CLUSTERING IRRADIANCE VALUES USING UNSUPERVISED MACHINE LEARNING

YEOH YEE JUN

BACHELOR OF ELECTRICAL ENGINEERING WITH HONOURS
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

CLUSTERING IRRADIANCE VALUES USING UNSUPERVISED MACHINE LEARNING

YEOH YEE JUN

A report submitted
in partial fulfilment of the requirements for the degree of
Bachelor of Electrical Engineering with Honours

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

DECLARATION

I declare that this thesis entitled “CLUSTERING IRRADIANCE VALUES USING UNSUPERVISED MACHINE LEARNING is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :

Name : YEOH YEE JUN

Date : 24 MAY 2019

APPROVAL

I hereby declare that I have checked this report entitled “Clustering Irradiance Values Using Unsupervised Machine Learning” and in my opinion, this thesis it complies the partial fulfilment for awarding the award of the degree of Bachelor of Electrical Engineering with Honours

Signature : ....

Supervisor Name : ....

Date : ....

DEDICATIONS

To my beloved mother and father

ACKNOWLEDGEMENTS

Final year project exposes students to a world of wonder in a way that knowledge gained through our respective course might not be harnessed as it ought to be. Accountability is magnified beyond our reach and I, as a final year student, embarked on a new journey to discover the given title as a beginner. I faced a wide variety of challenges along the way and I had to seek out for help at times. Hence, I would like to express my appreciation towards a list of great people.

I wish to express my sincere gratitude to my supervisor, Mr. Kyairul Azmi Bin Baharin for exposing me to the unknown. I am grateful for his undying patience in guiding me to the right path. He showed me the right way to think in an organized manner and handle things differently. He was willing to share his experience and solution to the problems I faced without hesitation. He gave me an opportunity to go through this journey with a free will to discover the job I was held accountable for. His grit and wisdom taught me great dedication and determination. His teaching is engraved in my mindset and I shall implement them in my future undertakings.

It goes without saying that constructive comment is a huge factor in this process. I would like to thank my panel, Ms. Nurliyana Binti Baharin and Ir. Dr. Aminudin Bin Aman for commenting on my work, as well as giving suggestion for future improvement.

Finally, I would like to extend my appreciation to my friends who rendered their help directly and indirectly throughout my final year project.

ABSTRACT

Clustering is an unsupervised machine learning that works by splitting a large dataset into multiple distinctive groups. As a fast developing renewable anergy, output of photovoltaic is prone to fluctuation due to some factors. The purpose of clustering solar irradiance is to determine the daily pattern of irradiance and possibly group those having similar profile. Through this grouping, we can use the clusters obtained as a precursor for solar energy forecasting. The focus of this project is on both Self organizing map(SOM) and K-Means clustering. SOM utilizes plots for visualization purpose and to aid in manual classification. K-Means, on the other hand, makes use of silhouette analysis, elbow analysis and gap statistics analysis to determine the number of cluster. With the help of MATLAB software, a series of supporting detail and evidence is produced with minimal issue. In preliminary clustering, both SOM and K-Means are able to show similarity in the outcome, leading to a high confidence conclusion. In final clustering, an additional software, Weka is used alongside Matlab utilising only K-Means. The final outcome is the same as preliminary result where the optimum number of cluster is three. Irradiance profiles are plotted for categorization consisting of " clear sky", "cloudy" and "overcast".

ABSTRAK

Clustering adalah pembelajaran mesin tanpa pengawasan yang berfungsi dengan memisahkan kumpulan dataset yang besar ke dalam beberapa kumpulan tersendiri. Fotovoltaik merupakan penjana kuasa elektrik mesra alam sekitar yang sedang membangun. Jumlah kuasa elektrik yang dihasilkan tidak tetap. Tujuan pengelasan sinar matahari adalah untuk menentukan corak harian sinaran dan kebarangkaliannya untuk mewujudkan kumpulan yang mempunyai profil yang sama. Melalui kumpulan ini, kita boleh menggunakan kluster yang diperoleh sebagai pendahulu bagi ramalan tenaga solar. Tumpuan projek ini adalah Self Organizing (SOM) dan K-Means. SOM menggunakan plot untuk tujuan visualisasi dan untuk membantu klasifikasi secara manual. Manakala K-Means menggunakan teknik-teknik pengesahan tertentu untuk menentukan bilangan kelompok. Dengan bantuan perisian MATLAB, keputusan terperinci mampu dihasilkan tanpa isu. Pada tahap awal, kedua-dua kaedah ini dapat menunjukkan hasil yang seiras. Oleh itu, kesimpulan dapat ditentukan dengan keyakinan yang tinggi. Pada tahap terakhir, satu perisian tambahan yang bernama Weka telah diguna di samping perisian asal. Kedua-dua perisian ini dapat menunjukkan keputusan yang sama seperti pada tahap awal. Misi menentukan kategori setiap kluster dapat dijalankan dengan label “langit yang cerah”, “mendung sederhana” dan “mendung penuh”.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

UTeM	-	Universiti Teknikal Malatsia Melaka
FKE	-	Faculti Kejuruteraan Elektrik
SOM	-	Self Organizing Map
KMOR	-	K-Means Outlier Removal
VRKM	-	Variance Reduced K-Means
ALO	-	Ant Lion Optimization
LM	-	Last Leap
LML	-	Last Major Leap
PCA	-	Principle Component Analysis
DSOM	-	Deep Self Organizing Map
HRSOM	-	High Resolution Self Organizing Map
LRSOM	-	Low Resolution Self Organizing Map
U-AHC	-	Agglomerative Hierarchical Clustering
MH	-	Metropolis Hastings
Qe	-	Quantization Error
Te	-	Topographic Error
U	-	Neuron Utilization
i	-	Assignment of cluster
M_i	-	New centroid
$s(i)$	-	Silhouette value
$\mathrm{SSE}_{\mathrm{total}}$	-	Total sum of squared error
D_r	-	Sum of pairwise distance
W_k	-	Within dispersion measure
K	-	Number of cluster
$\mathrm{Gap}(k)$	-	Gap between reference and observation data
s_{k+1}	-	One standard error

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Overview

Photovoltaic(PV) technology is a growing industry in the past 10 years with promising rate. This trend sees large scale PV system getting integrated into the grid. However, the grid experiences more fluctuations as more PV is integrated into the grid. This phenomenon is caused by the variability of the PV output while the variability itself comes from natural causes and is uncontrollable and inevitable. Sudden changes in weather can cause immediate drop in the output. Hence, if the grid operator is not well prepared, stability problems might happen inside the grid. To address the issue, s solution through forecasting is implemented. With accurate forecasting, we can properly anticipate the changes before the actual occurrence. In order to obtain accurate forecasting, we need to classify the profile of irradiance on any day. If the classification can be done properly, then we can develop forecasting model for each weather profile. This project investigates the ability of unsupervised machine learning to automatically cluster the daily irradiance based on weather type. The main focus of this chapter is separated into five subtopics for better illustration. Project background, motivation, problem statement, objective and scope of the project are included in this proposal. Purpose of this project will be elaborated with a series of supporting details.

1.2 Project Background

Efficiency can be dubbed as a prime mover of any industry. Even the slightest fall in efficiency can deeply impact the overall income of an industry. As significant as it is, the smallest drawback can be magnified in larger industries as compared to smaller industries. Hence, the role of data scientist is highly anticipated in recent years. Given the opportunity to take up the role as a beginner in machine learning, research in the relevant field is extremely important in foundation building.

Deep comprehension of the project title plays a pivotal role in achieving the objectives. The origin, working principles, factors relating to the problem are put into account to foster understanding in depth. Photovoltaics is the direct conversion of energy obtained from the sun radiation into electricity. This takes advantage of the photoelectric effect that can be induced in some materials, causing them to release electrons when absorbing photons. Current is then induced when free electrons are captured.

Edmund Bequerel was the first person credited for the photoelectric effect in 1839, when he discovered that only specific materials would produce small amounts of current under the exposure of light. About 60 year later in 1905, Albert Einstein won a Nobel prize in physics after his breakthrough in accurately describing the fundamentals of photoelectric effect. The first photovoltaic module was created by Bell Laboratories in 1954. The module was costly at first and in the 1960s, the space industry began to make the first serious use of the technology to provide power aboard spacecraft. Space industry's involvement accelerated the technology advancement and thus, its reliability was established, and the cost began to drop. However, this technology was truly recognized in the 1970s due to energy crisis, as a source of power for non-space applications.

Figure 1.1 shown below portrays the basic operation of a solar cell. Solar cells are constructed from the same semiconductor materials, typically silicon. In solar cells, a thin semiconductor wafer is specifically treated to form an electric field, with positive and negative polarity opposite one another. Electrons are able to escape the atoms in the semiconductor material when photons reach the solar cell. If electrical conductors are connected to the positive and negative sides, completing an electrical circuit, electric current is produced due to the electron movement.

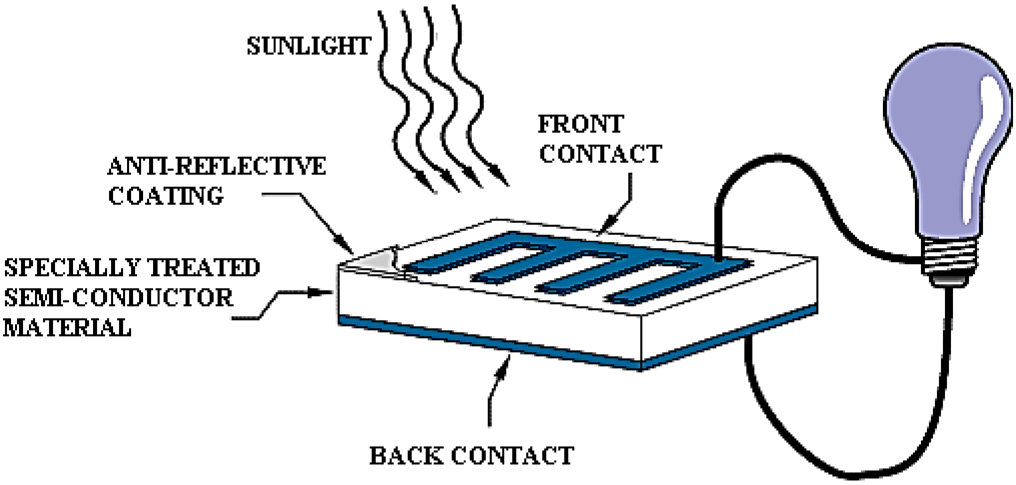


Figure 1.1: Basic operation of a photovoltaic cell

Placing a number of solar cells electrically connected to each other in a support structure or frame is called a photovoltaic module and the current induced is directly proportional to the amount of light hitting the module [1]. An illustrative diagram of photovoltaic module is shown in Figure 1.2 below.

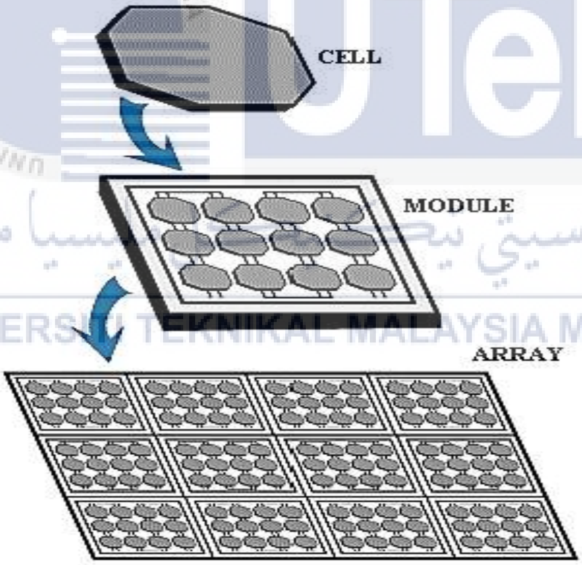


Figure 1.2: Photovoltaic module

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups. Clustering is important in data analysis. A good clustering algorithm is able to identity clusters irrespective of their shapes. Clustering is also known to extract information from a large data set and transform it into an understandable form for further use. The simplest stages involved in clustering algorithm are shown in Figure 1.3 [2].

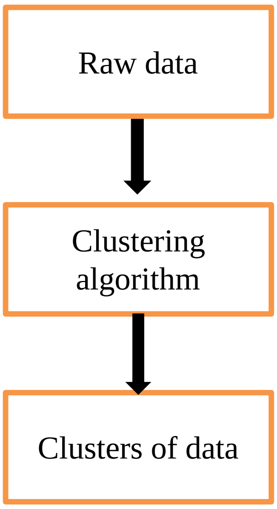


Figure 1.3: Basic clustering stage

1.3 Motivation

Photovoltaic has been around for as long as when technology advancement took place. It had a fun filled beginning that took the whole world by storm. The focus is always on the efficiency in an endeavor to powering the entire nation. Today, photovoltaics are 20% efficient and combined cycle power plants are 60% efficient. Table 1.1 shows a series of power plant type along with their efficiency [3,4,5].

Table 1.1: Efficiency of power plant type

Type	**Efficiency(%)**
Coal fired power plant	32 – 42
Natural gas fired power plant	32 – 38
Hydro turbine	85 – 90
Wind turbine	30 – 45
Solar thermal system	20
Photovoltaic system	15
Geo thermal system	35
Nuclear plant	0.27
Diesel engine	35 – 42

As we all know, photovoltaic is a better option in term of environmental friendliness. Any effort that could ease the solar related work with good accuracy is desired. Thus, having an opportunity to bring about enhancement to the solar process would ignite a sense of responsibility and willpower in me to kick start a meaningful project. Deep understanding in the relationship between solar irradiance and other variables is crucial for critical thinking in the attempt of acquiring best solution. Making full use of current software to help ease the whole lengthy process is ideal. Clustering is a known method in unsupervised machine learning and is widely used in many areas in data analysis and data mining applications. It is the task of grouping a set of objects so that objects in the same group are more similar to each other than to those in other groups. To begin with, I would use Matlab software and Weka software to discover the hidden patterns within the datasets from FKE's Solar Lab.

1.4 Problem Statement

In order to provide accurate forecasting, we need to have sufficient dataset for each weather type. Typical classification is done by separating the data into three; clear sky, cloudy and overcast. However, manual classification can be a hassle. It is especially tedious if the amount of dataset high. In each of 'big data', we already have the computational ability to process huge amount of information in a relatively short time. In term of energy forecasting, it is possible to use machine learning to do the classification. However, a lot of questions remain unanswered such as the accuracy of machine learning, data size required for machine learning to perform sufficiently and which model is the best to be used. Thus, an effort to discover the grouping, hidden patterns in solar irradiance values dataset provided by FKE's Solar Lab would help to answer the questions. A proper procedure using two known high accuracy methods, Self Organizing Map and K-Means are deployed with Matlab. At the same time, K-Means is also used in Weka software for comparison purposes.

1.5 Objectives

i) To find out the hidden pattern and strength of relationship between meteorological variables from the database in FKE’s Solar Lab.

ii) To evaluate the natural clustering of daily irradiance profiles using selected unsupervised machine learning method.

iii) To validate the results with manual classification.

1.6 Scope

a) This project utilizes Matlab R2016B software and Weka software.

b) The focus is on unsupervised machine learning, clustering.

c) The results and findings are solely based on dataset from FKE’s Solar Lab.

d) Only 9 a.m. to 7 p.m. dataset is used to avoid insignificant result.

e) One month data and three years data are used in preliminary clustering anf final clustering respectively.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

This chapter focuses on the variation of clustering method due to modification of the-state-of-the-art. Theories, benefits, comparisons and researchers' discovery will be discussed in this chapter to narrow down the options available. A litany of relevant applications dealing with real life problems are reviewed for idea expansion and to set a baseline for this project.

2.2 Machine Learning

Machine learning is a part of artificial intelligence (AI) that serves to equip systems the ability to automatically learn and improvise from past experience without being explicitly programmed. This implies the absence of specific algorithms that run specific tasks. The algorithm is better represented as shapeless with high conformity that adjusts or adapts itself in accordance to the input. The input can be in any form, ranging from numeric, shape, color, image and whichever data that have the ability to show variations [6].

The learning process is dependent on the type of algorithm. Each application requires different optimization for maximum accuracy. This process involves capturing patterns in data and learn from previous computations to produce reliable, repeatable decisions and results, thereby, producing precise predictions in the future based on the previous data [7]. The frequentative aspect of machine learning is imperative as the computed or constructed models are exposed to new input. The model is capable of adapting independently. The ultimate mission is for the computer to learn automatically without human intervention or assistance and adjust actions accordingly. Machine learning algorithms are often categorized as supervised or unsupervised.

2.2.1 Supervised Machine Learning

Supervised machine learning algorithms works by learning the pattern in the input dataset and apply the acquired model to predict the future event. Supervised learning has a prerequisite that the dataset used to train the algorithm must link to the correct answer. An inference is produced from large input dataset and is used to compare with another dataset to predict possible outcome. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly. Supervised learning usually consists of regression and classification [8]. The techniques of Supervised Machine Learning algorithms can be further split into the followings in Figure 2.1 [9].

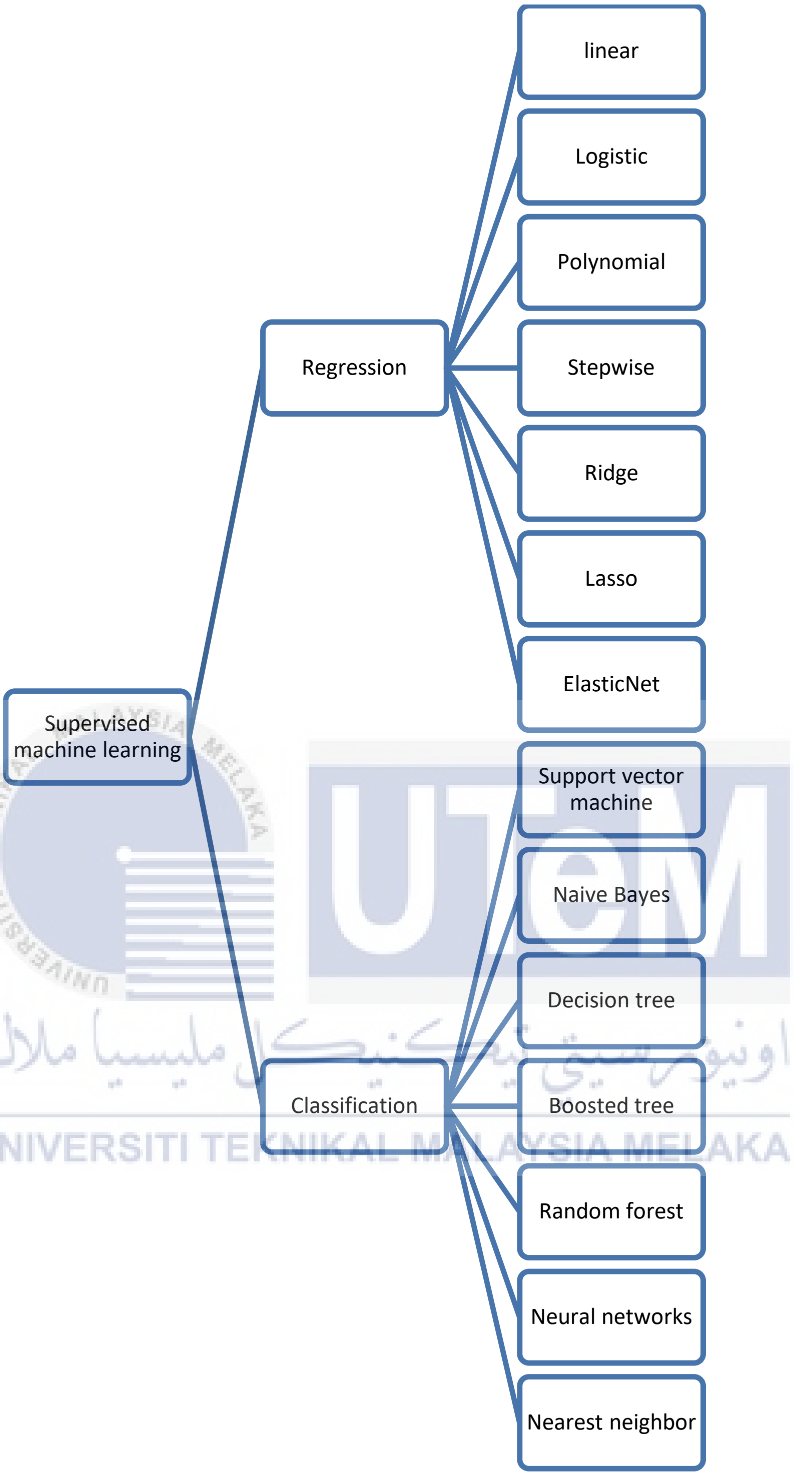


Figure 2.1: Supervised machine learning

2.2.2 Unsupervised Machine Learning

Unsupervised machine learning algorithms are utilized when the dataset's feature is unknown. Hidden pattern, classification, and characteristic are the main findings of this algorithm. Unsupervised learning picks up how a system can deduce a function to describe a hidden pattern from untagged data. The system doesn't identify the correct output. Instead, it investigates the data to acquire inferences from datasets to describe hidden structures from untagged data [10]. Unsupervised machine learning includes the followings as portrayed in Figure 2.2 below.

