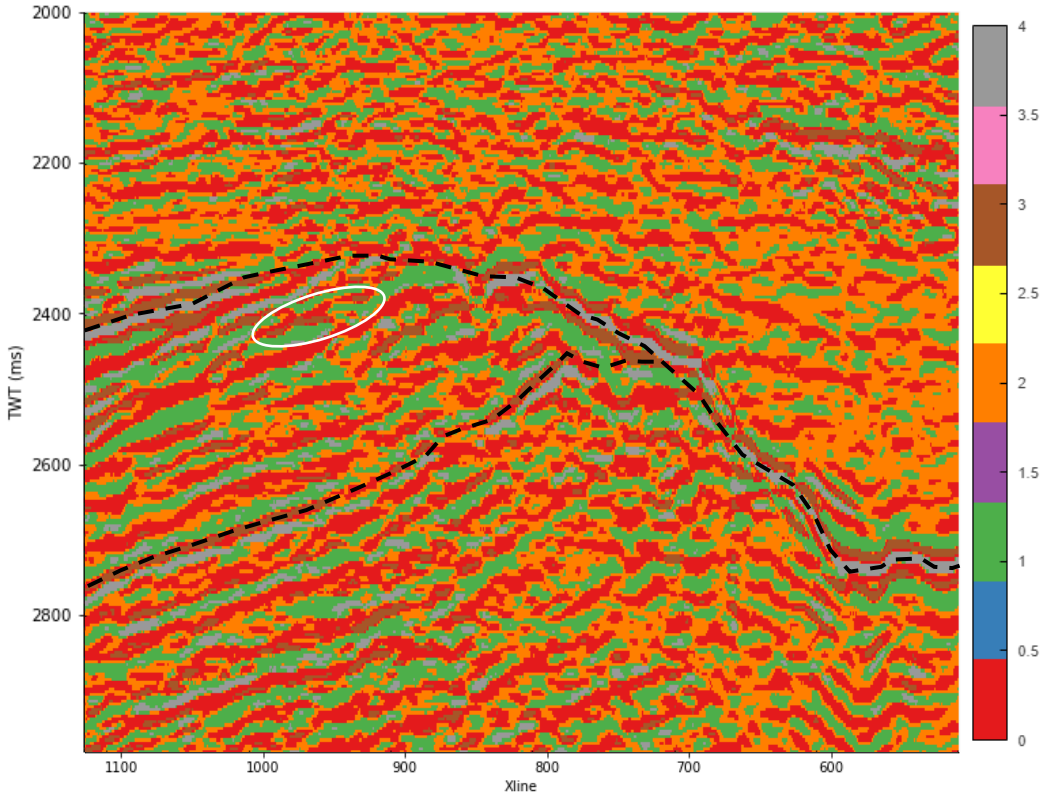

Figure 5.8: Seismic inline classified by SOM clustering with BCU and Top Statfjord Fm interpretations in black, dashed line.