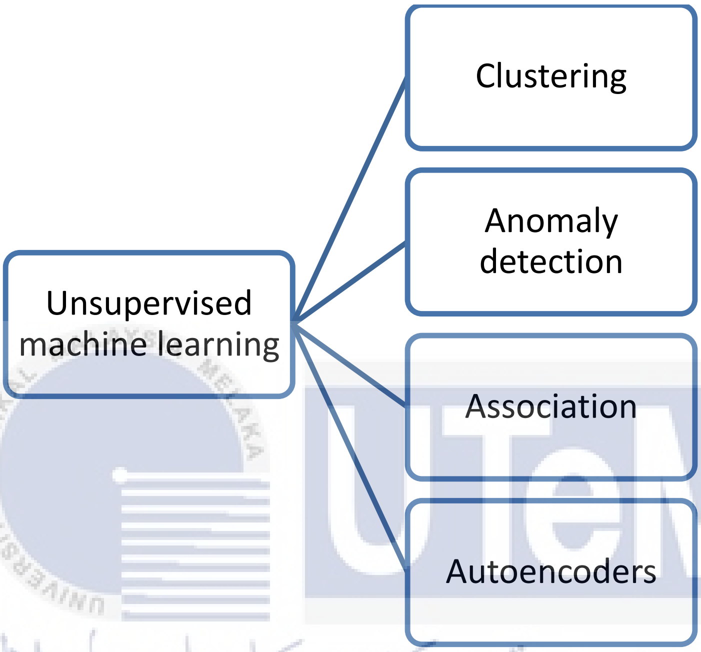


Figure 2.2: Unsupervised machine learning

2.3 Application of Clustering

Taking heed from real life examples gives us a clearer picture of the steps, precaution, errors and all sort of possibilities that we might encounter. In medical field, medical imaging has a vital role in body examination. Forming segmentation technique with the help of k-means clustering, fuzzy c-means and artificial neural network helps in developing a thorough system that detects early stage cancer. Extraction of the desired part of the image is made easier with the combination of clustering methods. This breakthrough aids doctor heavily in diagnosis, thereby, producing an even more accurate analysis. This endeavor reduces human error by a huge fraction and increases the confidence level of decisive moment [11]. Dengue, a known fatal illness caused by mosquito has been on the rise at an alarming rate. This infection exists in a particular part of the world. By utilizing self-organizing map, the hidden pattern of this endemicity can be revealed. Health sectors can greatly benefit from this discovery in timely decision making and in turn, creating a precise solution to the problem. Hence, an area with high risk can be given the highest priority in the strategy [12]. Unnecessary radiation exposure can be monitored through clustering of data. It is tremendously difficult to find the right dose due to large variability of the machine. Justification in which the exposure to patient is probable can be a tough feat. Clustering of the multivariate data obtained from the monitoring system could help doctors in identifying the right conduct [13]. In agriculture, seasonal plantation is often practiced to maximize the product output. Rotation of different plantation along with the prerequisites takes place all year round. Pollution is inevitable due to fertilization, irrigation and land use pattern. The relationship between land use and water is crucial in determining the water resource protection regulations. Implementation of self-organizing map clustering helps categorizing samples from ponds into groups with different nutrient intensity and factors in relation to their respective land use indices. The findings could show the pollution level with respect to the land use and thus, simplifying the multiple factors complications and making water resource management easier [14]. In finance, through financial statement, bankruptcy trajectory can be studied through clustering. Visualization and analysis using self-organizing map can help in identifying the pattern of finance. By using the data over the years as input, a company can set a better direction in administration [15]. In engineering field, power plants have always been the main source of energy. It plays an important role in supplying electrical energy for domestic, commercial and industrial uses. However, the conventional power plants are all fossil fuel based and tend to cause pollution to the environment. Renewable energy comes into play to curb fuel shortage problem as well as the environmental pollution issues. Photovoltaic power generation can greatly make use of renewable energy and replace the conventional way but it has a shortcoming concerning the output of PV power. The PV succumbs to intermittence and fluctuation in accordance to weather condition making it difficult to identify the approximate output. A prediction model comprised of improved k-means, grey relational analysis and Elman neural network is used to acquire accurate prediction. Firstly, k-means clusters the power datasets. This is followed by the grey relational analysis to identify the similarity days and optimal similarity days and finally, Elman neural network is used to analyze the nonlinear relationship between the variables [16].

2.4 Type of Clustering

Clustering is a complex work of splitting a population or dataset into a defined or predefined number of group. The data points pertaining to one particular group share the highest similarities among themselves. Each group has its distinctive characteristic and is different from other groups. In short, the duty of clustering is to partition groups with identical or similar criteria and place them into clusters [17, 18]. Clustering variations are shown in Figure 2.3 below.

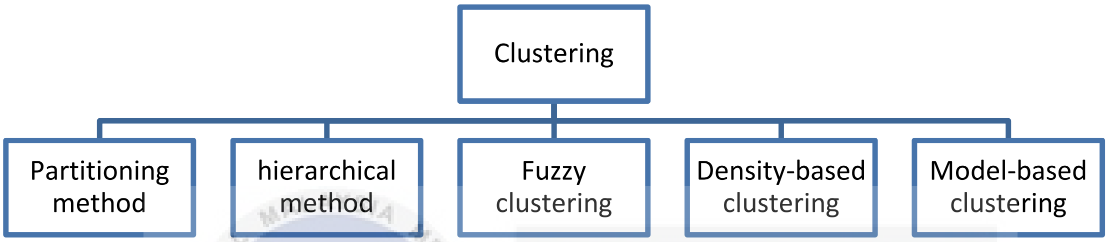


Figure 2.3: Type of clustering

Figure 2.3 shows the common types of clustering methods in the current world. However, the search will be based on research works done by data scientists to foster comprehension in this project. The weaknesses, strengths and opportunities for improvement are analyzed. The choice of method in this literature part is solely reflected by its relevancy to the data type and also, the compatibility of the data used in this project, irradiance.

2.4.1 K-Means Clustering

K-Means is the most famous partitioning clustering method. The traditional K-Means clustering algorithm is easily affected by the noise, and outliers while also being prone to overfitting. Modification and advancement of all sort pave a clearer way for smoother and greater result. A deep insight into k-means clustering algorithm based on self-determined learning speed oversees space for improvement by selecting the best training subgroup to construct the starting cluster model based on self-paced learning theory is of paramount importance. It enhances the ability for initial model generalization by placing good subsets of samples consequently until optimal performance is obtained or when the training datasets are final and absolute.

By analyzing the experimental results, the proposed clustering algorithm achieves better performance compared with the traditional K-means clustering algorithm. This method simplifies the process of solving the K-means clustering algorithm to make the result more intuitive [19]. Addition of self-paced learning feature to the original K-Means clustering algorithm proves to be better than the conventional method. It extracts training samples by step starting from easy to difficult. With self-paced regularization factor, the noise in K-Means clustering training and the influence of outliers on the clustering result are reduced, leading to apparent improvement in problem involving the ease of K-Means clustering falling into local optimal solution.

Speaking of outlier influence, the next method is K-means Outlier Removal, KMOR. This method has an additional “cluster” that traps the outliers. Given a desired number of clusters k, the KMOR algorithm separates the dataset into user defined cluster number groups and an extra group with the purpose of containing outlier KMOR algorithm is found to be capable of assigning all outliers into a group during the clustering procedure and can cluster data and detect outliers concurrently. This strongly compensates existing clustering algorithm with outlier detection. Since outlier detection in the KMOR algorithm is a natural part of the clustering process, points can move in and out of normal clusters and similarly for outlier cluster, thus translating to better accuracy and runtime.

Runtime has always been an important feature of clustering especially when it comes to huge amount of data [20]. Variance Reduced K-Means, VRKM decreases stochastic noise caused by the conventional K-Means, and accelerates k-means by using variance reduction technique [21]. Specifically, it is a position correction mechanism to correct the drift of the cluster centers. Further optimization for computational cost reduction leads to a new variant of the variance reduced K-Means named VRKM. VRKM and VRKM++ outperform the state-of-the-art method, and can reach up to 4 times faster in terms of clustering processing speed.

Random initialization of centroid positions has always been a drawback of the state-of-the-art. Ant Lion Optimization, a nature inspired optimization technique, has the ability to improve the quality of clustering of K-Means clustering algorithm. To obtain better quality of clusters, the value of intra-cluster distance should be minimum and the value for F-measure should be maximum. The hybrid of K-Means and Ant Lion Optimization method, k-means-ALO performs preferably better than the state-of-the-art in terms of sum of intra-cluster distances and F-measure. The level of confidence for the statistical analysis is considered as 0.10 which translates into an accuracy of 90% by K-Means-ALO [22]. Next, another sample algorithm that uses animal's survival wisdom to enhance the ability of k-means, elephant herding. The whole idea of this algorithm stands firmly based on the assumption that the location of each elephant is considered as one solution. The population is split up into smaller group or clan. Each clan is led by a female leader. In each generation, male elephants with lowest fitness function value are left behind and away from their group. Hence, the elimination is dependent on the function values [23].

Correlation information between features is deemed significant. Through Laplacian smoothing, a structured sparse k-means clustering algorithm concurrently acquires the cluster assignment and feature weights [24]. The features are capable of distinguishing different clusters and can be selected structurally. This implementation has great structure for better visualization and hence, greater interpretability. At the same time, the modified algorithm shows visibly better accuracy in comparison with its predecessor, sparse k-means.

Conventional k-means algorithm requires user to predetermine the number of cluster upon initiation. Hence, the implementation of precise automatic estimation of the natural number of clusters present in a dataset is crucial to make a clustering method truly unsupervised. This modification definitely betters its predecessor by bringing in a huge leap in performance. In k-means, the minimum of the pairwise distance between cluster centers plunges as the defined number of clusters rises. Through observation, the last significant reduction can be observed just as the defined number of cluster goes beyond the actual number of clusters. The Last Leap (LL) and the Last Major Leap (LML) are designed in such a way that the number of cluster can be determined without prompting manual input of cluster number. LL is able to determine the number of clusters that are distributed with distinctive boundary, whereas LML seeks the number of clusters with similar size. Any difference between the values of LL and LML can thus inform a user about the hidden cluster structures within the dataset. The proposed techniques are not bound by the size of the dataset, making them an ideal solution for seemingly large datasets. LL and LML remains its' integrity as the dataset number rises. However, this trend persists no more than 50 clusters as well as the number of dimensions at 50. The findings are proven to support the claim that performance of LL and LML are on par with the best cluster number estimation mechanisms around while having a much lower computational burden [25].

Uncertain data clustering is an important task in the research of data mining. Various traditional clustering methods are modified with new similarity measurements to solve this issue. The concept focuses more on the evaluation of distribution similarity between uncertain data objects. Based on the Kullback Leibler divergence(KL-divergence) and the Jeffreys divergence(JS-divergence), a novel K-medoids method for clustering uncertain data, named UK-medoids, has the ability to solve the said issue. This method is claimed to perform significantly better than the classical algorithm in the quality of classification. Looking into the newly proposed method, the JS-divergence outperforms KL-divergence in term of stability [26].

2.4.2 Self Organizing Map Clustering

The Kohonen card (Self Organizing Map, SOM) forms a very well structured neural network model with distinctive boundaries, shapes, color as well as distance. This method is often used by researchers in applications involving declassification. Som has the ability to discover pattern beyond human eyes. Oftentimes, hidden pattern can be unearthed from the plots generated by SOM. The Self-Organizing Neural Network Map (SOM) is considerately great for classification and clustering. The benefit of SOM is prevalent in some mixed applications as it preserves the topology structure while mapping. This adds more detail as well as information to the existing applications without losing its identity [27]. However, it still has flaws. The computational time cost and the input space normalization are the main drawbacks faced in this method. During normalization of the inputs space, the classification loses its precision and the neurons are unable to differentiate between the original inputs. This is due to the abandoned magnifications in due dataset because every dataset has its own magnitude. SOM is used without any kind of supervision or earlier training. A proposed improvement aimed at addressing these drawbacks relies on improving the normalization mechanism with more intelligent techniques that can reduce not only the time required but also the computer processing capability. PCA (Principal Component Analysis) is found to be able to help achieve this in preparing the inputs space prior to using the neural network classifier [28].

Multiple efforts and modifications are done to enhance the performance of Self Organizing Map. The next effort focuses on comparative study of Self Organizing Map and Deep Self Organizing Map using MATLAB. DSOM consists of SOM and the sampling operators are positioned in a way to overtake the SOM in the outcome. The result includes generating another 2D map, then it is used as an input to the next self-organizing layer. The result improves as it progresses through the higher layer. Profound Self-Organizing Map (DSOM) computation which consists of multiple layers of self-sorting out guide and examining administrator are reengineered in the lucrative way to bring about enhancement to the performance [29].

Going deeper, the next strategy takes winning frequency of neurons into account. Winning frequency based SOM converges faster than conventional SOM. The modified SOM is able to organize itself in a better way than the conventional SOM in every part of the input data possible. The modified SOM, too, has lower quantization error (Qe) and topographic error (Te) and at the same time, it improves neuron utilization(U) [30]. Quantization Error measures the average distance from each input data to its winner while Topographic Error describes how well does the SOM preserve the topology of the input dataset. Neuron Utilization measures the percentage of map neurons that stand out of one or more input data in the map.

$$Q_e = \frac{1}{N}\sum\nolimits_{i=1}^{N} \|X_i - \overline{W}_i\| \tag{2.1}$$

Where N = input data quantity

Where $\overline{W}_i$ = winner's weight vector of the input vector

Where $\|.\|$ = Euclidean distance metric

$$T_e = \frac{\sum_{j=1}^{N} u(X_j)}{N} \tag{2.2}$$

Where N = input data quantity

Where $u(X_j) = 1$ if the winner and 2nd winner are not 1-neighbors

Where $u(X_j) = 0$ if the winner and 2nd winner are 1-neighbors

$$U = \frac{1}{mn}\sum_{i=1}^{mn} u_i \tag{2.3}$$

Where, $u_i = 1$: if the neuron i is the winner of some inputs

Where, $u_i = 0$: if the neuron i is not the winner of some inputs

Taking advantage of modern computing hardware, High Resolution Self Organizing Map, HRSOMs can act as an automated mechanism in a way that it constantly raises the visualization ability in its' intrinsic nature to discover the complex relationships in inputs with high number of variable to a great length for vast applications. It possesses an ability to raise the classification performance when used as a pre-processor intended to pre-dispose the classification outcome, for a number of learning problems [31]. There is no much proven evidence claiming that Low Resolution Self Organizing Map, LRSOMs would raise the results quality in visualization work. However, HRSOMs shows the opposite. HRSOMs perform especially well in visualization, showing complex details with higher variable number input vectors when laid down to a lesser variable display space, such outcome could not be seen in LRSOMs. Furthermore, classification performances score between 10% and 20% when using HRSOMs as pre-processors in ensemble systems when compared to those obtained using LRSOMs. The findings, too, prove the existence of optimal cost-benefit SOM size for any given learning problem. However, the resolution does not perform at high consistency for all tasks encountered. As a pre-processor, dataset improvement using SOMs can aid the improvement in generalization of a supervised classifier and this is followed by HRSOM implementation that boosts the results further. Lastly, HRSOM is found to have dimensionality reduction and in turn, reduces the computation time.

2.4.3 Hierarchical Clustering

Hierarchical method clusters data objects in the form of a tree known as hierarchy. Each group in hierarchy is reflected by a node. There are two forms in Hierarchical clustering, namely Agglomerative clustering and divisive clustering. Quality of the clusters is of paramount importance. Agglomerative clustering inspects the quality of clusters using three parameters: Cohesion measurement, Silhouette index and Elapsed time. The proposed method perceives that a cluster with high cohesion, zero and positive values of silhouette index and a low elapsed time depicts a high quality cluster. A descriptive dendrogram for agglomerative Hierarchical clustering is shown in Figure 2.4 below [32].

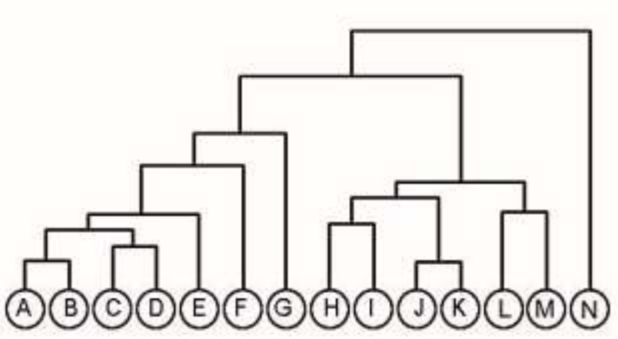


Figure 2.4: The dendrogram for an agglomerative hierarchical clustering (bottom-up)

There exists a kind of data that is not fixed in space. Uncertain data clustering representation is closely related to imprecision, overload, or random feature that is indirect when storing real life dataset. An established solution to deal with the issue through an information-theoretic approach, prototype-based agglomerative hierarchical clustering method, dubbed as U-AHC, uses a new uncertain linkage criterion for cluster merging [33]. This criterion allows the comparison of uncertain objects based on information-theoretic as well as expected-distance measures. The findings show that U-AHC generally outperforms other methods of common purpose in term of accuracy and, by looking tthe efficiency alone, this method is up on par with the fastest baseline version of agglomerative hierarchical clustering.

2.4.4 Hybrid Clustering

A hybrid involves more than just one method. A combination of different methods would increase the efficiency as each method compensates for the weakness of other method. To promote better understanding of the relationship between clusters, dataset is first separated by K-means clustering algorithm. Then, the clustering results are grouped into clusters with similar characteristic via hierarchical clustering method [34].

A new method of Gaussian mixture learning comprising of hierarchical clustering and expectation-maximization algorithm works both accurately and efficiently for large datasets. Hierarchical clustering provides an initial guess for the expectation-maximization algorithm to take place. Adaptive splitting for hierarchical clustering, too, is introduced in this modification. It steps up the quality of the initial guess and subsequently improves both the accuracy and efficiency of the combination [35].

2.5 Distance Metric

Distance metric is an imperative part of clustering as it could alter the end result. It is the distance that the algorithm uses in computation and the type of distance metric chosen is dependent on the data type. Different distance metric could generate different outcome to a certain extent and therefore, getting the same outcome with different metric would indicate result with high accuracy.

Distance metric learning, whose purpose is to find the optimal distance that separate different classes is fundamental for producing good machine learning results. It is known that every dataset possesses different characteristic and thus, behaves differently in nature. It is learned that learning a global distance metric is never enough to obtain satiable outcome when highly diversified distributed data is considered. Kernel embedding methods are proven to be efficient in coping with the issue. However, it immediately becomes computationally uncontrollable as the number of samples rises. An efficient method is proposed to learn multiple local distance metrics instead of a single global metric [36]. More specifically, the training samples are separated into clusters that are unconnected to each other. Then, a distance metric is used to train each cluster independently. Furthermore, common properties that exist within separate clusters are preserved through the use of a global regularization coefficient in the learned metric space. By concurrently learning multiple distance metrics using a single combined optimization mechanism, this method stands out against single distance metric learning methods. This method also proves to be much more viable when compared to other conventional methods of the same purpose.

Taking a closer look at single distance metric, with a mindset that the previous method is better in every way possible, it is imperative that we look into each metric distance to gain deep comprehension before making finalized points. Each metric distance possesses different theory and working principle with different feature uniqueness. Metric distance can be great for one particular application while showing opposite result when other used on other application. To begin with, we look into Hierarchical clustering algorithm with Euclidean distance. This method is analyzed alongside with conventional K–Means. The finalized findings prove that the former method is indeed more superior than the latter in term of accuracy. However, the former method has a setback, a longer computational time [37].

Joining both the fuzzy c-means clustering and distance metric learning in an end to end terminology, the findings are proven to maximize the splitting ability among different clusters [38]. Metropolis Hastings (MH) in this method is able to solve issue leading to convergence to local optima, the drawback that most methods are constantly trying to avoid. At the same time, excellent Global optima is achievable when repetition is enough. The performance of this method is akin to most dimensionality reduction algorithms, and is able to find the number of influential variable in the projected space in an automatic manner, and in most cases, it has the best clustering accuracy in these dimensions.

2.6 Determination of Number of Cluster

In SOM, the number of cluster is selected through visualization. The parameter involved includes the distance between clusters, size of cluster and shape of cluster. In K-Means, determination of number of cluster is uncertain due to the nature of the algorithm. The initial K is predetermined by user beforehand. Silhouette analysis, Elbow analysis and Gap statistics analysis are chosen as verification methods in Weka and Matlab.

2.6.1 Visualization in Self Organizing Map

It is challenging to examine multivariate data as the best model achievable in visualization is limited to three dimensions. Having data point with more than three variables would require dimension reduction. SOM transforms multidimensional data into two dimensional data and projects it on a map. Visualization requires the user to determine the number of cluster based on specific needs. On the map, the identifiable traits include the apparent number of group and the distinctiveness of each group. On the other hand, the outlier is an identifiable yet subjective trait. User has to evaluate the importance of the outlier as it is problem specific [39, 40].

2.6.2 Silhouette Analysis in K-Means

Silhouette method is a graphical interface portraying the classification of clusters with their respective values. The silhouette value simply shows the strength of relation between a data point and other data points in the same cluster while being compared to other data points in other clusters. In short, each data point is given a specific score of how likely they are to be in a given cluster. If there are too many or too few clusters, some of the clusters will typically exhibit much narrower silhouettes than the rest. Thus silhouette plots and averages are useful in determining the natural number of clusters within a dataset [41].

2.6.3 Elbow Analysis in K-Means

Elbow method is widely accepted as a more superior method as compared to silhouette method. This method determines the optimal cluster number through sum of squared error (SSE) computation. For a predetermined range of cluster number, the total SSE is calculated for every different cluster number. For instance, setting the cluster number to two, the distance between all data points and the centroid within a cluster is summed up for both clusters. The total within cluster distance between data point and centroid of two of the clusters is known as the SSE. To identify the appropriate cluster number, elbow or knee point is identified from the output trend. As the SSE drops significantly throughout the trend with increasing cluster number, the first SSE with lower gradient would form a knee-like shape on the previous SSE with one cluster number lesser. The cluster number at the knee point is then selected as the optimum choice [42,43].

2.6.4 Gap Statistics Analysis in K-Means

Gap statistics is applicable to all kinds of clustering methods. The gap statistics compares the sum of variation within a cluster for a selected range of cluster number between observation data and reference data. The distance measured can be varied through the distance metric choice. The reference data is obtained from Monte Carlo sample extracted from reference distribution. Similar to K-Means mechanism, user is required to select a range of cluster number to run the test. To make the analysis better, the size of reference data can be manipulated to acquire desired gap curve. The size of reference data set, 500 is found to generate outcome with considerably high accuracy even after multiple runs. Hence, the same dataset size is used across all computation to acquire outcome of the same category. A standard deviation is defined to be the allowable range of uncertainties for the optimal cluster number selection. This deviation minimizes the optimal cluster number size and is known as one standard error [44,45,46,47].

2.7 Matlab Software

Matlab is an interactive software used in engineering fields. It stands for Matrix Laboratory in full and is a high performance and high level language. It is capable of working with entire matrices instead of one value at a time. It comes with multiple sets of tools for visualization and computation [48]. For machine learning wise, unsupervised learning can be accessed through Neural Net Clustering app.

The Neural Net Clustering app solves a clustering problem with the help of a self-organizing map (SOM). It aids in data selection, defining the network architecture, and training the network. The dataset concerned can be selected from MATLAB® workspace. Alternatively, Matlab also provides sample datasets within the software for learning purposes. Upon finishing the network training, the results obtained can be analyzed with a list of visualization tools. Performance evaluation of the network can then be made on a test set. If the acquired result is not satisfactory, retraining of the network can be done by varying the settings or replacing the dataset with an even larger dataset to increase the accuracy of the result.

MATLAB scripts can be generated based on the exact desired settings to replicate the result. Also, the training can be modified through the script. The trained network can be saved to examine new dataset or homogeneous clustering matters. The app is equipped with options to generate multiple deployable versions of the trained network. For instance, the trained network can be employed using MATLAB Coder™, Simulink® Coder tools or MATLAB Compiler™ [49].

2.8 Weka Software

Weka is an open source graphical user interface(GUI) for various users. It was developed at the University of Waikato in New Zealand. It has a series of machine learning algorithms for data analysis. It consists of a wide variety of tools for classification, clustering, regression, visualization and data preparation that converts data file to ARFF format. Weka has included an extensive built-in help in the software. A clear manual is included in the software while online material is becoming increasingly ubiquitous to aid in comprehension part of the project.

2.9 Simple Random Sampling

For extremely large amount of dataset, the computation time can be underwhelming. Hence, choosing the data points as representatives is an important task. In data mining, or specifically, machine learning, it requires that the chosen data best reflects the supposed criteria. This method is also used to narrow down the large data points to make analysis easier [50,51]. While it is crucial to choose the best data point, avoiding bias is a prerequisite in random sampling. The evaluation on the performance of random sampling focuses on the ability of the estimation to remain unbiased under all circumstances [52]. In this project, random sampling is used to ease the manual classification and to produce unbiased analysis.

2.10 Summary

To sum up, all modified single methods perform considerately better than their respective conventional methods. An undeniably great method might not be at peak performance when tested on other dataset with different nature. A combination of two methods is proven to be more efficient as one method can compensate for another method's weakness. Distance metric, on the other hand, has a clear winner. Learning multiple local distance metrics instead of a single global metric does a wonder. The former is able to outshine single distance metric learning methods, while being competitively useful than other multi-metric learning methods. After gaining insight into the working principle and efficiency of each method, a combination of Self Organizing Map and K–Means as a cross-check to each other, is reckoned as plausible in preliminary clustering using Matlab. The distance metrics used are 'cityblock' and 'euclidean' in this preliminary clustering. As for the final part of clustering, Matlab and Weka software are both used for comparison using only K-Means clustering method.

CHAPTER 3

METHODOLOGY

3.1 Overview

From acquiring preliminary results by using one month's dataset to full inspection using one year's dataset and three years' dataset, this project utilizes Self Organizing Map clustering(SOM) and K-Means clustering with MATLAB 2016b and K-Means clustering with WEKA software to acquire a wholesome outcome with supportive evidence. In this chapter, methods as well as the steps, are described in detail to illustrate the whole working principle involved to achieve the objectives. The respective diagram and coding are included in each method.

3.2 Gantt Chart

The undergone flow of project is sketched and listed in Gantt chart as shown in Appendix A and Appendix B to keep track of the progress.

3.3 Flowchart of Clustering

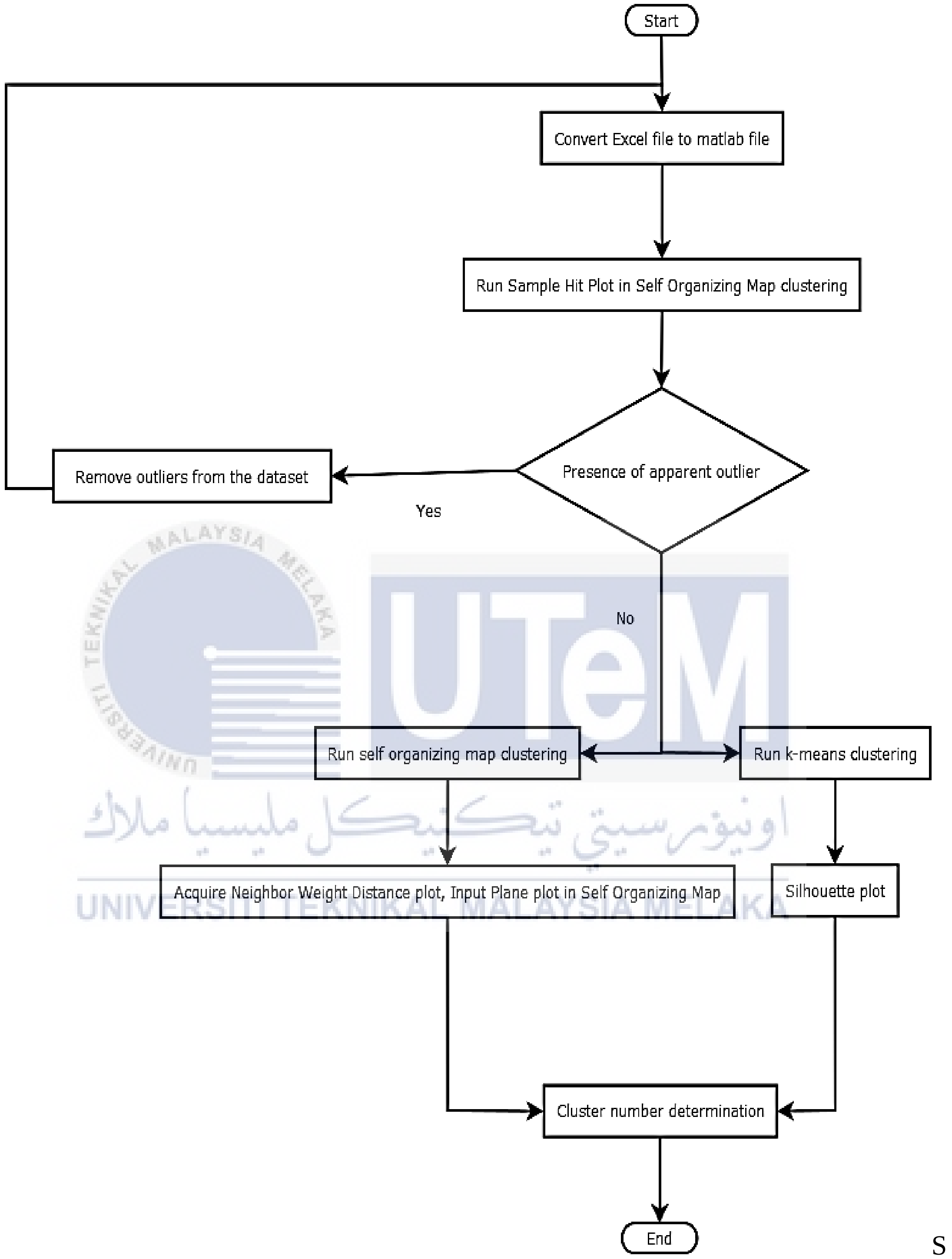

Start
Convert Excel file to matlab file
Run Sample Hit Plot in Self Organizing Map clustering
Presence of apparent outlier
Remove outliers from the dataset
Yes
No
Run self organizing map clustering
Run k-means clustering
Acquire Neighbor Weight Distance plot, Input Plane plot in Self Organizing Map
Silhouette plot
Cluster number determination
End
S

Figure 3.1: Flowchart of preliminary clustering

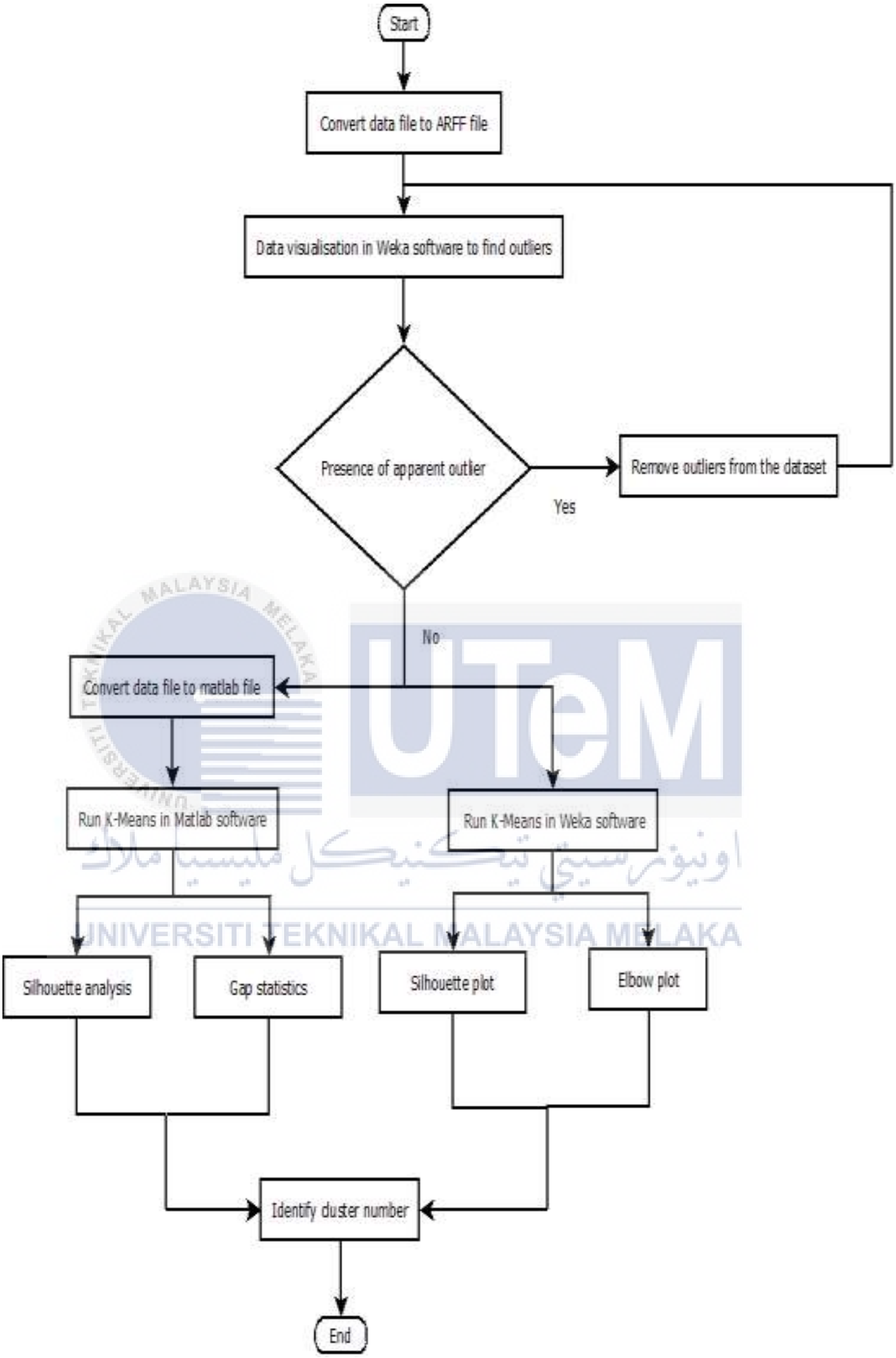

Start
Convert data file to ARFF file
Data visualisation in Weka software to find outliers
Presence of apparent outlier
Yes
Remove outliers from the dataset
No
Convert data file to matlab file
Run K-Means in Matlab software
Run K-Means in Weka software
Silhouette analysis
Gap statistics
Silhouette plot
Elbow plot
Identify cluster number
End

Figure 3.2: Flowchart of final clustering

3.4 Self Organizing Map Clustering

This method is only utilized in Matlab software for visualization. SOM portrays a set of data with high number of variable as a quantized two-dimensional image in an orderly fashion. Every data vector is positioned into one point or node in the map. Graphically, the distances of the data vectors in the map reflect similarities between the items. The average smoothed distances between the nearest SOM models are represented by light colors for small mean differences, and darker colors for larger mean differences.

3.4.1 Data Structure

It is important to understand the nature of data in clustering. There are two fundamental types of input vectors: One that occurs simultaneously and one that occurs sequentially in time. This project utilizes concurrent vector as irradiance data is treated as a multi-dimensional parameter which happens at the same time.

3.4.1.1 Simulation with Concurrent Input in A Static Network

First, a static network with no feedback or delay is simulated. To promote better comprehension in this part, the network is assumed as a single input vector. A static network for concurrent input is shown in Figure 3.3 below.

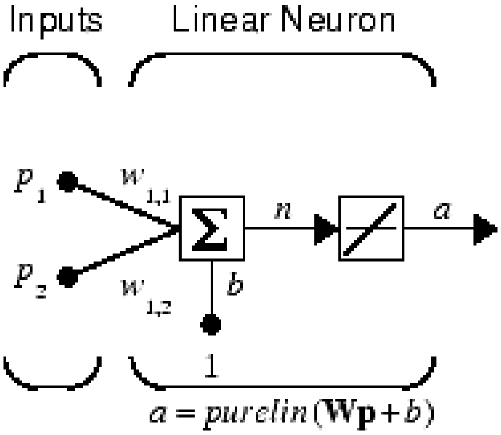


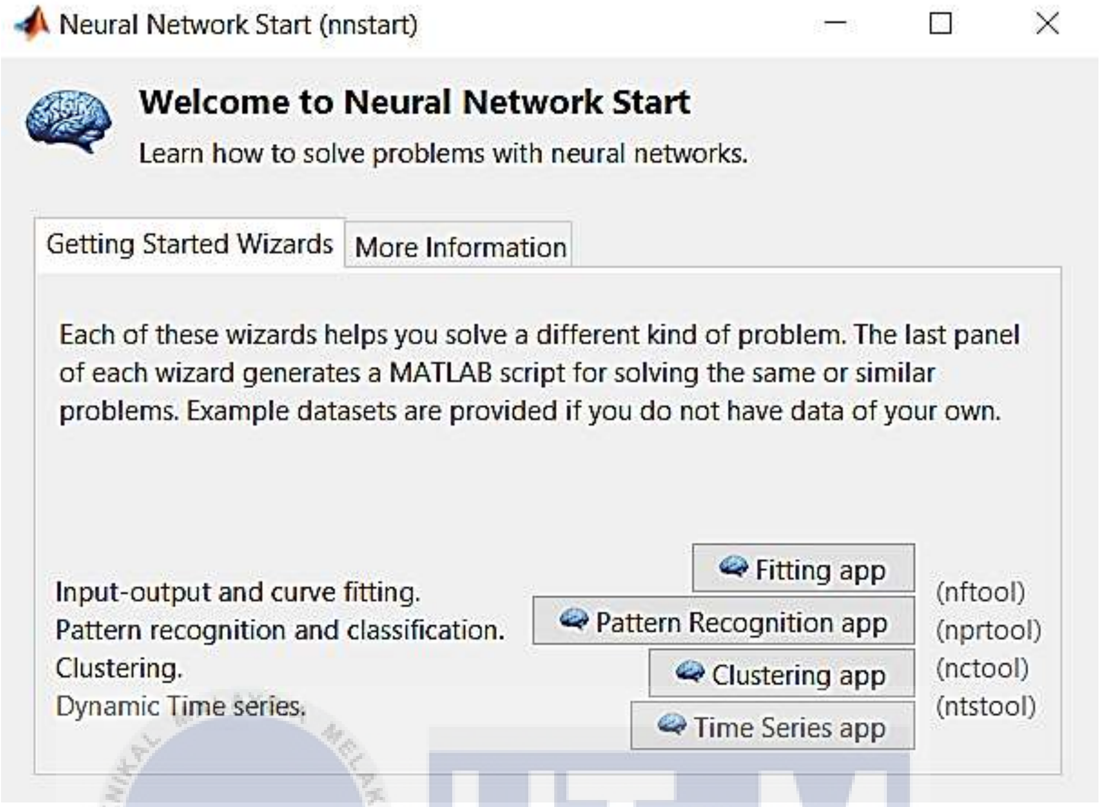
Figure 3.3: Static network for concurrent input

A single matrix of concurrent vectors is used as an input to the network, and the expected output is of the same kin as the input. The result would be the same if there were four networks operating in parallel and each network received one of the input vectors and produced one of the outputs. The sequence of the input vectors is not significant as they do not interact with each other.

3.4.2 Neural Network Clustering Tool

Neural Network Clustering tool is used in Matlab to cluster dataset with SOM. This part describes the steps involved in the making of Self Organizing Map network with the help of software. The steps are as follows;

1. First, Neural Network Start GUI is opened with “nnstart” command. This network will then show multiple options as shown in Figure 3.4 below.



Neural Network Start (nnstart)

Welcome to Neural Network Start

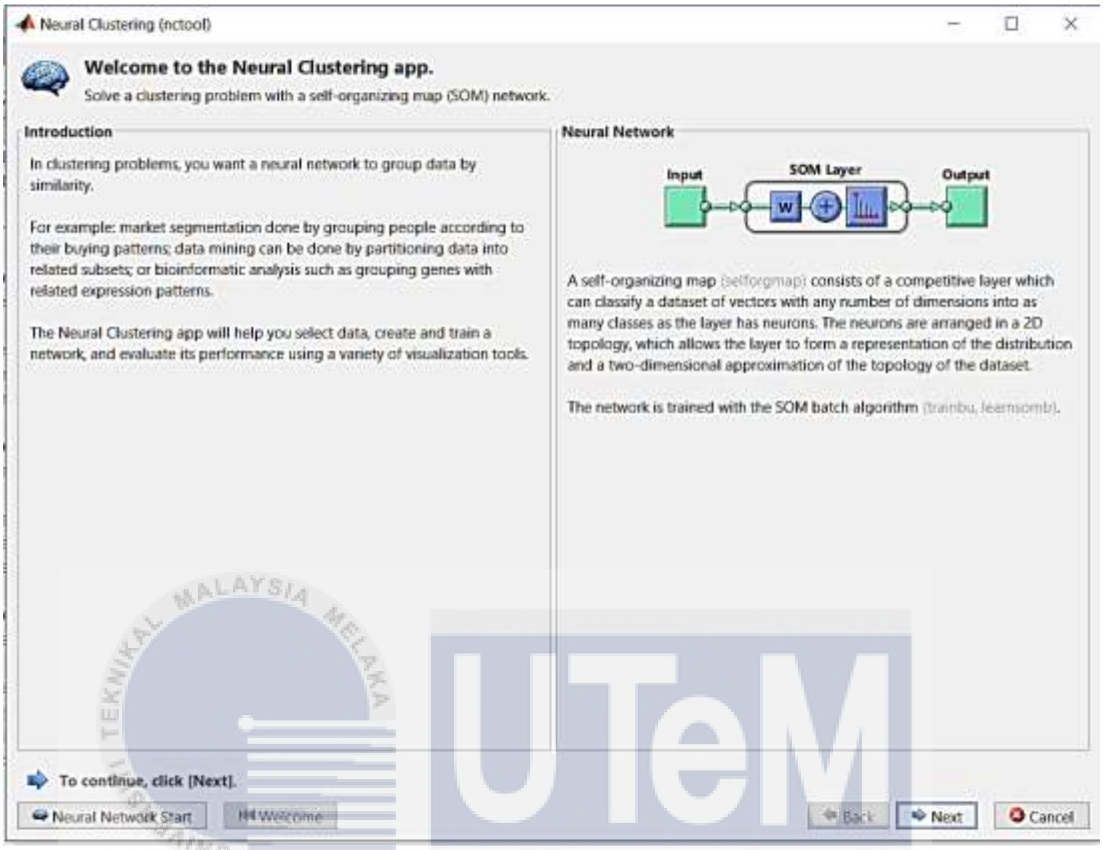
Learn how to solve problems with neural networks.

Each of these wizards helps you solve a different kind of problem. The last panel of each wizard generates a MATLAB script for solving the same or similar problems. Example datasets are provided if you do not have data of your own.

Input-output and curve fitting.	Fitting app	(nftool)
Pattern recognition and classification.	Pattern Recognition app	(nprtool)
Clustering.	Clustering app	(nctool)
Dynamic Time series.	Time Series app	(ntstool)

Figure 3.4: Neural network start GUI

2. Clustering Tool is chosen to open the Neural Network Clustering Tool. An app will pop up as shown in Figure 3.5 below.



Neural Clustering (nctool)

Welcome to the Neural Clustering app.

Solve a clustering problem with a self-organizing map (SOM) network.

Introduction

In clustering problems, you want a neural network to group data by similarity.

For example: market segmentation done by grouping people according to their buying patterns; data mining can be done by partitioning data into related subsets; or bioinformatic analysis such as grouping genes with related expression patterns.

The Neural Clustering app will help you select data, create and train a network, and evaluate its performance using a variety of visualization tools.