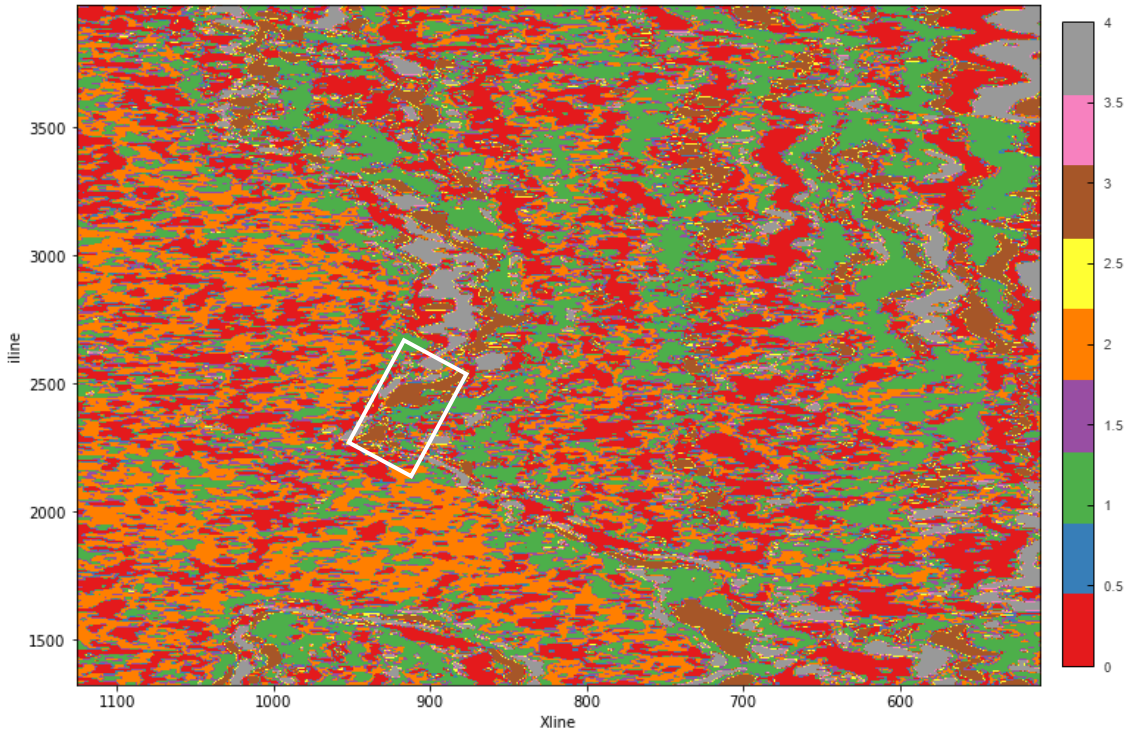

Figure 5.9: SOM cluster classification time-slice corresponding to time slice -2436ms in the original seismic.

5.2 Discussion

The calculated seismic attributes provided good insight to detect hydrocarbon indicators, such as ‘bright spots’ and ‘flat spots’, to characterize the reservoir. The accumulation marked in Figure 5.3a lies in the Etive Formation of the Brent group, which is a highly prolific reservoir in the field. The Derivative attributes such as I*G and (F-N)*F highlighted the AVO Class 3 behaviour well, while other attributes did not. Class 3 response can be correlated to the clean, porous sands that were observed on the gamma-ray log of the 33/9-1 well. Although Class 2 anomalies are present in the Statfjord Field area and can also be seen on the AVO Class Cube, they were hard to detect from these attributes. None the less, the attributes were successful in identifying DHIs, which was the aim of generating these seismic attributes, so they can be used as input for the SOM analysis.

The resultant Hit map of the SOM training shown in Figure 5.5, displays the spread of data across the SOM map. The Hit map had no neurons with 0 hits, which implies that all the sampled data points were represented by a BMU and that different dimensions of the

data are connected. However, with large and non-linear sizes it should be expected that not all data points will have a BMU as dimensionality reduction cannot be connected everywhere. These characteristics of the SOM map express the natural clustering in the data. The natural clusters can also be viewed on the cluster map (Figure 5.6), assuming that the training was completed successfully, and the correct number of clusters was selected. The data are grouped into clusters based on the similarity of their characteristics. That is why it is safe to assume that DHI anomalies could be found in small clusters since they have distinct characters, making them stand out from the rest of the data. Considering this, it is possible that the DHIs were not highlighted in the output results because the cluster map does not represent the data well and a higher cluster number that does not separate the anomalies.

SOMPY Python package is well built and offers sound functionalities to perform SOM analysis. It offers good amount of freedom to users to freely design their SOM model based on their objectives. However, for this study, there were some limitations that possibly led to not achieving desired results. To start with, SOMPY only offered standardization of the data samples by the variance method, although this is not the preferable method for non-linear data. A method such as log-normal normalization would have been more suitable for non-linear seismic data. For re-modelling and re-training purposes, there is no ‘seed’ function offered by the program to ensure that the same data samples are used each time model is re-trained. This function would help understand how each parameter affects the outcome and make it easier to reproduce the model that was considered to be the best one. Different models were compared based on the TE, which should be as low as possible. The value of 0.287 was deemed to be good as compared to other models that had higher values. However, the calculation of TE is computationally expensive, especially for large datasets, such as the one used in this study. QE, on the other hand, is a standalone assessment and helped determine the number of iterations to run to improve the results. Figure 5.10 shows how QE value decreased with each iteration. Models trained with minimum 3 and maximum 5 fine-tune training iterations yielded QE values between 0.41 and 0.45. Since the values did not reach a plateau, it is assumed that the model did not converge and that there is still room to reduce the error further. Even though, running more iterations and/or increasing the