Neural Network

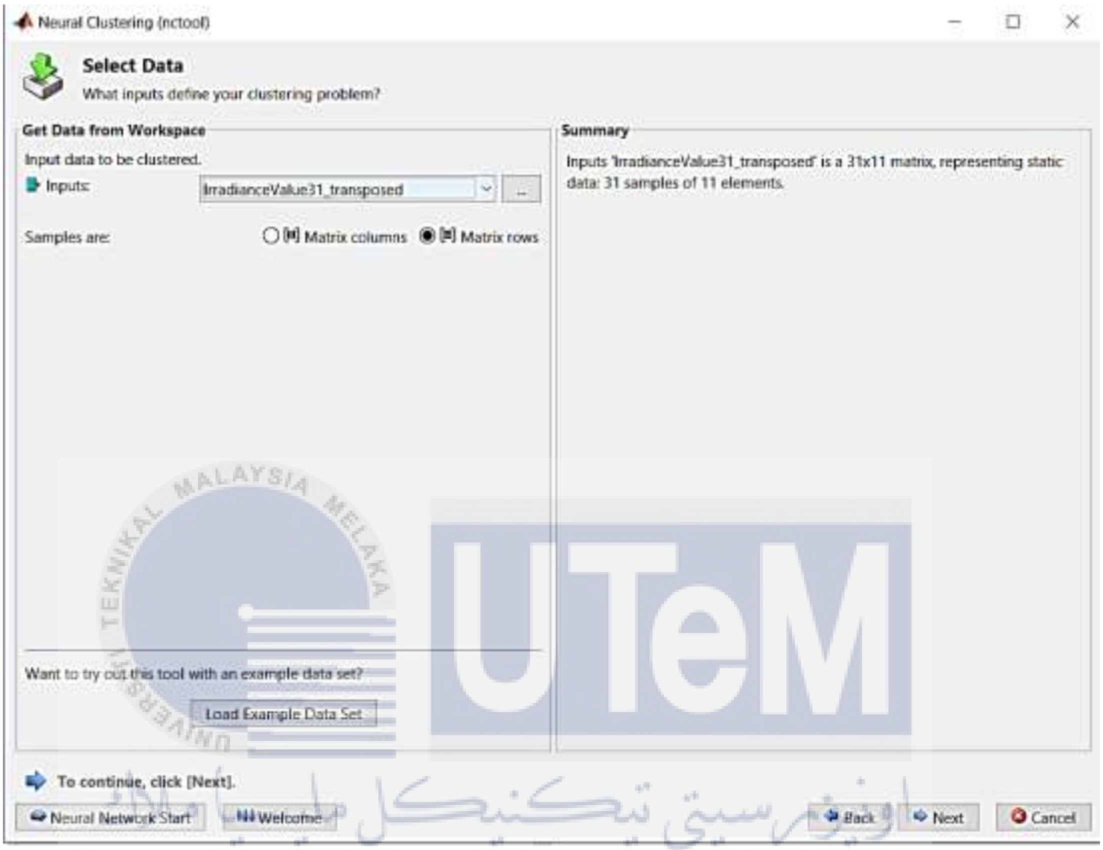
A self-organizing map (selforgmap) consists of a competitive layer which can classify a dataset of vectors with any number of dimensions into as many classes as the layer has neurons. The neurons are arranged in a 2D topology, which allows the layer to form a representation of the distribution and a two-dimensional approximation of the topology of the dataset.

The network is trained with the SOM batch algorithm (trainbu, learnsomb).

To continue, click [Next].

Figure 3.5: Neural clustering app

3. “Next” is then clicked to trigger the Select Data window to appear as shown in Figure 3.6. In this window, "…” at the input row is clicked to select “IrradianceValue31_transposed” dataset. Then, “Matrix rows” is selected due to the respective alignment of the data set.



Select Data

What inputs define your clustering problem?

Get Data from Workspace

Input data to be clustered.

Inputs: IrradianceValue31_transposed

Samples are: ○ Matrix columns ● Matrix rows

Summary

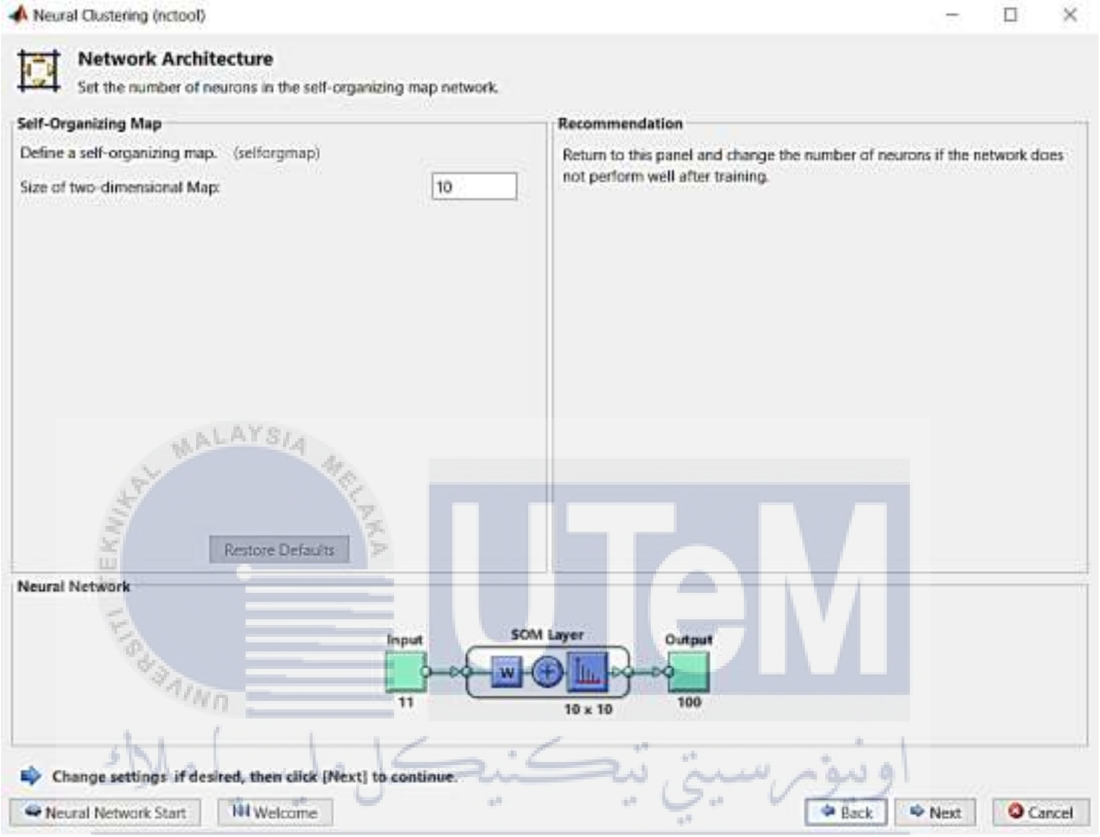
Inputs 'IrradianceValue31_transposed' is a 31x11 matrix, representing static data: 31 samples of 11 elements.

Want to try out this tool with an example data set?

To continue, click [Next].

Figure 3.6: Select data window

4. Upon clicking “Next” in the previous step, Network Architecture window appears. Neuron number is set in this part before proceeding to the next step. A neural network is then displayed at the bottom as seen in Figure 3.7.



Neural Clustering (nctool)

Network Architecture

Set the number of neurons in the self-organizing map network.

Self-Organizing Map

Define a self-organizing map. (selforgmap)

Size of two-dimensional Map: 10

Recommendation

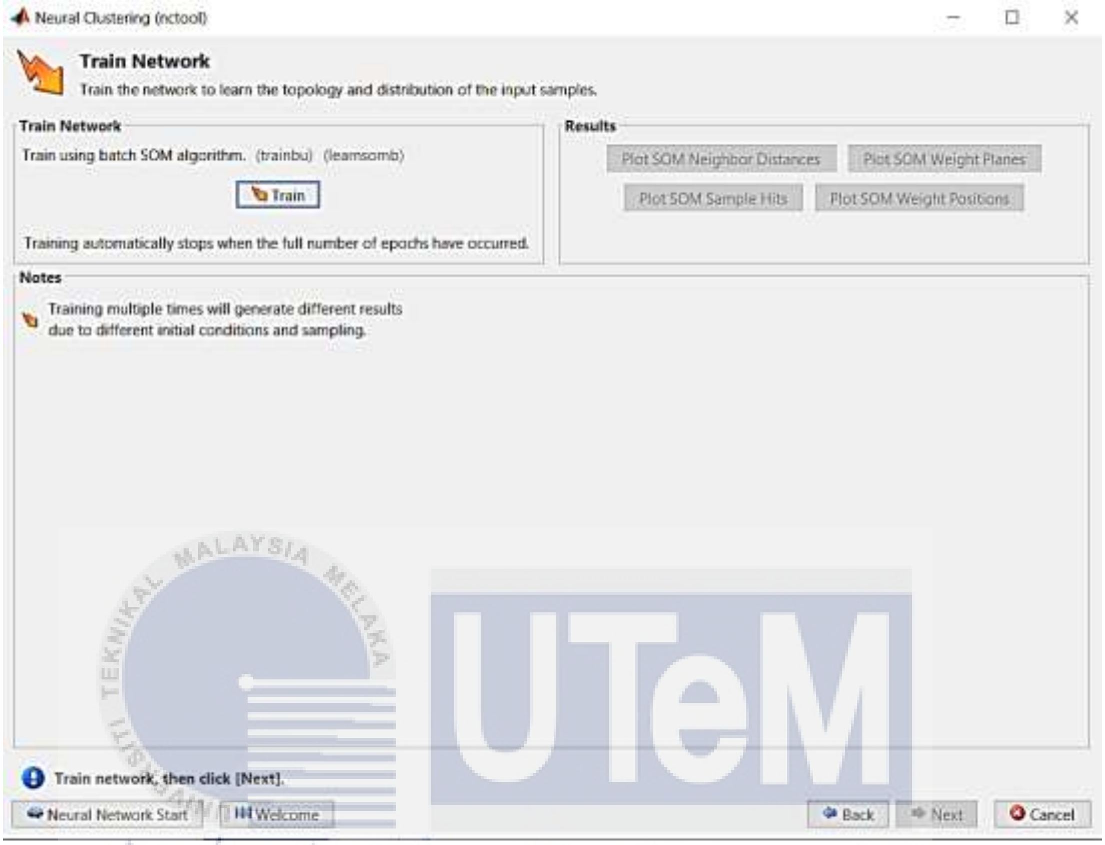
Return to this panel and change the number of neurons if the network does not perform well after training.

Neural Network

Change settings if desired, then click [Next] to continue.

Figure 3.7: Network architecture

5. “Next” is then clicked to load the data set and trigger Train Network as shown in Figure 3.8 below.



Neural Clustering (nctool)

Train Network

Train the network to learn the topology and distribution of the input samples.

Train Network

Train using batch SOM algorithm. (trainbu) (learnsomb)

Training automatically stops when the full number of epochs have occurred.

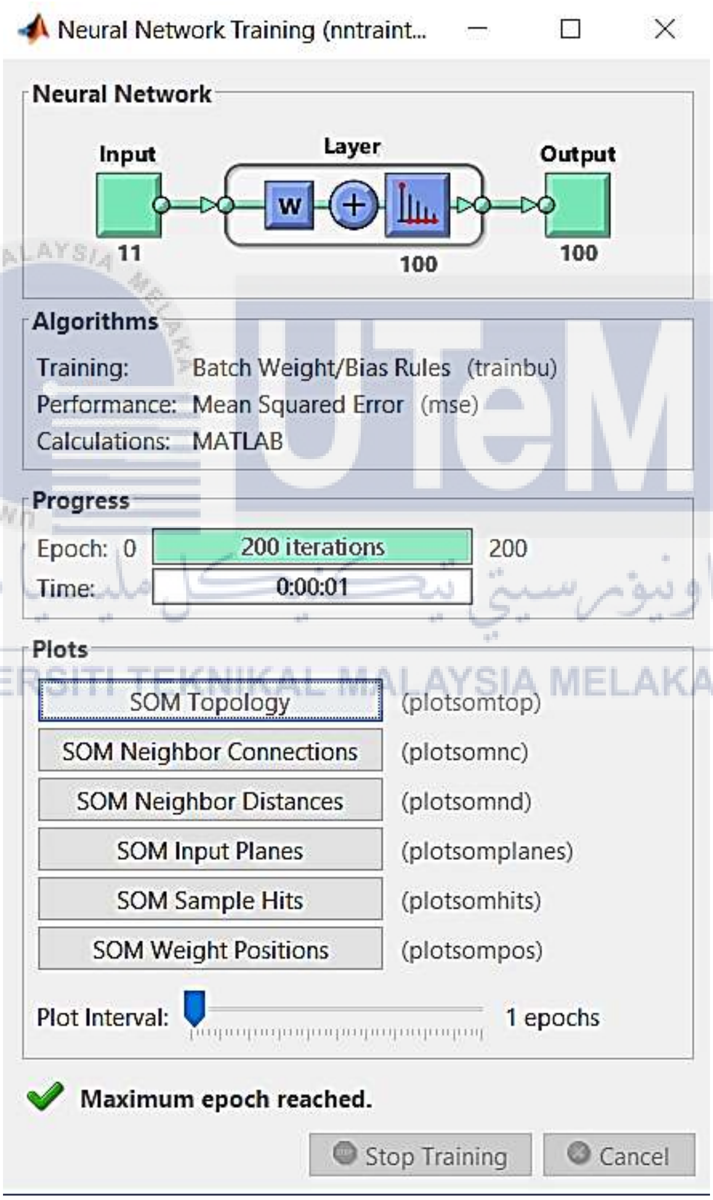
Notes

Training multiple times will generate different results due to different initial conditions and sampling.

Train network, then click [Next].

Figure 3.8: Train network 1

6. In the Train Network, “Train” is clicked to trigger a new window, Neural Network Training. The Epoch is set at maximum, 200 by default which represents the number of iteration. The iteration number is proportional to the accuracy of the resulting outcome. However, this setting also corresponds to the execution time. Thus, setting the number of Epochs in accordance with the data set is essential through trial and error. A number of plots will be available for view after the training as shown in Figure 3.9 below.



Algorithms

Training: Batch Weight/Bias Rules (trainbu)
Performance: Mean Squared Error (mse)
Calculations: MATLAB

Progress

Epoch: 0 200 iterations 200
Time: 0:00:01

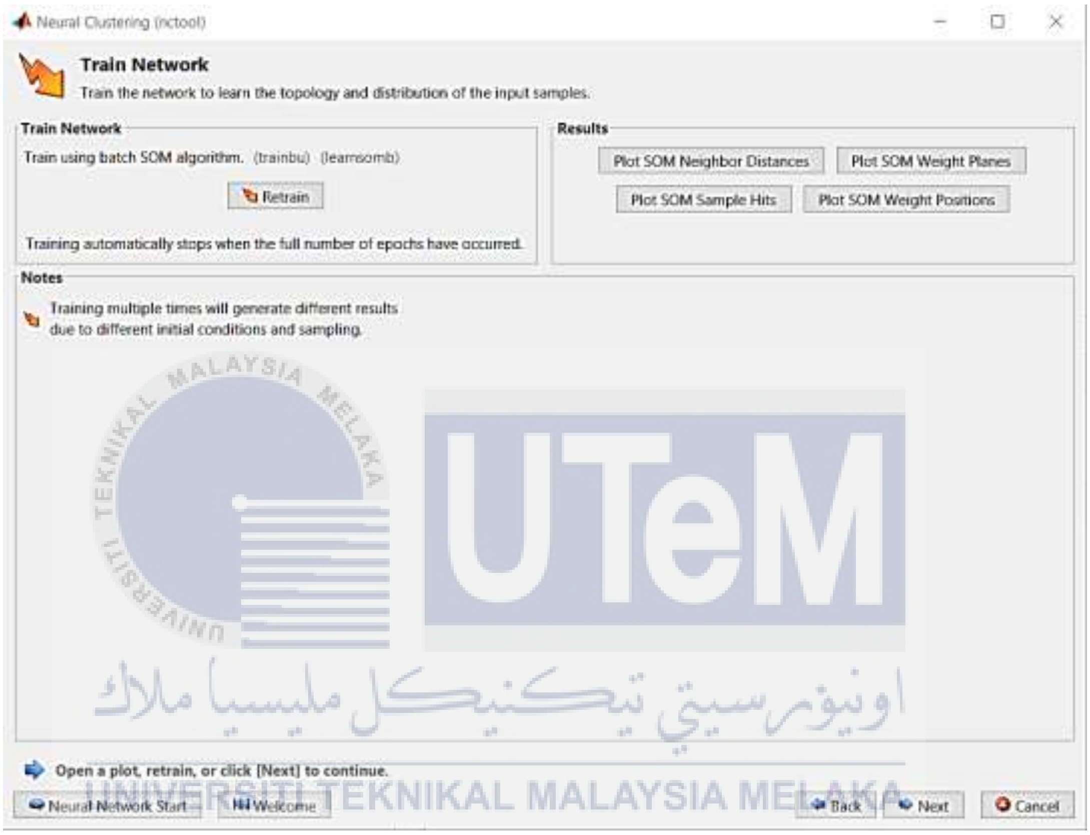
(plotsomtop)
(plotsomnc)
(plotsomnd)
(plotsomplanes)
(plotsomhits)
(plotsompos)

Plot Interval: 1 epochs

✓ Maximum epoch reached.

Figure 3.9: Neural network training

7. Going back to the Train Network, “Retrain” option is available and is clicked to repeat step 7. This process is repeated for a few times and the plots are analyzed for each repetition for comparison. The process is then stopped when the comparisons between repetitions show high similarity or resemblance. The network with retrain option is located in Figure 3.10 below.



Train Network

Train the network to learn the topology and distribution of the input samples.

Train Network

Train using batch SOM algorithm. (trainbu) (learnsomb)

Training automatically stops when the full number of epochs have occurred.

Results

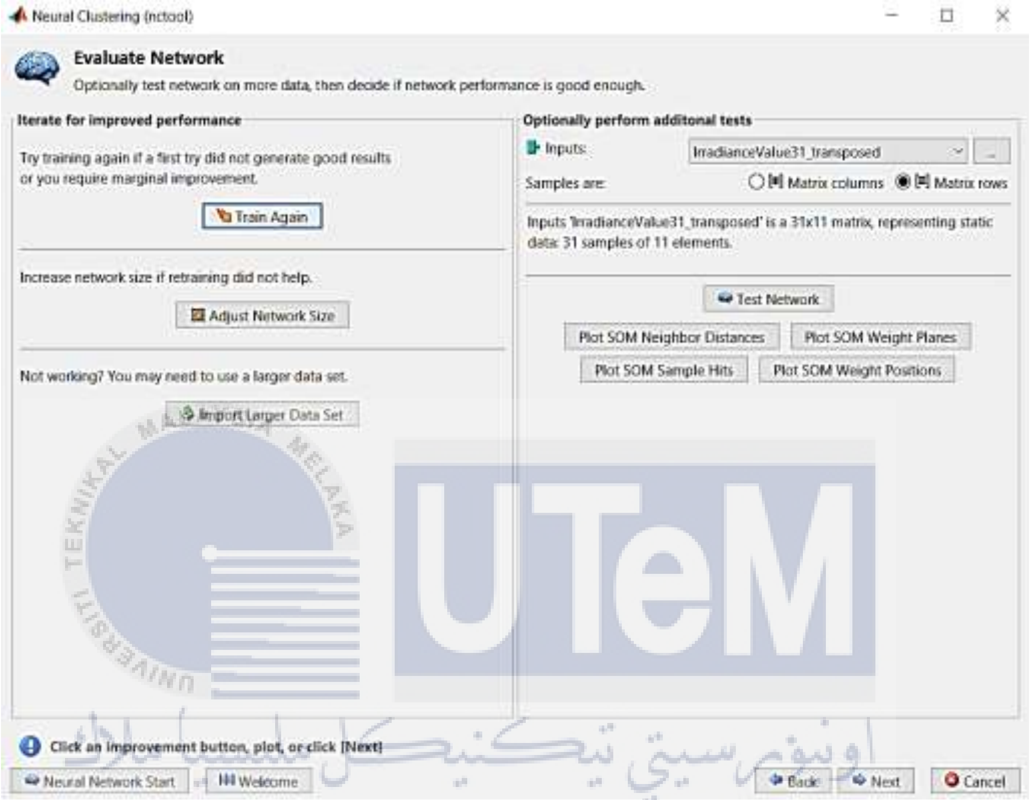
Notes

Training multiple times will generate different results due to different initial conditions and sampling.

Open a plot, retrain, or click [Next] to continue.

Figure 3.10: Train network 2

8. In the Train Network, “Next” is clicked to trigger Evaluate Network as shown in Figure 3.11. More options are found in this network including network size adjustment. The number of neuron can be adjusted to acquire a totally different plots. Manual classification is then done on the plots generated.



Neural Clustering (nctool)

Evaluate Network

Optionally test network on more data, then decide if network performance is good enough.

Iterate for improved performance

Try training again if a first try did not generate good results or you require marginal improvement.

Increase network size if retraining did not help.

Not working? You may need to use a larger data set.

Optionally perform additonal tests

Inputs: IrradianceValue31_transposed

Samples are: ○ Matrix columns ● Matrix rows

Inputs 'IrradianceValue31_transposed' is a 31x11 matrix, representing static data: 31 samples of 11 elements.

Click an improvement button, plot, or click [Next]

Figure 3.11: Evaluate network

3.4.3 Command Line Function

Command line function is an alternative way of clustering. A simple script can be generated from the Neural Network Clustering tool for further modification to customize network training. A user friendly code generated is shown in Appendix B1.