map size improves the quality of the map, it was extremely time-consuming to run iterations more than 5 and was not feasible with available computational power. This is a fundamental shortcoming that can be improved with the help of a more efficient computation solution by, for example, pipelining and/or utilizing GPU to speed up the process without overburdening the computer's memory.

```
Training...
random_initialization took: 3.635000 seconds
Rough training...
radius_ini: 5.000000 , radius_final: 1.000000, trainlen: 2

epoch: 1 ---> elapsed time:  1926.655000, quantization error: 10.378081

epoch: 2 ---> elapsed time:  1993.280000, quantization error: 1.842716

Finetune training...
radius_ini: 1.250000 , radius_final: 1.000000, trainlen: 3

epoch: 1 ---> elapsed time:  2019.363000, quantization error: 0.822095

epoch: 2 ---> elapsed time:  2177.856000, quantization error: 0.565014

epoch: 3 ---> elapsed time:  2145.540000, quantization error: 0.449909

Final quantization error: 0.449909
train took: 10312.249000 seconds
```

Figure 5.10: Resultant QE after each training epoch.

Although the outcome of the SOM classifications can be used to interpret horizons, they do not offer improved visualization of anomalies as they were expected to. Apart from poor SOM model construction, this could be due to several other reasons. It is possible that there were not enough input attributes or that they did not offer variation in data to derive new results. Utilizing instantaneous and inversion attributes could have provided more depth and insight to intensify DHI characteristics on the SOM map. Moreover, other SOM classifications, such as classification based on the distance of the sample from the winning neuron, could have been explored to find better results.

The study has a lot of room for improvement, and it is believed that with these improvements can achieve the desired results. A deeper understanding of the SOM algorithm and exploring other Python libraries available for SOM analysis can prove to be beneficial. Improving the quality of the input data by developing more variety of seismic attributes can tremendously influence the outcome of SOM analysis and offer more information.

Chapter 6: Conclusion

The purpose of this study was to conduct multi-attributes analysis using an unsupervised machine learning method called Self-Organizing Maps (SOM) to detect hydrocarbon indicators in the field. The aim was to understand how SOM can assist in developing a better understanding of the data and finding new geological information related to hydrocarbon fluid presence in the Statfjord field, that can improve the current interpretation.

For this study, however, SOM did not produce the desired results owing to a number of reasons. While some of the attributes used detected the DHI features, not all of the attributes provided distinct information to better delineate the hydrocarbons. Therefore, SOM failed to provide new observations. Moreover, better construction of the SOM model could have yielded better results. Despite the poor results of this study, it is important to recognize that SOM offers a lot of potential and is a powerful visualization tool for multi-dimensional data. This is proved in multiple studies done by geologists such as Rocky Roden, Deborah Stacey, Sharareh Manouchehri and many more, where SOM has helped define the geological features better.

The analysis of this study can be upgraded to achieve desired results by implementing instantaneous and inversion attributes, and by improving the model design. Other SOM packages based on Python's powerful machine learning libraries, such as Tensorflow and Keras, can be explored to improve the model for analysis. Furthermore, it is important to program more efficient ways to compute the model that can accommodate large datasets such as seismic data. To do this, a deeper understanding of data mining and machine learning is required.

Chapter 7: Future Work Recommendations

To address the shortcomings of this thesis and improve the results, following issues can be addressed in the future work:

- Calculate more seismic attributes of different types that relevant to the problem, such as instantaneous and inversion attributes, multi-trace and single-trace attributes, so they provide more variation in the data.
- Explore other SOM libraries and Python packages, such as Tensorflow, that offer more comprehensive calculations for model design and training.
- Design efficient pipelines and generators for large datasets to utilise GPU and reduce computer's memory consumption so that model can be trained well.

References

Ali, A. (2019). *Self Organizing Map(SOM) with Practical Implementation*. Retrieved from Medium: https://medium.com/machine-learning-researcher/self-organizing-map-som-c296561e2117

Banzhaf, W. (2009). Self-organizing Systems. *Encyclopedia of Complexity and Systems Science*, *14*, 589.