3.5 K-Means Clustering

K-Means clustering is used in both Matlab and Weka software. It separates dataset into clusters with minimum linkage between the clusters. Then, it computes the index of the cluster to which it has assigned each observation. K-Means clustering works on actual observations rather than the larger set of dissimilarity measures, and creates a single level of clusters. Each observation in the dataset is treated as an object having a particular location in space. Then, it searches for a partition in which objects within each cluster are as close to each other as possible, and as far from objects in other clusters as possible. Each cluster in the space can be related to its own input vectors and centroid, or better known as center. The centroid for each cluster is the point to which the sum of distances from all objects in that cluster is the lowest making them as compact as possible. K-Means computes cluster centroids differently for each distance measure to lower down the sum in view with the specified measure. Details of the minimization can be altered using several optional input parameters to K-Means. This covers the ones for the initial values of the cluster centroids, and for the maximum number of repetition. Random initial centroid is used in both software in a feat to acquire highest similarity in the outcome.

In this project, the repetitive K-Means algorithm used can be manifested with the steps below.

I. Determine initial centroid for each cluster

II. Assign multivariate data points to their respective cluster. A data point belongs to the cluster that reflects the lowest sum of distance between the initial centroid and the data point itself.

$$i = \arg\min_{k=1,\ldots,k} ||X_j - M_k||^2 \tag{3.1}$$

Where i = assignment of cluster

Where X_j = multivariate data point

Where M_k = initial centroid

Where ||.|| = Euclidean distance metric

III. Relocate better centroid's position

$$M_i = \frac{1}{|C_i|} \sum_{X_j \in C_i} X_j \tag{3.2}$$

Where M_i = new centroid

Where C_i = cluster

Where X_j = multivariate data point

IV. If the centroid position changes, step II is repeated and followed by step III. The process only stops when the centroid position remains unchanged.

Table 3.1 shows a number of distance metrics for K-Means clustering. The distance metric is switchable manually depending on the user and the nature of the dataset. Distance metric selection is essential as it greatly impacts the outcome of clustering. Outcome consistency varies from one metric to another. Thus, repetition is a must while garnering result. 'sqeuclidean' and 'cityblock' distance metrics are selected in this project [42].

Table 3.1: Distance metric

Distance metric	**Description**
'sqeuclidean;	Squared Euclidean distance (default). Each centroid represents mean of the points in that cluster.
'cityblock'	Sum of absolute differences. Each centroid is the component-wise median of the points in that cluster.
'cosine'	One minus the cosine of the included angle between points (treated as vectors). Each centroid is the mean of the points in that cluster, after normalization of those points to unit Euclidean length.

‘correlation’	One deducts the sample correlation between points (treated as sequences of values). Each centroid is the component-wise mean of the points in that cluster, after centering and normalizing those points to zero mean and unit standard deviation.

3.5.1 Cluster Number Validation Method

After clustering process, technique or verification used can alter the outcome of the result. Multiple verification methods are used in this project to increase the precision of the final outcome. The verification methods are described below in a step by step manner.

3.5.1.1 Silhouette Method

Visualization can be tough in manual classification. Silhouette value is a parameter showing the correctness of the cluster number selected. The silhouette value for each point is a measure of how similar that point is to points in its own cluster, when compared to points in other clusters [43]. The assumption involved in the silhouette value computation is as follows.

I. With the assumption that a data point, i is within cluster A while cluster B is the nearest to cluster A, the silhouette index can be computed.

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)} & if\ a(i) < b(i) \\ 0 & if\ a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1 & if\ a(i) > b(i) \end{cases} \quad , -1 \ll s(i) \ll 1 \tag{3.3}$$

Where $s(i)$ = silhouette index or silhouette value

Where $a(i)$ = mean distance between data point i and all other data points in the same cluster A

Where $b(i)$ = mean distance between data point i and all other data points nearest to the cluster A

A high silhouette value indicates accurate selection of cluster number. Setting the number of cluster manually, the code is constructed as seen in Appendix B2. The silhouette value for each cluster may vary slightly causing complication to the user. Thus, implementing a simple code for mean calculation is ideal.

3.5.1.2 Elbow Method

As ambiguous as it is, K-Means requires setting of a range of cluster number. Sum of squared error(SSE) is the sum of the squared differences between each data point and its cluster's mean. The equation below computes the total SSE for any cluster number selected. Setting a range of number of cluster in K-Means clustering, the SSE for each cluster is calculated and totaled up [44].

$$\mathrm{SSE_{total}} = \sum_{\mathrm{i=1}}^{\mathrm{k}} \sum_{\mathrm{X_j \in C_i}} ||\mathrm{X_j} - \mathrm{m_i}||^2 \tag{3.4}$$

Where $\mathrm{SSE_{total}}$ = the total of sum of squared error in each cluster

Where $\mathrm{X_j}$ = data point in each cluster respectively

Where $\mathrm{m_i}$ = mean distance of cluster

Elbow method is implemented on SSE graph with their respective number of cluster. The point where the SSE begins to stabilize after a sharp plunge is identified as the elbow point. The elbow point is then selected as the appropriate number of cluster.

3.5.1.3 Gap Statistics Method

As a statistical testing method, gap statistics works by comparing data points against null hypothesis. Setting a range of probable cluster numbers, the gap curve portrays the gap value for each cluster number choice along with their one standard error. To identify the ideal cluster number choice, the steps are in the following sequence below.

i. Choose the global maximum point on the gap curve.

ii. Look for the lowest point pertaining to the lower cluster number choice that falls within the one standard error of the chosen global maximum point.

iii. The cluster number choice is determined from the global maximum point if step ii is not applicable.

Keeping in mind that the dataset is in the form of matrix with row, i representing the observation and column, j representing the variable. x_{ij} is used to represent data point in the matrix. With each row representing a single observation, the principle of the equations introduced focuses on the distance between each observation. The distance metric, on the other hand, can be selected manually. The equations along with the explanations are described in this section with steps. The sequence of the gap statistics analysis is as follows:

i. Cluster the observation data with K-Means.

ii. Compute the $\log(W_k)$ for each cluster of observation data with the equation below

$$D_r = \sum_{i,i' \in C_r} d_{ii'} \tag{3.5}$$

Where D_r = Sum of the pairwise distances for all data points in cluster r

Where C_r = Cluster r

Where $d_{ii'}$= distance between data point i and data point i'

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r \tag{3.6}$$

Where k = Number of cluster

Where W_k = Pooled within-cluster sum of square around the cluster mean or within dispersion measure of observation data

Where n_r= Number of data point in cluster r

iii. Generate reference data (Monte Carlo sample) from reference distribution

iv. Cluster the reference data with K-Means

v. Compute the within dispersion measure, $\log(W_{kb}^{*})$ for the reference data.

vi. Repeat step iii and v for B times to acquire the total of $\log(W_{kb}^{*})$ before computing the average by dividing the total with B.

$$Gap(k) = \left(\frac{1}{B}\right)\sum_{b} \log(W_{kb}^{*}) - \log W_k \tag{3.7}$$

Where $Gap(k)$ = Gap between reference datasets and observation datasets

Where B = Number of data sets generated from the reference distribution in the form of positive integer value

Where b = The dataset generated from the reference distribution up to B (b = 1, 2, …, B)

Where k = Number of cluster

Where W_{kb}^{*}= Pooled within-cluster sum of square around the cluster mean or within dispersion measure of reference data ($k = 1, 2, \ldots, K, b = 1, 2, \ldots, B$)

vii. Compute One Standard Error, s_{k+1} with the formula below

$$s_k = \left[\left(\frac{1}{B}\right)\sum_b \left\{\log(W_{kb}^*) - \left(\frac{1}{B}\right)\sum_b \log(W_{kb}^*)\right\}^2\right]^{\frac{1}{2}} \sqrt{\left(1 + \frac{1}{B}\right)} \tag{3.8}$$

viii. Finally, select the optimum cluster number based on the requirement below

$$Gap(k) \gg \ Gap(k+1) - \ s_{k+1} \tag{3.9}$$

3.6 Summary

Overall, this chapter describes the methods used to acquire clustering result in both Matlab software and Weka software. In preliminary clustering, only Matlab is used. Weka is only used in final clustering to better assure the final result. The variation in distance metric is discussed. Procedures and methods used to compute the results are shown in detail along with appropriate explanations and figures. In Weka, outliers will be traced before clustering process. K-Means is utilized in this software and the verification method involves Elbow analysis and Silhouette analysis. On the other hand, in Matlab, after acquiring dataset without outlier from Weka, K-means clustering takes place. This is followed by verification method that uses Silhouette analysis and Gap statistics analysis. The outcome of these two software will be analyzed to reach a final conclusion. The verification methods are ranked based on their performance in real life. From researches, Gap statistics has the highest priority. This is then followed by Elbow method and finally, Silhouette method. The results generated will be shown and discussed in the next chapter.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Overview

This chapter intends to unravel the hidden pattern in the results obtained and bring out the similarity, as well as the dissimilarity for manual classification. Crosscheck, being the main comparison will lead to a logical conclusion. In preliminary clustering, SOM and K-Means clustering are applied in Matlab software. In final part, larger dataset is used to better identify the number of cluster and Weka software is introduced in this part along with the existing Matlab software.

4.2 Preliminary Clustering Result

One month’s dataset is used in this preliminary clustering. Being able to view the result in a bigger picture, the pattern of the result can illustrate the behavior of each data point. The significance of each data point is high and can possibly alter the final result.

4.2.1 Self Organizing Map Result

It is of paramount importance that comparison must be made with logical thinking. However, interpretation varies from one person to another. Hence, a series of comparison is analyzed to support the final conclusion. It is known that darker color represents farther distance for the Neighbor Weight Distance plot. Decision making is based on assumptions in Table 4.1 below.

Table 4.1: Assumption for self organizing map

Assumption	Action
A plot without apparent pattern correlates with stochastic noise.	Only examine plot with distinctive boundary.
Extremely low cluster number indicates insufficient neuron number.	Increase the number of neuron.
Extremely high cluster number indicates excessive neuron.	Decrease the number of neuron.
A small isolated cluster is an outlier.	Ignore the outlier.

Setting the number of neuron to 100, the largest apparent clusters (in yellow) are close to each other at upper right corner. The lighter color of these cluster signifies that points in each cluster are closely packed. The darker region acts as a border separating these clusters from one another. The darker the region, the farther the distance. Looking at Figure 4.1, it is safe to assume that the number of cluster is 3.

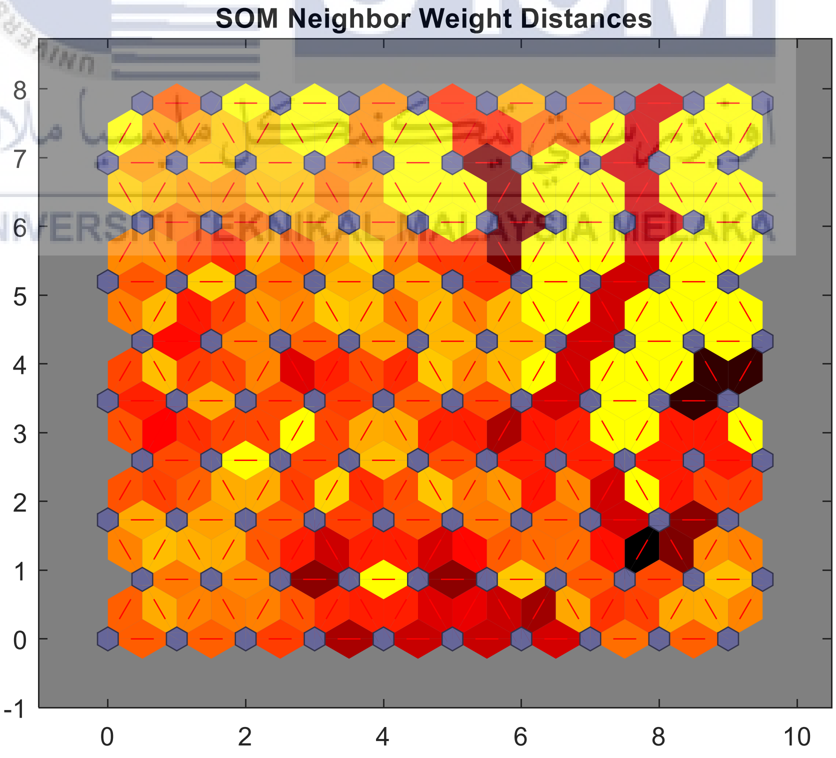


Figure 4.1: Neighbor weight distance plot for 100 neurons

Setting the number of neuron to 225, multiple clusters appear due to heightened sensitivity. Two data vectors can be put into different cluster even if they differ by a small fraction. Having a big difference, neuron number of 225 is still chosen due to its stability. Multiple tests have been run for a range of neuron number and the results are not as appealing as 225 neuron number. The result is deemed stable if the pattern persists after multiple tests. Figure 4.2 shown below possesses a considerably high stability and therefore, it is included in the section.

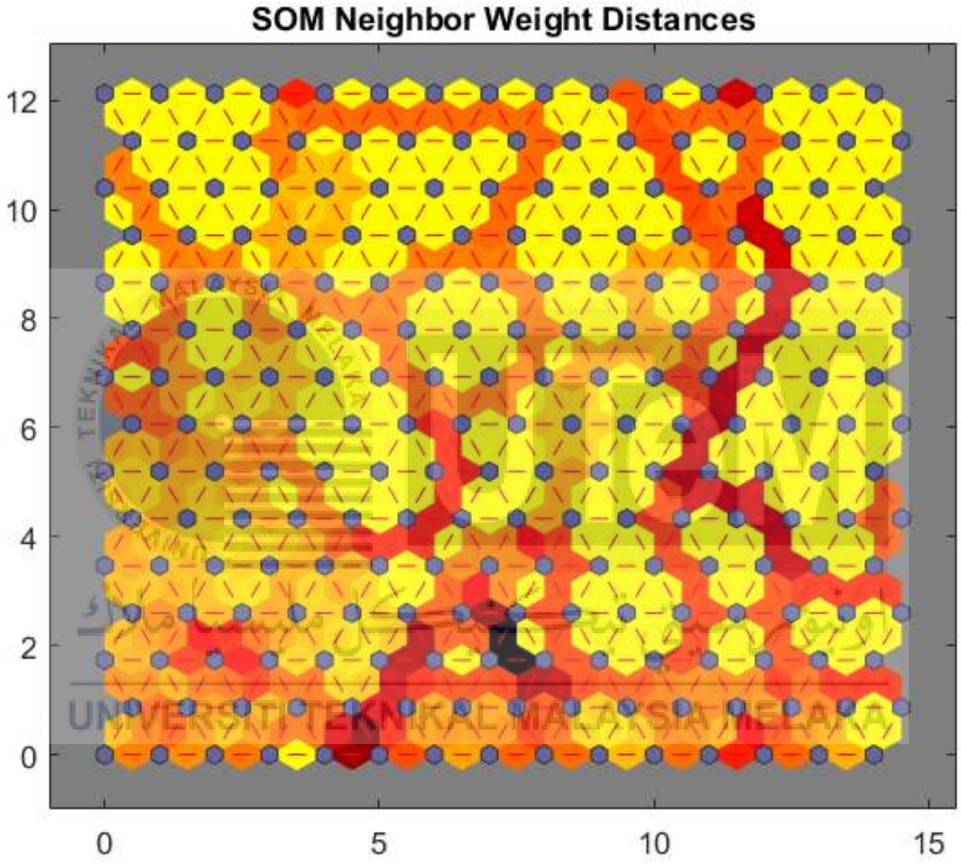


Figure 4.2: Neighbor weight distance plot for 225 neurons

Small region with light color, oftentimes, indicates the presence of outlier. Outlier can be categorized as a bad data as it deviates way too much from the rest. Outlier is a group containing very few data vector. Logically, a group of same species tend to have high similarity. This applies to the case of outlier, being the only group with significantly large difference as compared to the rest. Thus, outlier can be known as an outcast. After a proper analysis, a conclusion is made as shown in Table 4.2.

Table 4.2: Cluster analysis

Neuron number	Apparent cluster	Outlier
100	3	7
225	12	14

SOM is known for its dimension reduction ability. A successful analysis will decrease the computation time while maintaining the same accuracy. Input plane portrays each variable in their respective pattern. The color intensity is directly proportional to the density of the data vector in a particular region on the plot. Having a different terminology than the previous plots, a dark region represents a densely packed data points. Taking 9 a.m. to 7 p.m. data, eleven variables can be plotted with each representing a different hour. To reduce the dimension means dumping some variables while keeping the influential ones. Maintaining the neuron number, by examining plot in Figure 4.3 and Figure 4.4, variables with same pattern are drawn out for comparison. By doing a crosscheck on the common patterns. The overlapping variables can be determined.

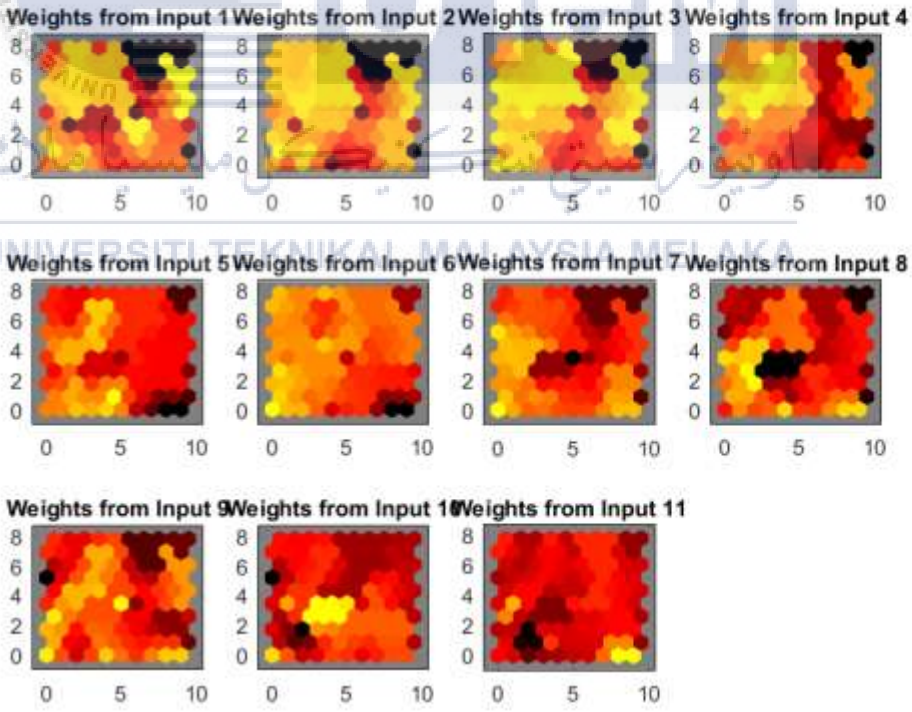


Figure 4.3: Input plane plot for 100 neurons

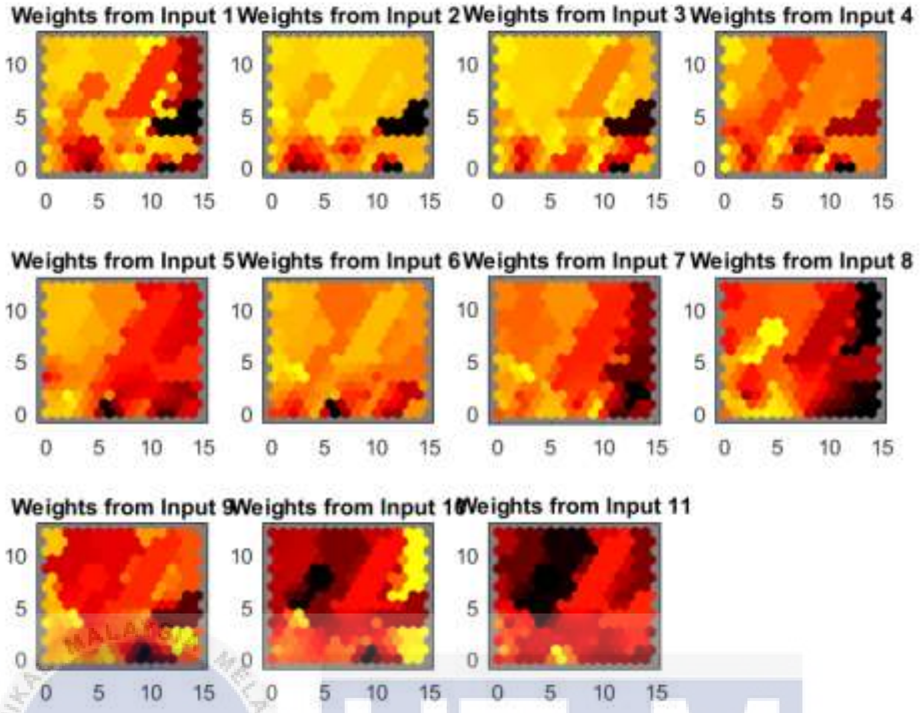


Figure 4.4: Input plane plot for 225 neurons

Table 4.3: Dimension reduction analysis

Neuron number	Common input plane	Overlapping
100	5 and 6	5 and 6
225	2 and 3, 5 and 6	

Based on analysis in Table 4.3, input plane 5 and input plane 6 of 100 neurons have high similarity. Similarly, input plane 2 and input plane 3, Input plane 5 and input plane 6 of 225 neurons possess the same trait. Overlapping means a double certainty. It is safe to deduce that the input plane 5 or input plane 6 is negligible, leading to a one dimensional reduction.

Moving forward to sample hit plot, a clear cut number of cluster can be seen after changing the neuron number to 4 and 9 respectively as shown in the figure below. Hexagon with larger dark blue region signifies a larger number of point in that space. The number of data vector hitting the region is tabulated in the hexagon itself. By comparison, plot in Figure 4.5 and Figure 4.6 show a stronger connection by assuming the number of cluster as 3.