Beach, A., Bird, T., & Gibbs, A. (1987). Extensional tectonics and crustal structure: deep seismic reflection data from the northern North Sea Viking Graben. Geological Society, London, Special Publications, 28(1), 467-476.

Breard, G. T. (2017). Evaluating self-organizing map quality measures as convergence criteria.

Brown, A. R. (2004). AAPG Memoir 42 and SEG Investigations in Geophysics, No. 9, Chapter 2: Color, Character and Zero-Phaseness.

Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., ... & Zdeborová, L. (2019). Machine learning and the physical sciences. *Reviews of Modern Physics*, *91*(4), 045002.

Castagna, J. P., Swan, H. W., & Foster, D. J. (1998). Framework for AVO gradient and intercept interpretation. Geophysics, 63(3), 948-956.

Castagna, J. P., Sun, S., & Siegfried, R. W. (2003). Instantaneous spectral analysis: Detection of low-frequency shadows associated with hydrocarbons. The leading edge, 22(2), 120-127.

Chen, Q., & Sidney, S. (1997). Seismic attribute technology for reservoir forecasting and monitoring. The Leading Edge, 16(5), 445-448.

Chopra, S., & Marfurt, K. J. (2005). Seismic attributes—A historical perspective. Geophysics, 70(5), 3SO-28SO.

Chopra, S., & Marfurt, K. J. (2007). Seismic attributes for prospect identification and reservoir characterization. Society of Exploration Geophysicists and European Association of Geoscientists and Engineers.

Copestake, P., Sims, A. P., Crittenden, S., Hamar, G. P., Ineson, J. R., Rose, P. T., ... & Bathurst, P. (2003). The Millennium Atlas: petroleum geology of the central and northern North Sea. Geological Society, London.

Cottrell, M., Olteanu, M., Rossi, F., & Villa-Vialaneix, N. (2018). Self-organizing maps, theory and applications. Revista de Investigacion Operacional, 39(1), 1-22.

Dramsch, J. S. (2020). 70 years of machine learning in geoscience in review. *Advances in Geophysics*.

Dreyer, T., & Wiig, M. (1995). Reservoir architecture of the Cook Formation on the Gullfaks field based on sequence stratigraphic concepts. In *Norwegian Petroleum Society Special Publications* (Vol. 5, pp. 109-142). Elsevier.

Equinor Segyio. https://github.com/equinor/segyio.

Equinor. segyio documentation. [Online]. Available: https://segyio. readthedocs.io/en/latest/index.html

Expert.ai Team. (2020). *What is Machine Learning? A definition*. Retrieved from Expert.ai: https://www.expert.ai/blog/machine-learning-definition/

Fahmy, W. A., & Reilly, J. M. (2006). Applying DHI/AVO best practices to successfully identify key risks associated with a fizz-water Direct Hydrocarbon Indicator in the Norwegian Sea. In *SEG Technical Program Expanded Abstracts 2006* (pp. 553-556). Society of Exploration Geophysicists.

Faleide, J. I., Bjørlykke, K., & Gabrielsen, R. H. (2010). Geology of the Norwegian continental shelf. In Petroleum Geoscience (pp. 467-499). Springer, Berlin, Heidelberg.

Farfour, M., Yoon, W. J., Ferahtia, J., & Djarfour, N. (2012). Seismic attributes combination to enhance detection of bright spot associated with hydrocarbons. *Geosystem Engineering*, *15*(3), 143-150.

Forrest, M., Roden, R., & Holeywell, R. (2010). Risking seismic amplitude anomaly prospects based on database trends. The Leading Edge, 29(5), 570-574.

Gibbons, K. A., Jourdan, C. A., & Hesthammer, J. (2003). The Statfjord Field, Blocks 33/9, 33/12 Norwegian sector, Blocks 211/24, 211/25 UK sector, Northern North Sea. *Geological Society, London, Memoirs*, *20*(1), 335-353.

Hesthammer, J., & Fossen, H. (1999). Evolution and geometries of gravitational collapse structures with examples from the Statfjord Field, northern North Sea. *Marine and Petroleum Geology*, *16*(3), 259-281.

Hossain, S. (2020). Application of seismic attribute analysis in fluvial seismic geomorphology. *Journal of Petroleum Exploration and Production Technology*, *10*(3), 1009-1019.

IBM Cloud Education - Machine Learning. (2020). Retrieved from IBM: https://www.ibm.com/cloud/learn/machine-learning#toc-real-world-Lyja9GSr

Johannessen, E. R., Mjøs, R., Renshaw, D., Dalland, A., & Jacobsen, T. (1995). Northern limit of the “Brent delta” at the Tampen spur—a sequence stratigraphic approach for sandstone prediction. In *Norwegian Petroleum Society Special Publications* (Vol. 5, pp. 213-256). Elsevier.

Johnson, A., & Eyssautier, M. (1987). Alwyn North Field and its regional geological context. In *Conference on petroleum geology of North West Europe. 3* (pp. 963-977).

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255-260. (Scikit-learn, 2020)

Kiang, M. Y. (2001). Extending the Kohonen self-organizing map networks for clustering analysis. *Computational Statistics & Data Analysis*, *38*(2), 161-180.

Kohonen, T. (1984). Cybernetic systems: Recognition, learning, self-organization. *Research Studies Press, Ltd., Letchworth, Herfordshire, UK*, 3.

Kohonen, T. (1995). Self-Organizing Maps Springer Verlag. Berlin, Heidelberg, New York.

Kohonen, T. (2012). Self-organization and associative memory (Vol. 8). Springer Science & Business Media.

Larsen, E., Purves, S. J., Economou, D., & Alaei, B. (2018). Is machine learning taking productivity in petroleum geoscience on a Moore's Law trajectory?. *First Break*, *36*(12), 135-141.

Liner, C., Li, C. F., Gersztenkorn, A., & Smythe, J. (2004). SPICE: A new general seismic attribute. In SEG Technical Program Expanded Abstracts 2004 (pp. 433-436). Society of Exploration Geophysicists.

Manouchehri, S., Pham, N., Hellem, T. A., & Roden, R. (2020). A multi-disciplinary approach to establish a workflow for the application of machine learning for detailed reservoir description–Wisting case study. *First Break*, *38*(7), 81-88.

Miljković, D. (2017, May). Brief review of self-organizing maps. In 2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO) (pp. 1061-1066). IEEE.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of machine learning*. MIT press.

Moosavi, V. P. (2014). SOMPY: A Python Library for Self Organizing Map (SOM). Retrieved from Github: https://github. com/sevamoo/SOMPY

Parkinson, D. N., & Hines, F. M. (1995). The lower jurassic of the north viking graben in the context of western european lower jurassic stratigraphy. *Sequence stratigraphy on the Northwest European margin: Norwegian Petroleum Society Special Publication*, *5*, 97-107.

Practical Seismic with Python Tutorial – Transform 21: Practical Seismic in Python. https://github.com/gmac161/practical-seismic-t21-tutorial/blob/main/T21_Practical_Seismic_with_Python.ipynb

Richards, P. C. (1993). *Jurassic of the central and northern North Sea*. British Geological Survey.

Roberts, J. D., Mathieson, A. S., Hampson, J. M., & Spencer, A. M. (1987). Statfjord. *Geology of the Norwegian oil and gas fields*, 319-340.

Roberts, A. M., Yielding, G., & BADDEY, M. (1990). A kinematic model for the orthogonal opening of the late Jurassic North Sea rift system, Denmark-Mid Norway. In Tectonic evolution of the North Sea rifts (pp. 180-199).

Roberts, A. M., Yielding, G., Kusznir, N. J., Walker, I. M., & Dorn-Lopez, D. (1995). Quantitative analysis of Triassic extension in the northern Viking Graben. Journal of the Geological Society, 152(1), 15-26.

Roden, R., & Chen, C. W. (2017). Interpretation of DHI characteristics with machine learning. *First Break*, *35*(5).

Roden, R., Forrest, M., & Holeywell, R. (2005). The impact of seismic amplitudes on prospect risk analysis. The Leading Edge, 24(7), 706-711.

Roden, R., Forrest, M., & Holeywell, R. (2012). Relating seismic interpretation to reserve/resource calculations: Insights from a DHI consortium. The Leading Edge, 31(9), 1066-1074.

Rudolph, K. W., & Goulding, F. J. (2017). Benchmarking exploration predictions and performance using 20+ yr of drilling results: One company's experience. AAPG Bulletin, 101(2), 161-176.