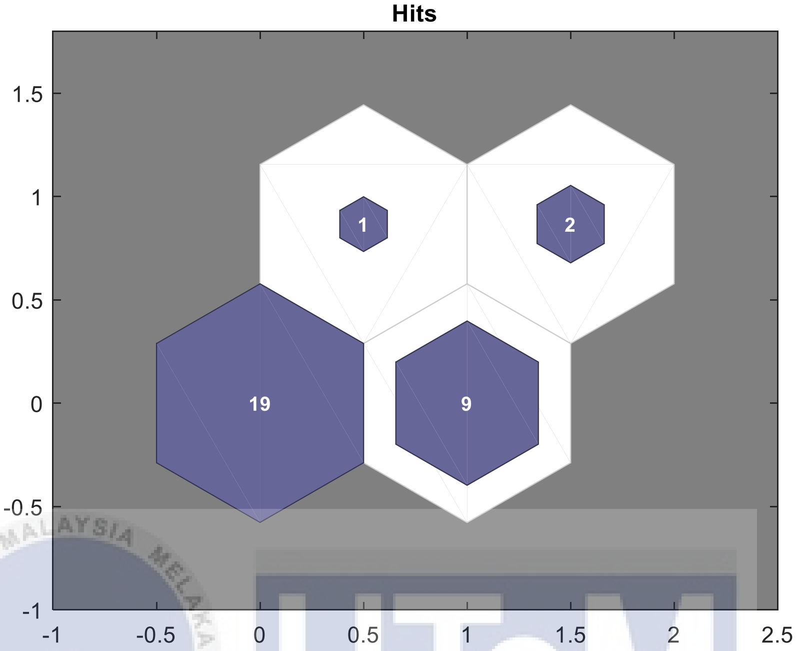


Figure 4.5: Sample hit plot for 4 neurons

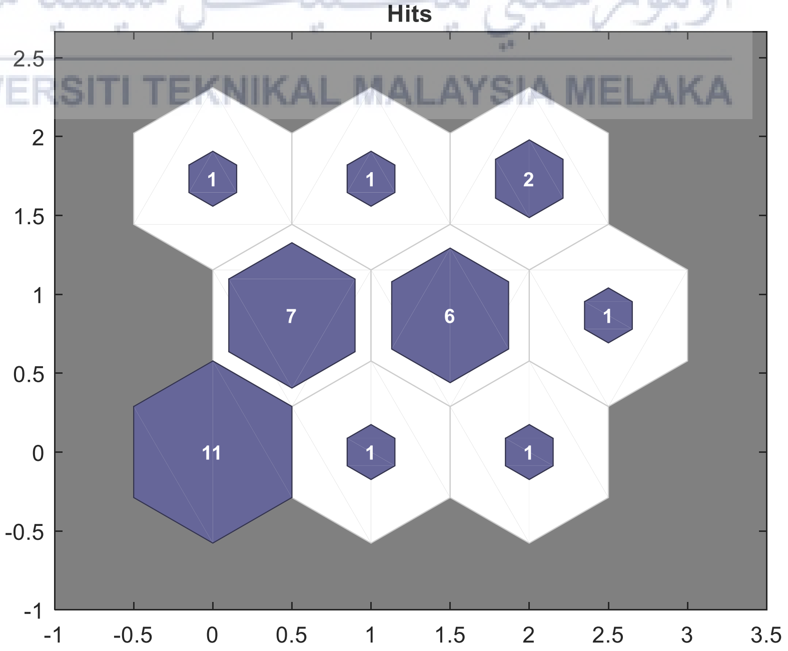


Figure 4.6: Sample hit plot for 9 neurons

Table 4.4: Sample hit analysis

Neuron	Cluster	Outlier
4	3	1
9	3	6

Sample hit plot further validates the exactness of cluster number as stated in Table 4.4. An overall hypothesis for Self Organizing Map result, 3 clusters and 10 dimension, can be made with high confidence.

4.2.2 K-Means Clustering

Silhouette plot shows the silhouette value that each cluster holds. A high and positive silhouette value dictates a good number of cluster chosen whereas a low and negative silhouette value proves the otherwise. Without a proper knowledge on the right number of cluster, it is wise to experiment with multiple number choices. Silhouette value can top at 1 at max. Figure 4.7, Figure 4.8, Figure 4.9 and Figure 4.10 are shown below for comparison.

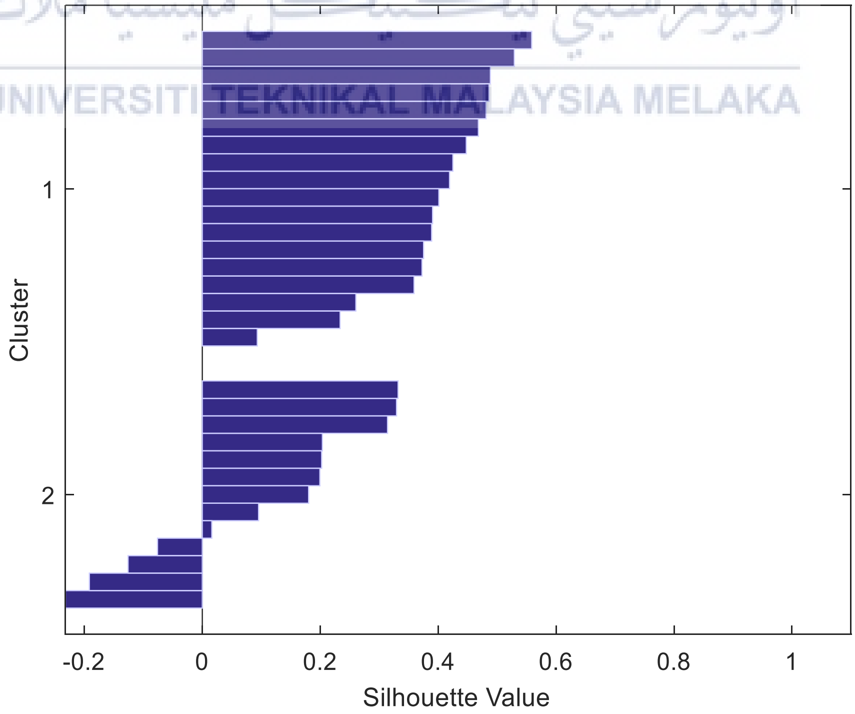


Figure 4.7: Silhouette plot for 2 clusters

In Figure 4.7, it can be seen that four data points in cluster 2 have negative values. This simply means that the four data points belong to other clusters other than cluster 1. Most of the data point shows high relevance in cluster 1 with the lowest silhouette at 0.1.

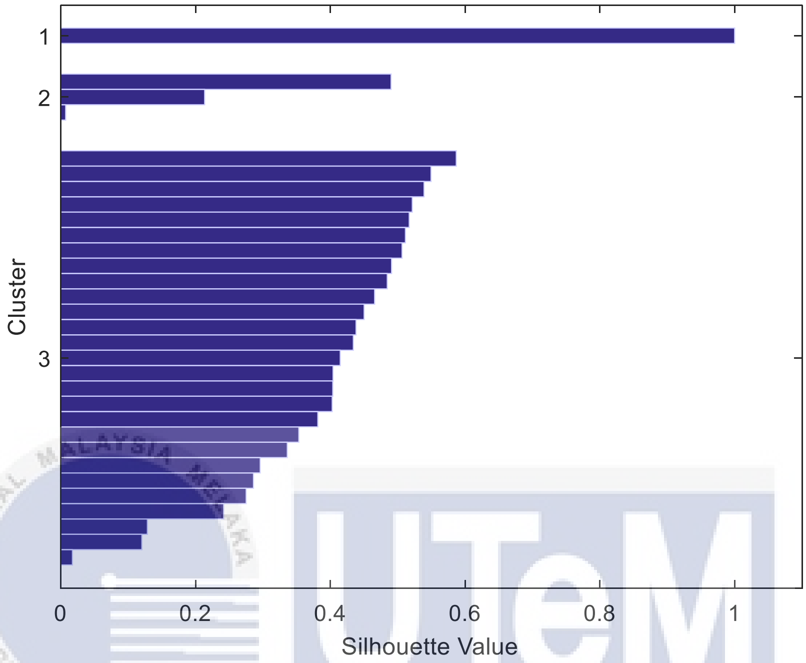


Figure 4.8: Silhouette plot for 3 clusters

In Figure 4.8, cluster 3 has the most data points with the lowest silhouette value at approximately 0.025. Cluster 1 shows the highest silhouette value at 1. However, having only one data point in cluster 1 is not satisfactory.

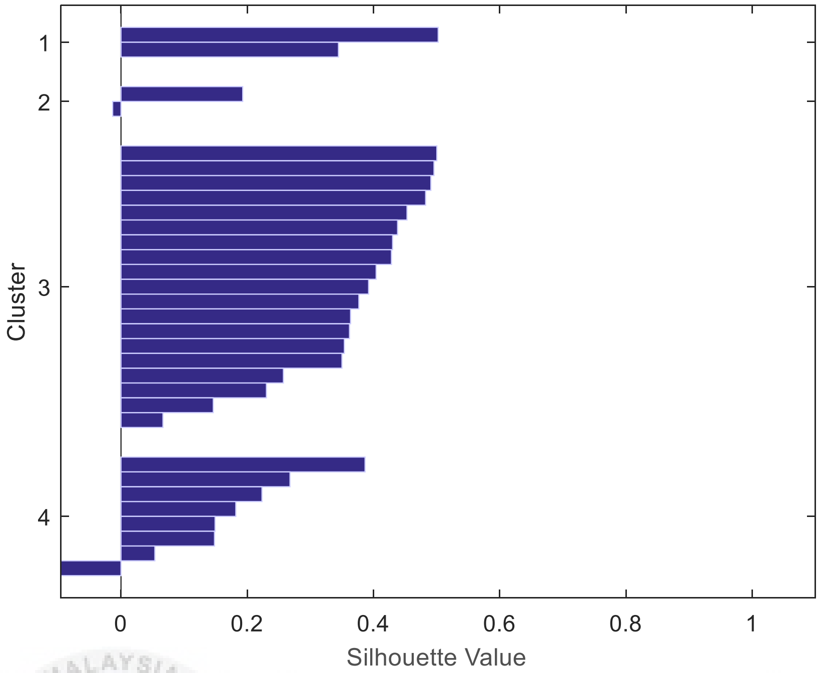


Figure 4.9: Silhouette plot for 4 clusters

Splitting the grouping further, cluster 3 shows no sign of negative silhouette value as seen in Figure 4.9. However, the decrease in data point gives rise to more deviations. The withdrawal of data points in cluster 3 suggests a possibility in category expansion. Having a significantly high negative silhouette value restricts the idea of category expansion. Cluster 2 has a data point with negligible negative silhouette. Meanwhile, cluster 4 contains a data point with high negative silhouette.

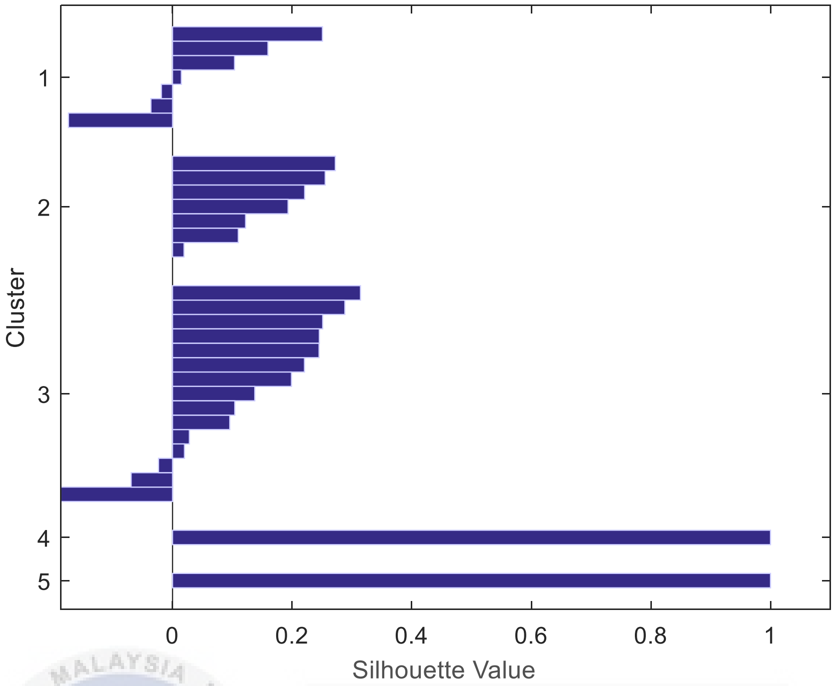


Figure 4.10: Silhouette plot for 5 clusters

Setting the cluster number to 5, both cluster 4 and cluster 5 have the highest positive silhouette value as seen in Figure 4.10. This shows that both data points in cluster 4 and cluster 5 are independent. They vary vastly with other data points and may possibly signify the presence of outlier. However, multiple negative silhouette values in cluster 1 and cluster 3 make the cluster choice of 5 invalid.

From the silhouette plots generated, it can be deduced that 3 clusters plot has a more appealing trend as the silhouette value of each point in those clusters is comparatively higher. This indicates that the clusters are somewhat, more far apart from each other. To validate the assumption, by using a more quantitative way, average silhouette value can be computed for each cluster number for comparison.

Table 4.5: Average silhouette value

Cluster number	Average silhouette value
2	0.2718
3	0.3958
4	0.3021
5	0.1730

As seen in Table 4.5, cluster 3 has the highest average silhouette value as compared to other cluster number, thereby, signifying the 3 clusters as an ideal choice. By looking at SOM and K-Means result, a conclusion can be drawn for preliminary clustering. Both methods show the same outcome, 3 clusters. However, the evidence is not concrete due to the less amount of data.

4.3 Final Clustering Result

Shifting the focus to only K-Means, Matlab and Weka softwares are used in this final clustering with larger dataset. Outliers are identified with Weka visualization tool before proceeding with clustering process.

4.3.1 Clustering in Weka

In this part, after conversion of data file to ARFF format, outlier detection is properly assessed before clustering process. Elbow analysis and silhouette analysis are the verification methods used to identify the optimum number of cluster in the end.

4.3.1.1 Evaluation of Outlier Removal

The irradiance profile as seen in Table 4.6 below displays the feature of a regular clear sky. The hourly irradiance peaks at 1p.m and is perfectly normal. However, being the only day with excellent output, it falls at the far end of the trend. This data point tends to create another cluster of its own, thereby producing more noise. If the location of this data point is perceived as considerably close by the clustering algorithm, the data point will be included as a part of a cluster. This instance would create an outcome with high SSE even when the supposedly low SSE number of cluster is selected. Hence, data points that stand out from the rest must be eliminated to produce precise result. As a result, data point of day 83 is excluded from the dataset.

Table 4.6: Irradiance profile of day 83 in year 2014

Hour	Irradiance(Wm^{-2})
9 a.m	325.4
10 a.m	581.3
11 a.m	798.3
12 p.m	912.3
1 p.m	931.8
2 p.m	1058.0
3 p.m	926.1
4 p.m	804.0
5 p.m	608.8
6 p.m	286.9
7 p.m	35.6

From Figure 4.11, Figure 4.12 and Figure 4.13, the outliers captured are in red circle. The data point in circle appears to have the same light blue color as the majority nearest to it. This makes it a noise that contributes to higher SSE. Three hourly irradiance profiles portray the same trait for day 83. Thus, day 83 is removed from the 2014 dataset.

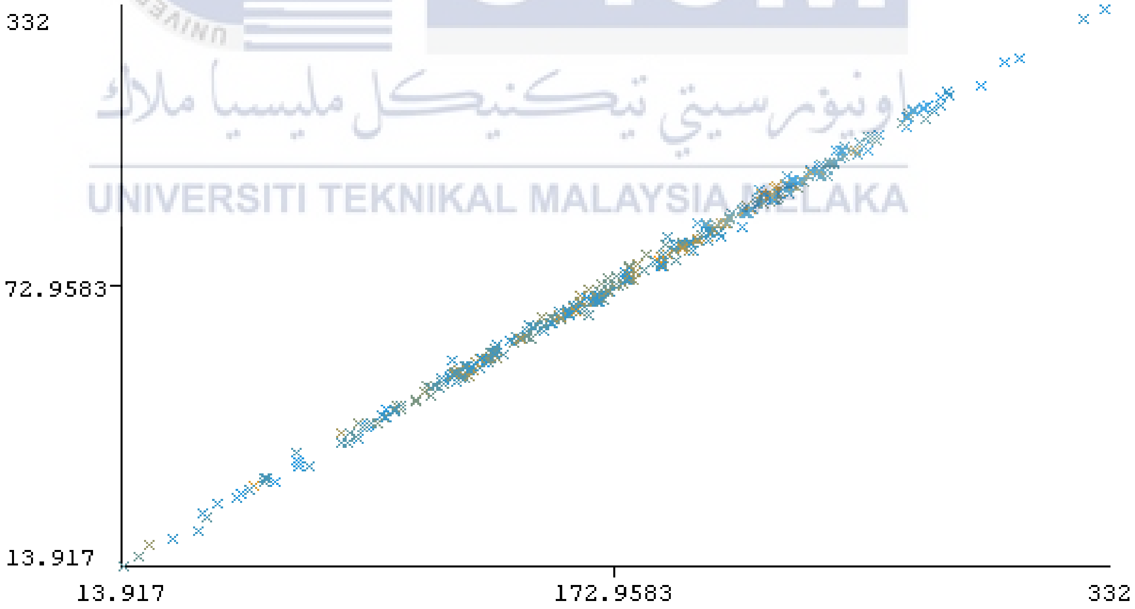


Figure 4.11: Visualization of 9 a.m irradiance profile of day 83 in year 2014

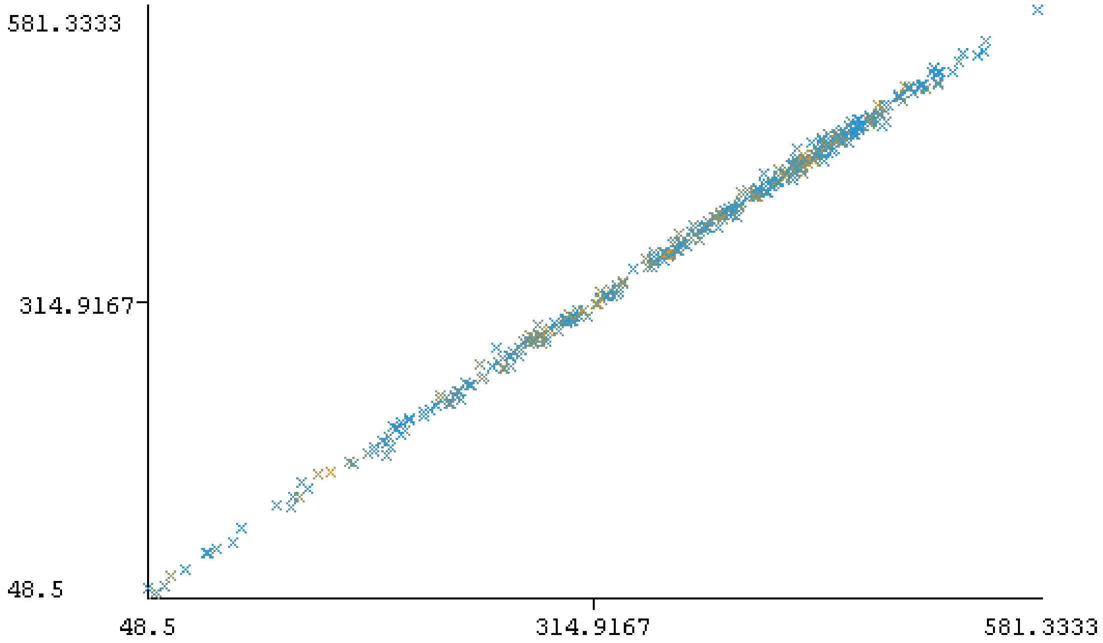


Figure 4.12: Visualization of 10a.m irradiance profile of day 83 in year 2014

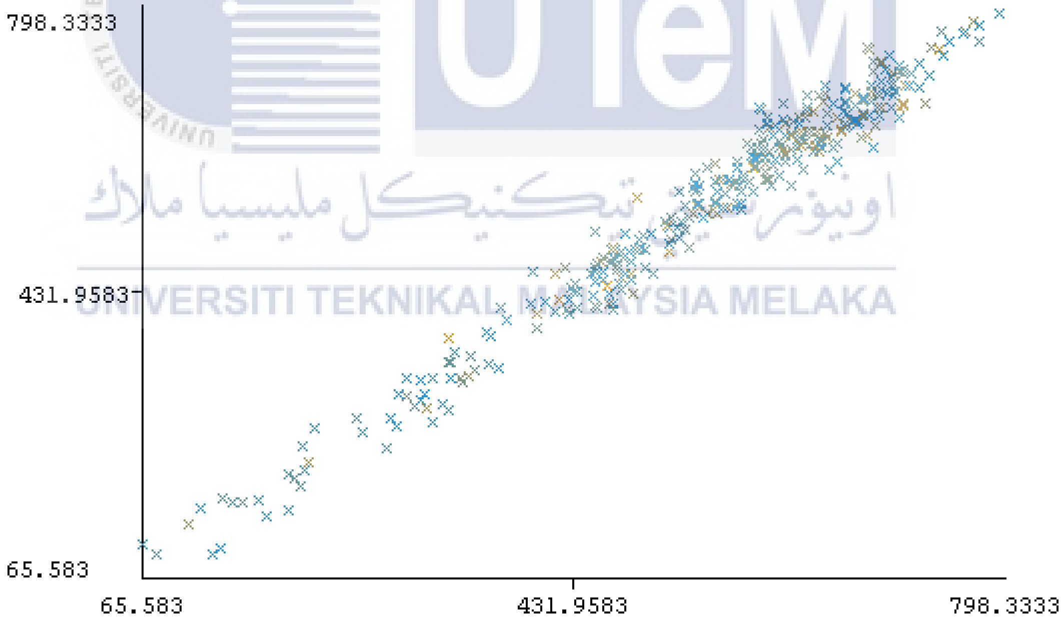


Figure 4.13: Visualization of 11a.m irradiance profile of day 83 in year 2014

In the same year, while analyzing the visualization graph only, day 327 shows two occurrences of noise inducing outlier in hourly irradiance profile based on Figure 4.14 and Figure 4.15. The data points in red circle are so far apart from the rest despite having the same light blue color. However, referring to the daily profile as tabulated in Table 4.7, six instances can be found to have the same irradiance value. This trend starts at 12p.m and ends at 6p.m. Hence, this data point is ruled out as a faulty data.

Table 4.7: Irradiance profile of day 327 in year 2014

Hour	Irradiance(Wm^{-2})
9 a.m :	254.9
10 a.m :	528.3
11 a.m :	759.3
12 p.m :	894.5
1 p.m :	879.0
2 p.m :	879.0
3 p.m :	879.0
4 p.m :	879.0
5 p.m :	879.0
6 p.m :	879.0
7 p.m :	72.6

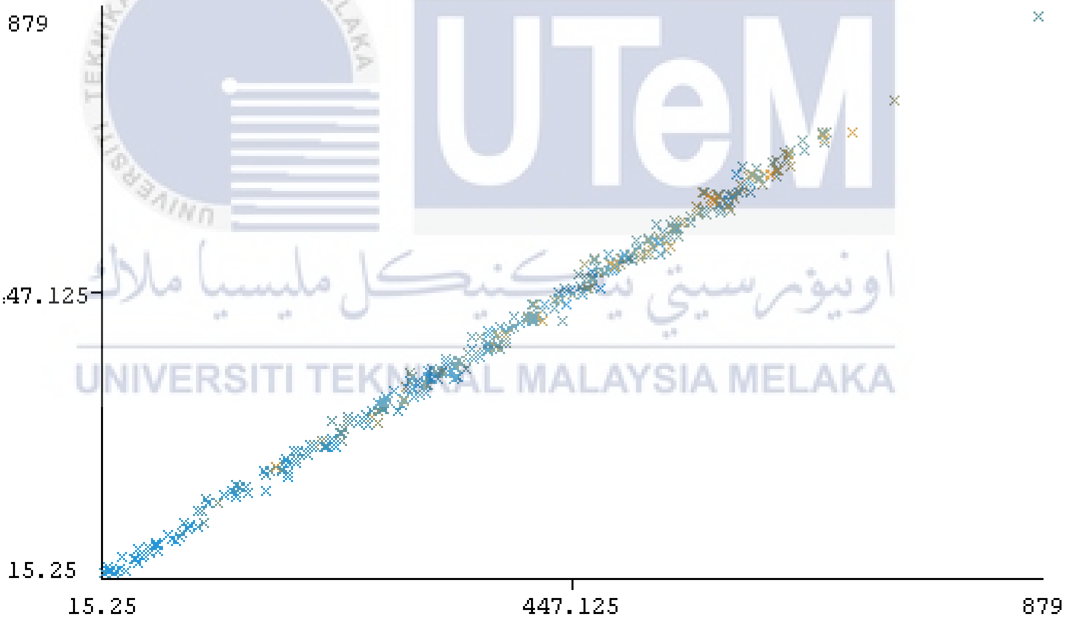


Figure 4.14: Visualization of 5p.m irradiance profile of day 327 in year 2014

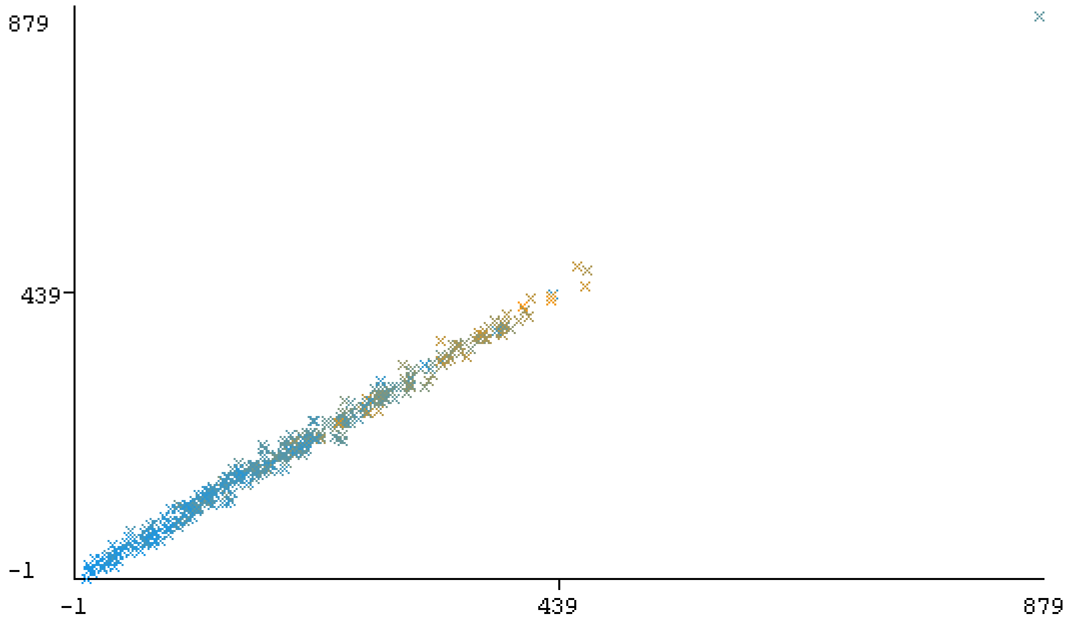


Figure 4.15: Visualization of 6p.m irradiance profile of day 327 in year 2014

In year 2015, as seen in Table 4.8 below, 3p.m to 7p.m has the same irradiance value of 1208. Having normal trend up to 2p.m only, this data point is not a logical feature of typical output as the irradiance remains high until evening. By looking through Figure 4.16, Figure 4.17, Figure 4.18, Figure 4.19 and Figure 4.20, the circled data point is red color. The difference in color between the data points denotes different cluster. Being the only data point with red color, it suggests that the data point is in a complete different class of its own. This finding, too, indicates high probability that the data point might be introduced as the only data point in its cluster. Therefore, day 50 of year 2015 is eliminated from the dataset.

Table 4.8: Irradiance profile of day 50 in year 2015

Hour	Irradiance(Wm^{-2})
9 a.m	65.4
10 a.m	169.2
11 a.m	293.1
12 p.m	427.0
1 p.m	746.5
2 p.m	1120.0
3 p.m	1208.0
4 p.m	1208.0
5 p.m	1208.0
6 p.m	1208.0

Hour	Irradiance(Wm^{-2})
7 p.m	1208.0

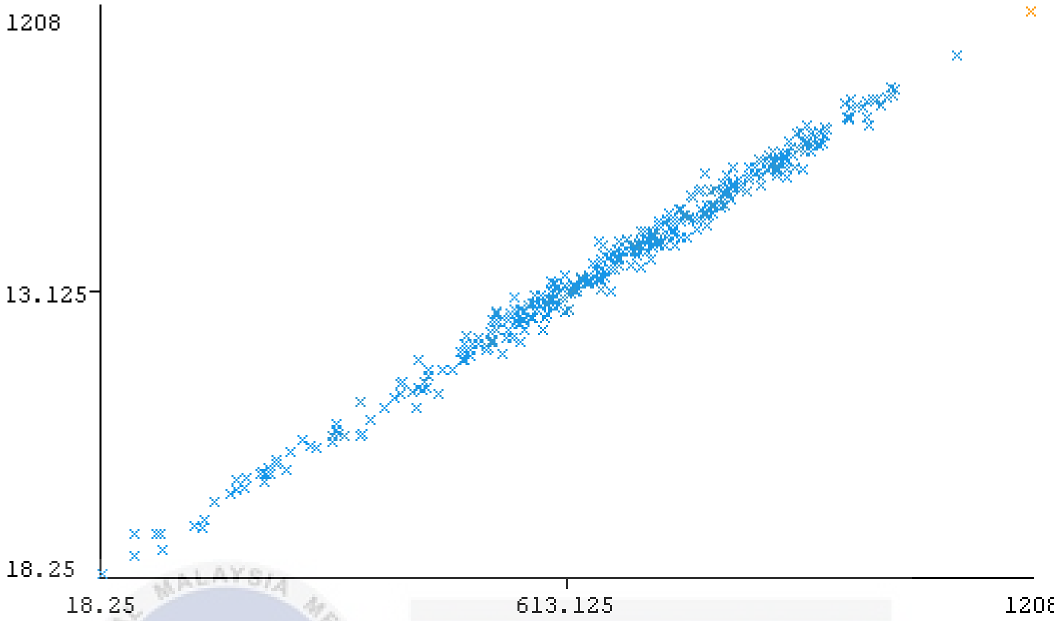


Figure 4.16: Visualization of 3p.m irradiance profile of day 50 in year 2015

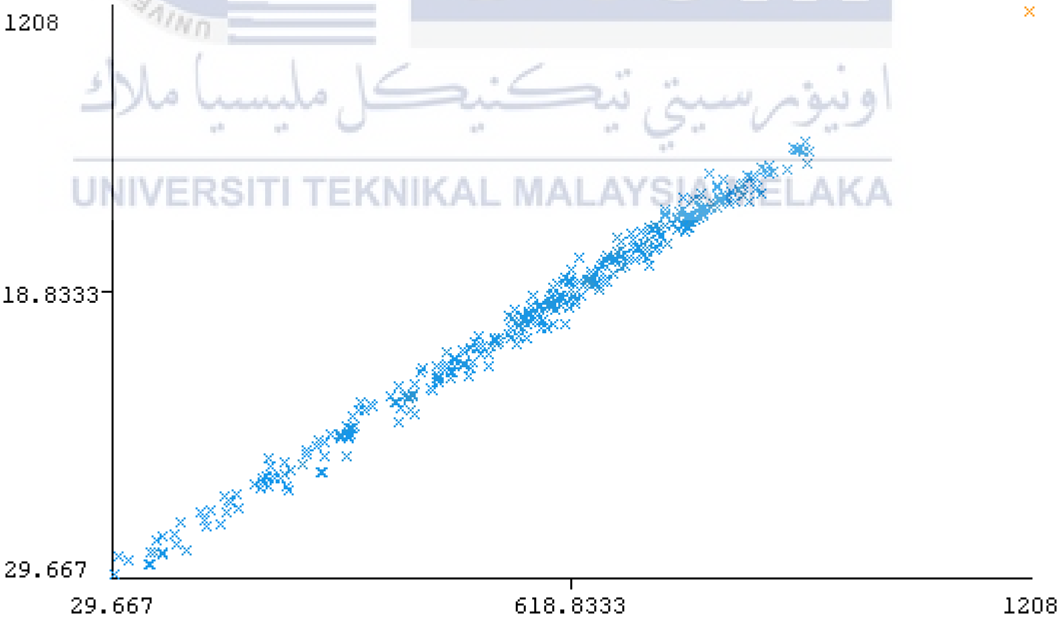


Figure 4.17: Visualization of 4p.m irradiance profile of day 50 in year 2015

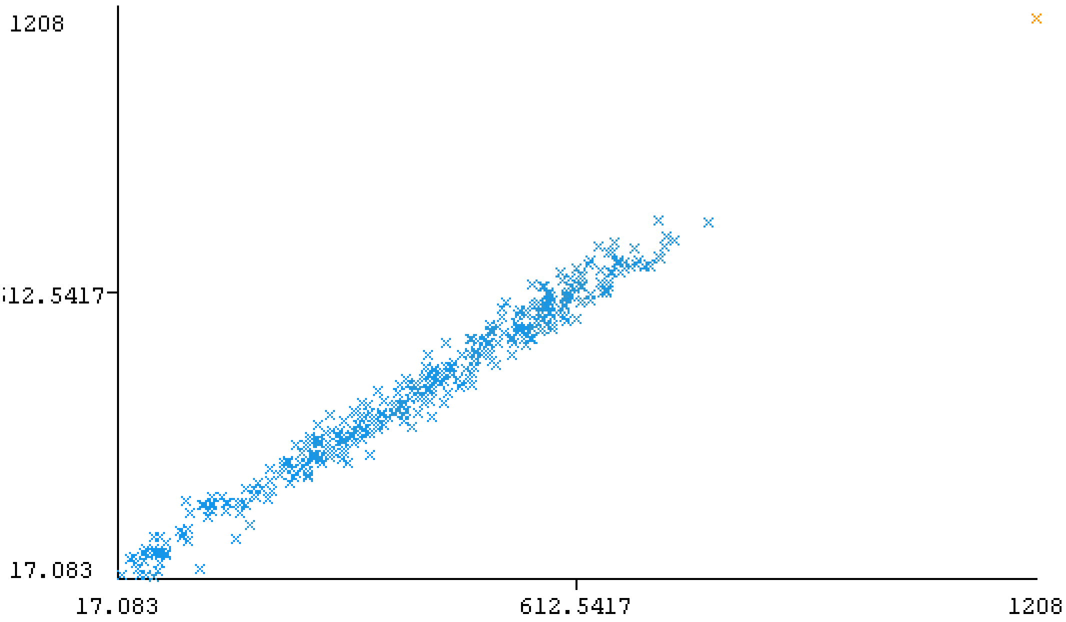


Figure 4.18: Visualization of 5p.m irradiance profile of day 50 in year 2015

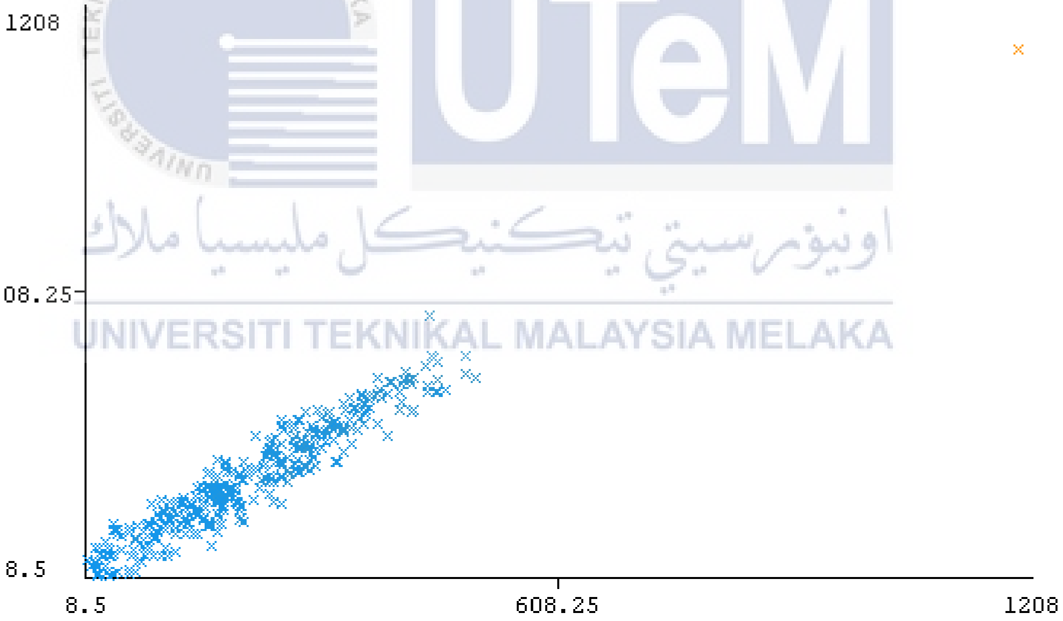


Figure 4.19: Visualization of 6p.m irradiance profile of day 50 in year 2015

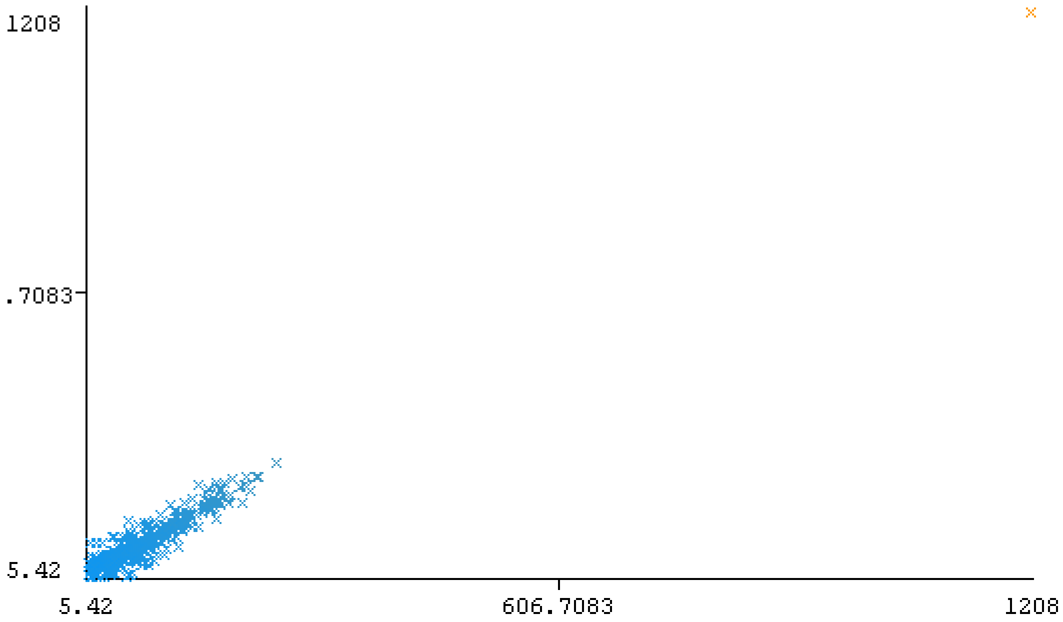


Figure 4.20: Visualization of 7p.m irradiance profile of day 50 in year 2015

In year 2016, all data points appear to have negligible deviation. Visualization of hourly irradiance profile of day 209 is shown in Figure 4.21. The circled data point is far away from the rest of the data points of the same class. However, based on Table 4.9, other variables, ranging from 9a.m to 6p.m, do not deviate from the majority. Besides, 7p.m has the least significance as compared to other variables. Hence, day 209 is an exception and it remains in the dataset.

Table 4.9: Irradiance profile of day 209 in year 2016

Hour	Irradiance(Wm^{-2})
9 a.m	129.583333
10 a.m	375.583333
11 a.m	620.583333
12 p.m	761.166667
1 p.m	532.166667
2 p.m	767.333333
3 p.m	778.833333
4 p.m	753.666667
5 p.m	632.166667
6 p.m	440.75
7 p.m	265.0

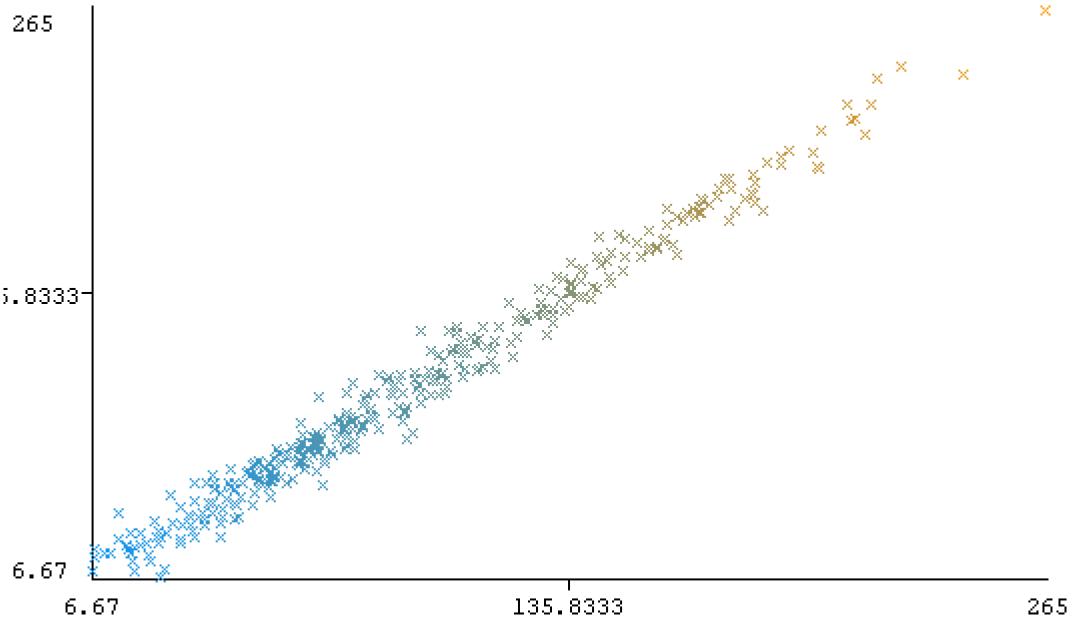


Figure 4.21: Visualization of 7p.m irradiance profile of day 209 in year 2016

4.3.1.2 K-Means Clustering Result in Weka

For each year, the zoomed in version of elbow analysis graph is included to ease the process of verification. In year 2014, the SSE only begins to stabilize after reaching 3 clusters (K=3). As seen in both Figure 4.22 and Figure 4.23, after a significantly high gradient from K=2 to K=3, the trend maintains at the same gradient from K=3 to K=6. The point where the sudden change of gradient takes place is known as elbow. Hence, the suitable number of cluster in year 2014 is three (K=3). Meanwhile, silhouette analysis shows different number of cluster. In Figure 4.24, the silhouette value tops at K=2. This is followed by K=3 at second place. The trend fluctuates slightly between K=4 and K=10 making that range improper. The mean distance between all the data points and centroid in each cluster is the lowest when the K selection is two. However, the difference in silhouette value between K=2 and K=3 is questionable. Thus, the possible outcome could be K=2 or K=3.