Rutherford, S. R., & Williams, R. H. (1989). Amplitude-versus-offset variations in gas sands. Geophysics, 54(6), 680-688.

Sarhan, M. A. (2017). The efficiency of seismic attributes to differentiate between massive and non-massive carbonate successions for hydrocarbon exploration activity. *NRIAG Journal of Astronomy and Geophysics*, *6*(2), 311-325.

Scikit-learn. (2020). *Scikit-learn User Guide - K-means.* Retrieved from Scikit-learn: Machine Learning in Python: https://scikit-learn.org/stable/modules/clustering.html#k-means

Tamir, D. M. (2021). *datascience@berkeley*. Retrieved from the online Master of Information and Data Science from UC Berkeley: https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/

Taner, M. T. (2001). Seismic attributes. CSEG recorder, 26(7), 49-56.

Taner, M. T., Schuelke, J. S., O'Doherty, R., & Baysal, E. (1994). Seismic attributes revisited. In SEG Technical Program Expanded Abstracts 1994 (pp. 1104-1106). Society of Exploration Geophysicists.

Veeken, P. C., & Rauch-Davies, M. (2006). AVO attribute analysis and seismic reservoir characterization. *First break*, *24*(2).

Xarray Developer. (2021). *xarray: N-D labeled arrays and datasets in Python.* Retrieved from http://xarray.pydata.org/en/stable/index.html#

Yuan, L. (2018). Implementation of self-organizing maps with Python.

Appendix

Seismic Data Loading

In [ ]:
```
# import required libraries
import numpy as np
import segyio
import matplotlib
import matplotlib.pyplot as plt
from matplotlib.offsetbox import AnchoredText
```

In [ ]:
```
# read segy file of the attribute and print EBCDIC header
gradient_crop_segy = r'H:\thesis\New folder\New_gradient_crop.segy'
f = segyio.open(gradient_crop_segy, ignore_geometry = True)
print (segyio.tools.wrap(f.text[0]))
```

In [ ]:
```
#chek sample rate
f.samples[:5]
```

In [ ]:
```
# set the geometry of the cube
with segyio.open(gradient_crop_segy) as segyf:
    ntraces =segyf.tracecount
    sample_rate=segyio.tools.dt(segyf)
    n_samples=segyf.samples.size
    n_il=len(segyf.iline)
```

In [ ]:
```
f = segyio.open(gradient_crop_segy, ignore_geometry=True)
ntraces =len(f.trace)
inlines =[]
crosslines =[]

for h in f.header:
    inlines.append(h[segyio.su.iline])
    crosslines.append(h[segyio.su.xline])

# QC number of traces, inline and crossline numbers.
print(f'{ntraces}traces')
print(f'first 10 inlines: {inlines[:10]}')
print(f'first 10 inlines: {crosslines[:10]}')
```

In [ ]:
```
matplotlib.rcParams['figure.figsize']=(11.75, 8.5)
plt.scatter(crosslines, inlines, marker='s', s=1)

#plot geometry to check if all the traces are aligned and not missing
```

In [ ]:
```
import itertools
uniqil = set(inlines)
uniqxl = set(crosslines)
real = set(zip(inlines, crosslines))
grid = set(itertools.product(uniqil, uniqxl))
missing = grid - real
missing
```

In [ ]:
```
# plot line numbers to check if any are missing
plt.plot(inlines)
```

In [ ]:
```
plt.plot(crosslines[:5000])
```

In [ ]:
```
# check if the numbers are regularly spaced
ils = np.unique(inlines)
xls = np.unique(crosslines)
```

```
inline_interval    = ils[1:] - ils[:-1]
crossline_interval = xls[1:] - xls[:-1]
print(inline_interval)
print(crossline_interval)
```

In [ ]:
```
# set the interval for inlines and crosslines
ils = sorted(uniqil)
xls = sorted(uniqxl)
lineindex = {
    (il, xl): i
    for i, (il, xl) in enumerate(sorted(grid))
}
lineindexils = sorted(uniqil)
xls = sorted(uniqxl)
lineindex = {
    (il, xl): i
    for i, (il, xl) in enumerate(sorted(grid))
}
```

In [ ]:
```
# set up the data for plotting
data = np.zeros((len(ils), len(xls), len(f.samples)))
lineard = d.reshape(d.shape[0] * d.shape[1], d.shape[2])
for il, xl, trace in zip(inlines, crosslines, f.trace[:]):
    lineard[lineindex[il, xl]][:] = trace[:]
```

In [ ]:
```
data.dtype
```

In [ ]:
```
ilines = np.array(sorted(uniqil))
xlines = np.array(sorted(uniqxl))
t = np.array(f.samples)

# Define the central line
central = len(ilines) // 2
```

In [ ]:
```
extent = [xlines[0], xlines[-1], t[-1], t[0]]

plt.imshow(data[central,:,:].T, cmap='RdGy_r', aspect='auto', extent=extent)

plt.ylabel('TWT (ms)')
plt.xlabel('Xline')
plt.show()
```

In [ ]:
```
# repeat the steps for each attribute
```

In [ ]:
```
import xarray as xr
import hvplot.xarray
import panel as pn
```

In [ ]:
```
darray_gradient = xr.DataArray(data= data, dims=['il','xl','twt'], coords={'il':ilin
```

In [ ]:
```
darray_gradient
```

In [ ]:
```
Attributes_dataset = darray_grad.to_dataset(name = 'Gradient')
Attributes_dataset
```

In [ ]:
```
Attributes_dataset['Intercept-Gradient']=(['il','xl','twt'], darray_IminusG)
Attributes_dataset.drop_vars(names='F-N')
```

In [ ]:
```
import dask.dataframe
```

In [ ]:
```
Attributes_df = Attributes_dataset.to_dask_dataframe()
print(Attributes_df)
```

In [ ]:
```
Attributes_df.head(-1)
```

In [ ]:
```
data_csv = Attributes_df[['Gradient', 'Intercept', '(F-N)xF', 'F-N']].values.compute
```

In [ ]:
```
Attributes_df.head(-1)
```

In [ ]:
```
data_csv = Attributes_df[['Gradient', 'Intercept', '(F-N)xF', 'F-N']].values.compute
```

In [ ]:
```
data_csv.shape
```

In [ ]:
```
from numpy import asarray
from numpy import savez
import numpy as np
```

In [ ]:
```
data_npz = savez('newdata1.npz', data_csv)
```

SOM Training

In [ ]:
```
# import libraries for SOM training
%matplotlib inline
import math
import glob
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import urllib3
import sklearn.externals
import random
import matplotlib
from sompy.sompy import SOMFactory
from sompy.visualization.plot_tools import plot_hex_map
import logging
```

In [ ]:
```
# data array
data = np.concatenate([data_arr,data_arr2],axis=1)
```

In [ ]:
```
# names of the attributes
names = ['Gradient', 'Intercept', '(F-N)xF', 'F-N','Intercept*Gradient', 'Intercept
```

In [ ]:
```
# model design
som_model = SOMFactory().build(data, mapsize=[15,15], normalization = 'var', initial
```

In [ ]:
```
#Model training
som_model.train(n_job=2, verbose ='info', train_rough_len=1, train_finetune_len=3)
```

In [ ]:
```
# calculating topographic error
topographic_error = som_model.calculate_topographic_error()
topographic_error
```

Visualizing the results in seismic sections

In [ ]:
```
#viewing component maps of each attribute to see the distribution of neurons.
from sompy.visualization.mapview import View2D
view2D  = View2D(15,15,"rand data",text_size=10)
view2D.show(som_model, col_sz=6, which_dim="all", denormalize=True)
```

In [ ]:
```
# plotting the BMU hit map
from sompy.visualization.bmuhits import BmuHitsView

vhts  = BmuHitsView(12,12,"Hits Map",text_size=7)
```

```
vhts.show(som_model, anotate=True, onlyzeros=False, labelsize=7, logaritmic=False, c
plt.show()
```

In [ ]:
```
# plotting the 'elbow curve' for data clustering
from sklearn.cluster import KMeans
from matplotlib import pyplot as plt

X = som_model._normalizer.denormalize_by(som_model.data_raw,som_model.codebook.matri

distorsions = []
for k in range(1, 20):
    kmeans = KMeans(n_clusters=k)
    kmeans.fit_predict(som_model._normalizer.denormalize_by(som_model.data_raw,som_m
    distorsions.append(kmeans.inertia_)

from kneed import KneeLocator
kn = KneeLocator(x=range(1, 20), y=distorsions, curve='convex', direction='decreasin
print(kn.knee)

figure = plt.figure(figsize=(15, 5))
plt.plot(range(1, 20), distorsions)
plt.vlines(kn.knee, plt.ylim()[0], plt.ylim()[1], linestyles='dashed')
plt.xlabel('Number of Clusters')
plt.ylabel('Distortion')
plt.grid(True)
plt.title('Elbow curve')
```

In [ ]:
```
# Plotting the result of data clustering
from sompy.visualization.hitmap import HitMapView
clusters = som_model.cluster(5)

hits  = HitMapView(12, 12,"Clustering",text_size=10, cmap=plt.cm.jet)
a=hits.show(som_model, anotate=True, onlyzeros=False, labelsize=7, cmap='Set1')
plt.show()
```

In [ ]:
```
#projecting the BMUs over the whole dataset
bmus = som_model.project_data(data)
```

In [ ]:
```
# assigning a cluster label to the whole dataset
cluster_data = clusters[bmus]
```

In [ ]:
```
cluster_data
```

In [ ]:
```
# preparing data to convert to seismic

df = Attributes_dataset_new.to_dataframe()
df.head(-1)
```

In [ ]:
```
df['BMU'] = bmus.tolist()
```

In [ ]:
```
df['Clusters'] = cluster_data.tolist()
df
```

In [ ]:
```
df2 = grad_df.drop('BMU', axis=1)
df2
```

In [ ]:
```
d_bmus = grad_df['BMU'].to_xarray()
d_bmus
```

In [ ]:
```
d_clusters = df2['Clusters'].to_xarray().values
```

```
d_clusters
```

In [ ]:
```
# data set with BMU classification
d2 = d_bmus.values
```

In [ ]:
```
d2.shape
```

In [ ]:
```
# dataset with cluster classification
d_clusters.shape
```

In [ ]:
```
# # plotting the BMU classification (inline 1791)
extent = [xlines[0], xlines[-1], t[-1], t[0]]

b = plt.imshow(d2[465,:,:], cmap='RdYlGn', aspect='auto', extent=extent)

plt.xlabel('Xline')
plt.ylabel('TWT (ms)')
plt.gca().invert_xaxis()
plt.show()
```

In [ ]:
```
# plotting the cluster classification (inline 1791)
extent = [xlines[0], xlines[-1], t[-1], t[0]]

a = plt.imshow(d_clusters[465,:,:], cmap='Set1', aspect='auto', extent=extent)

plt.xlabel('Xline')
plt.ylabel('TWT (ms)')
plt.gca().invert_xaxis()
plt.show()
```

In [ ]:
```
# time slice plotting (time slice -2436)
extent2 = [xlines[-1], xlines[0], ilines[0], ilines[-1]]

c = plt.imshow(d_clusters[:,109,:], cmap='Set1', aspect='auto', extent=extent2)

plt.xlabel('Xline')
plt.ylabel('iline')
# plt.gca().invert_xaxis()
plt.show()
```

In [ ]:
```
# interactive plotting with holoviews
from holoviews import opts

opts.defaults(
    opts.Image(
        width=800, height=600))
```

In [ ]:
```
def plot_inl(inl):
    """
    Plot a single inline using hvplot
    """
    idx = inl - ilines[0]
    da = xr.DataArray(d2[idx,:,:])
    p = da.hvplot.image(cmap='RdYlGn', flip_yaxis=True, flip_xaxis = True)
    return p
```

In [ ]:
```
pn.interact(plot_inl, inl = ilines)
```

In [ ]:
```
line_select = pn.widgets.Select(name='INL Selection', options=ilines.tolist())

pn.interact(plot_inl, inl = line_select)
```

In [ ]:
```
# for cluster data
def plot_inl_cl(inl_cl):
    """
    Plot a single inline using hvplot
    """
    idx_cl = inl_cl - ilines[0]
    da_cl = xr.DataArray(d_clusters[idx_cl,:,:])
    p_cl = da_cl.hvplot.image(cmap='Set1', flip_yaxis=True, flip_xaxis=True)
    return p_cl
```

In [ ]:
```
line_select = pn.widgets.Select(name='INL Selection', options=ilines.tolist())

pn.interact(plot_inl_cl, inl_cl = line_select)
```