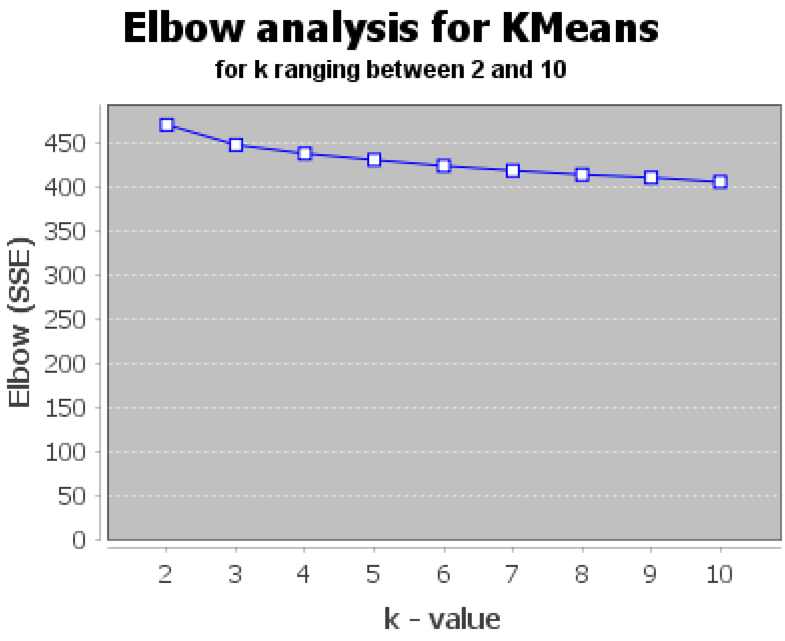


Figure 4.22: Elbow analysis graph for year 2014

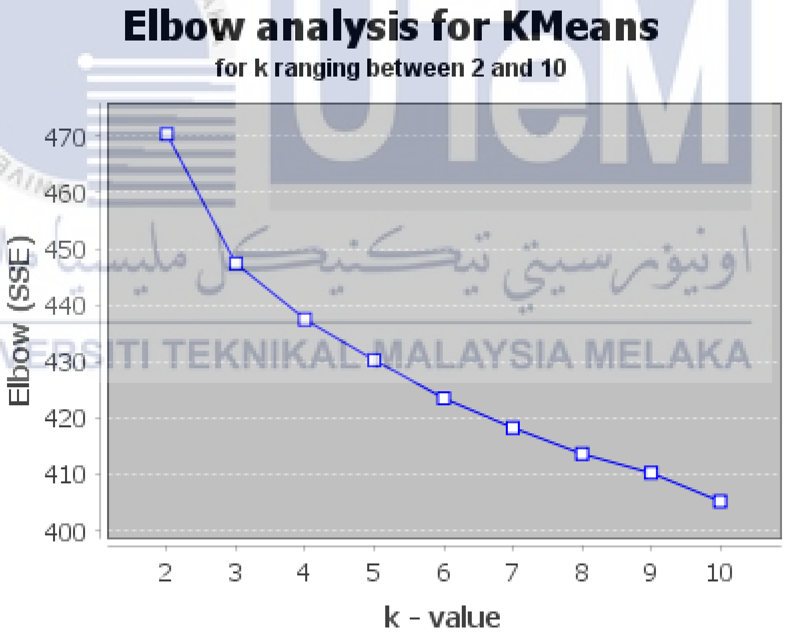


Figure 4.23: Elbow analysis for year 2014 (zoomed-in)

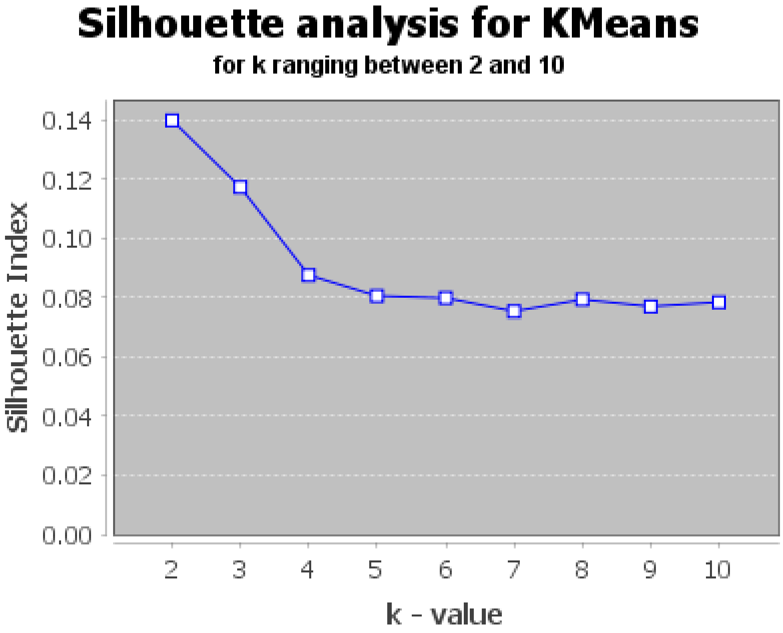


Figure 4.24: Silhouette analysis graph for year 2014

In year 2015, Elbow analysis displays minimum difference in the gradient across the increasing cluster size. Based on Figure 4.25 and Figure 4.26, the trend appears as a curve in the zoomed in graph. The observable change of gradient starts at K=3 making it the possible result. However, more consideration comes into play as the evidence has low integrity. Therefore, the result is subject to ambiguities. Looking at Silhouette analysis in year 2015, as seen in Figure 4.27 below, the trend portrays a somewhat similar trait as compared to silhouette analysis in year 2014. However, the difference in silhouette value between K=2 and K=3 is greater than that in year 2014. K=2 is selected as the proper number of cluster. On the other hand, K=5 has a noteworthy feature as it rises up again after a continuous downward trend.

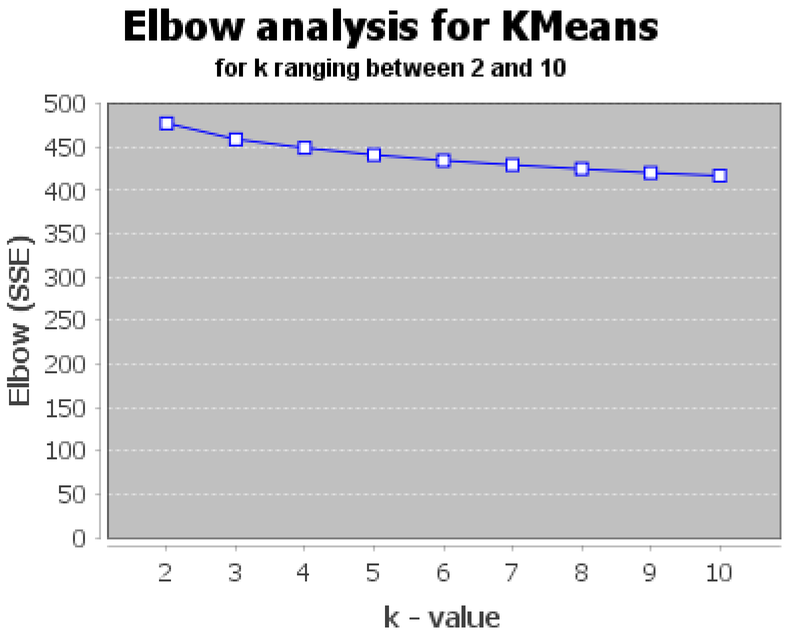


Figure 4.25: Elbow analysis graph for year 2015

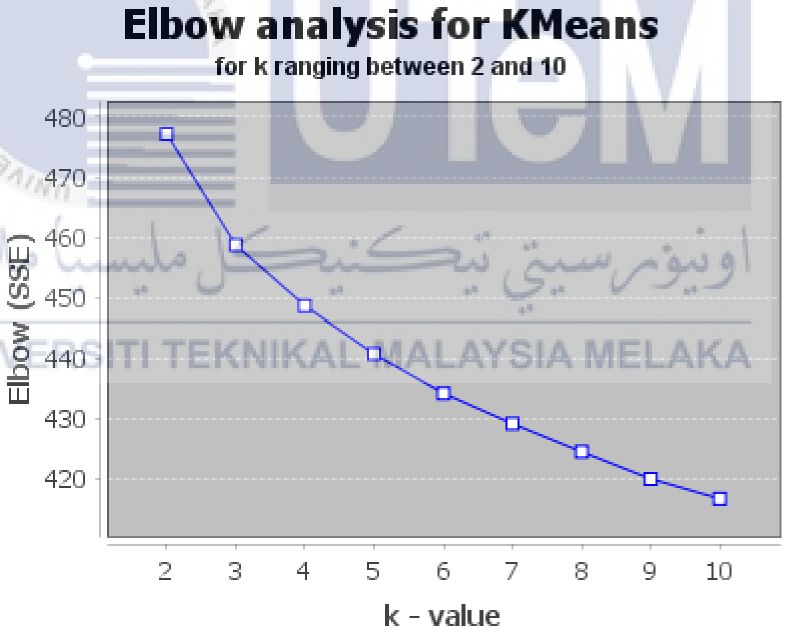


Figure 4.26: Elbow analysis graph for year 2015 (zoomed-in)

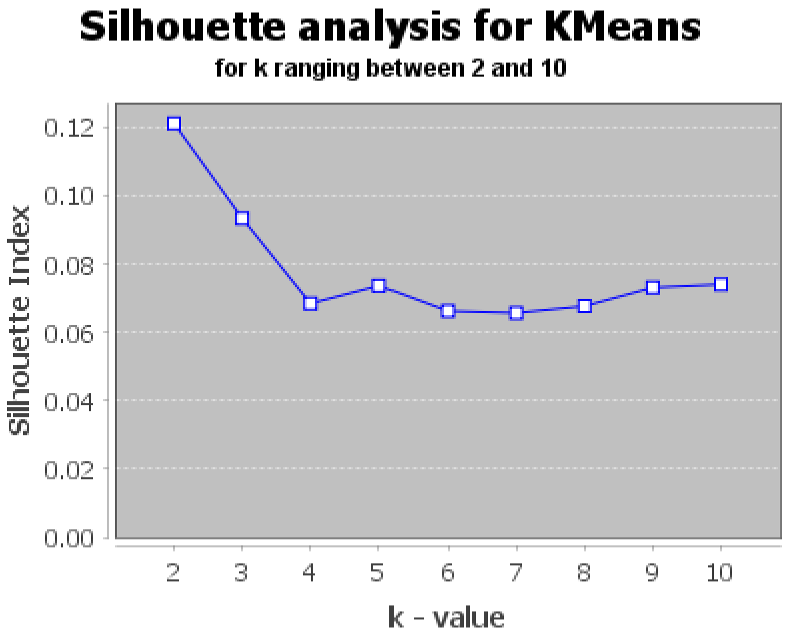


Figure 4.27: Silhouette analysis for year 2015

In year 2016, as seen in Figure 4.28 and Figure 4.29 below, the trend of Elbow analysis graph is almost the same throughout the range of cluster. No apparent difference in gradient can be sighted. The absence of elbow point makes the Elbow analysis unfit for verification in year 2016. The result is therefore, uncertain. In Silhouette analysis, the trend shows a significant difference in silhouette value between K=2 and K=3. In Figure 4.30 below, K=2 is positioned at the highest point with about 0.07 difference in silhouette value as compared to K=3. Hence, K=2 is selected as the outcome.

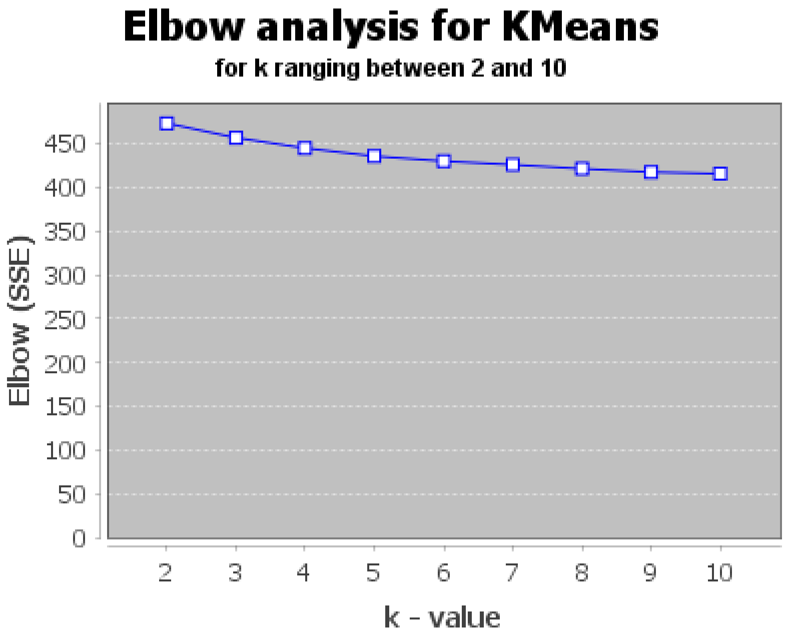


Figure 4.28: Elbow analysis for year 2016

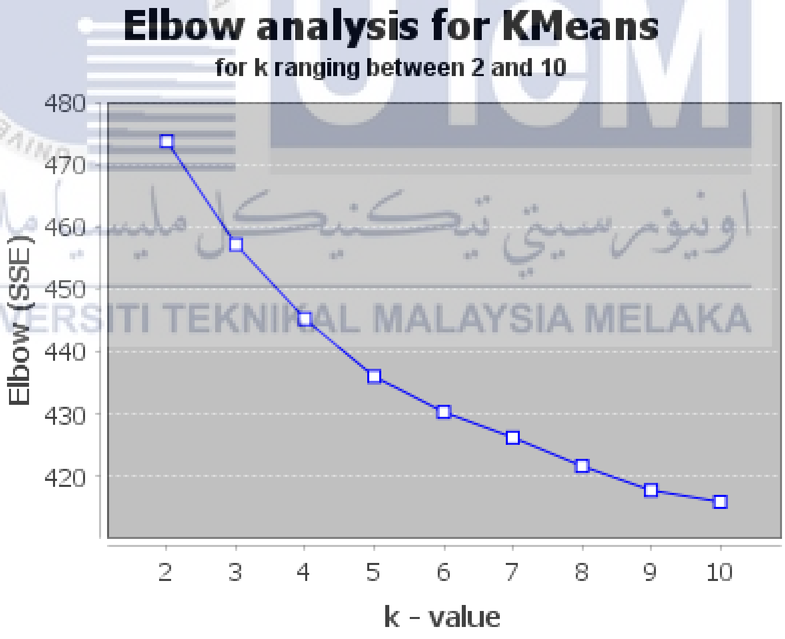


Figure 4.29: Elbow analysis for year 2016 (zoomed-in)

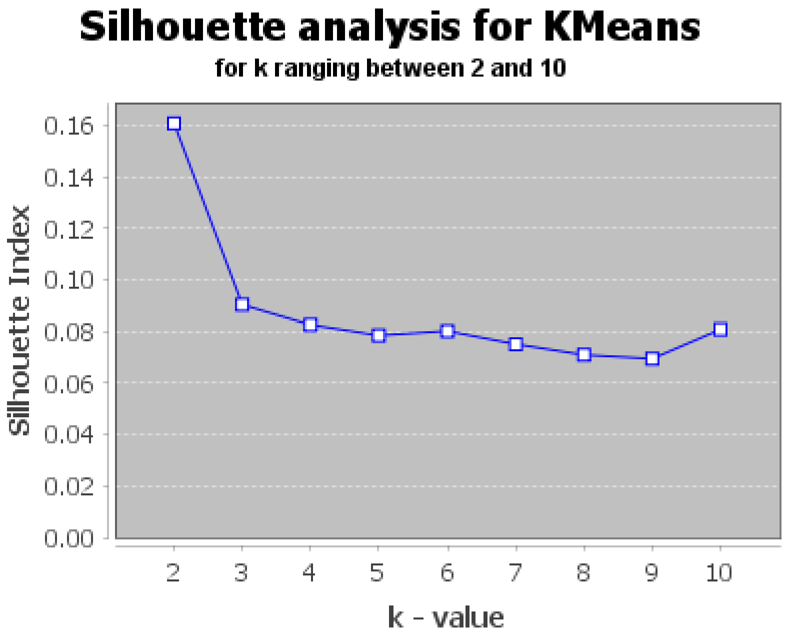


Figure 4.30: Silhouette analysis for year 2016

Data analysis requires big amount of data to generate result with high accuracy. A dataset comprised of data from year 2014, 2105 and 2016 is used for the final clustering in Weka. Analyzing Figure 4.31 and Figure 4.32, the zoomed-in graph shows an elbow point at K=3. Another elbow point can be observed at K=5. However, the gradient difference at K=3 is greater. K=3 is selected as the possible result. In Silhouette analysis, as seen in Figure 4.33 below, the silhouette value peaks at K=2 with notable gap with the point at K=3. K=2 is taken as the outcome.

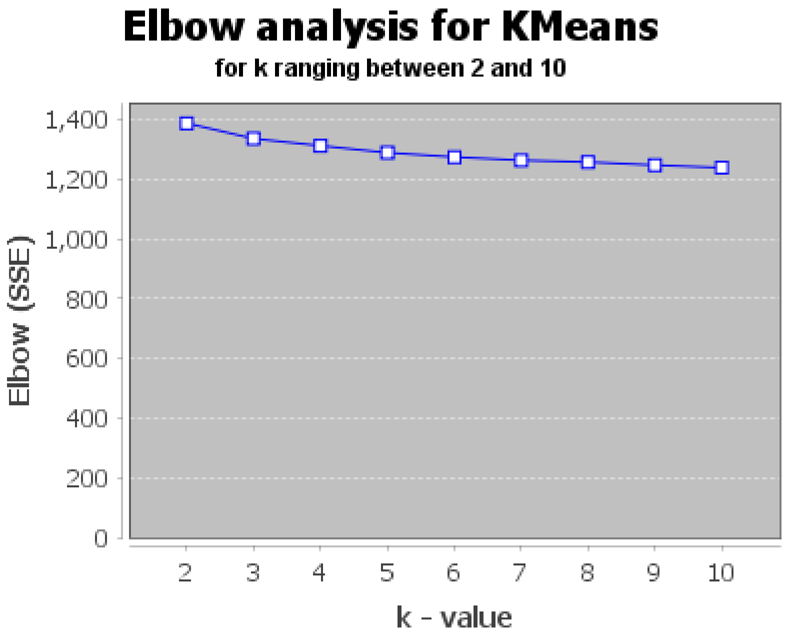


Figure 4.31: Elbow analysis graph for year 2014,2015 and 2016

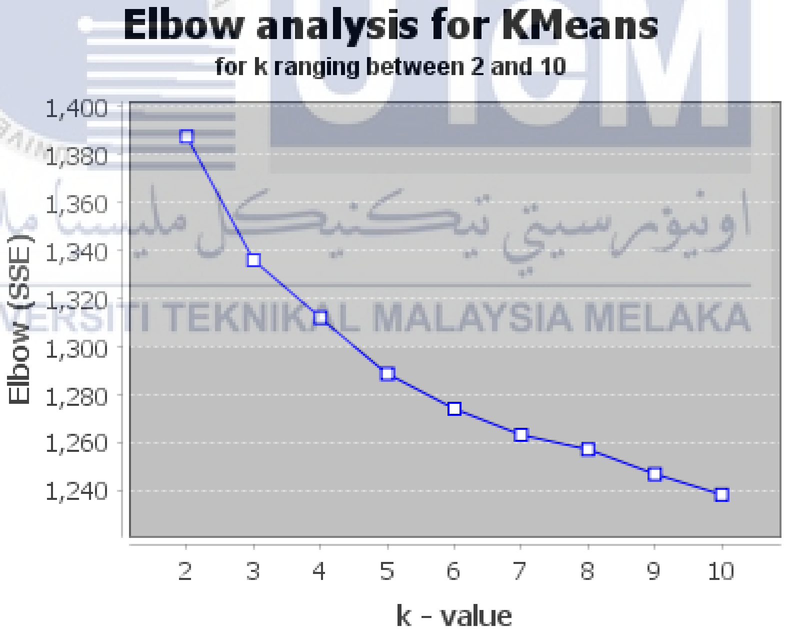


Figure 4.32: Elbow analysis graph for year 2014,2015 and 2016 (zoomed-in)

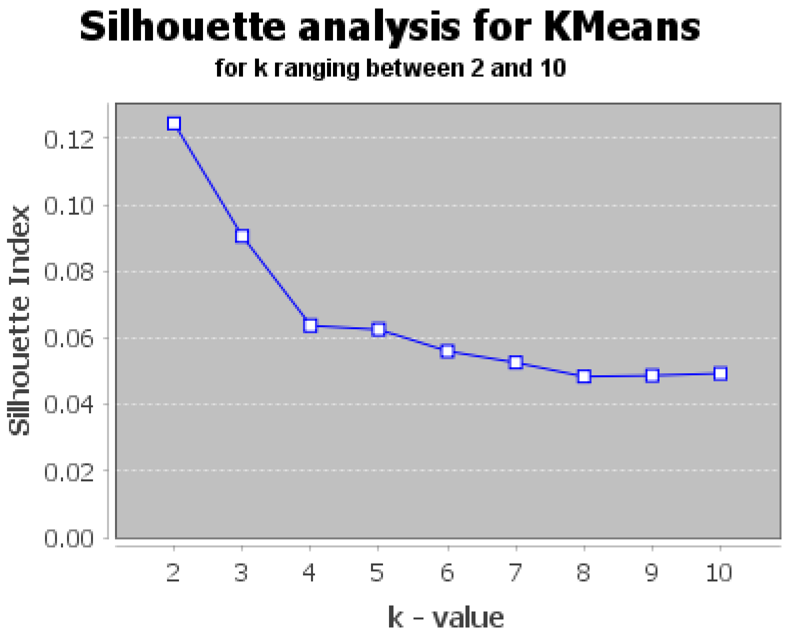


Figure 4.33: Silhouette analysis graph for year 2014, 2015 and 2016

4.3.2 K-Means Clustering in Matlab

In Matlab, the same Silhouette method is used as in preliminary clustering. In year 2014, the mean silhouette value is tabulated in Table 4.10 for evaluation. K=2 has the highest average silhouette value as compared to the rest. Looking at Figure 4.34, cluster 1 has some data points with negative silhouette value. This signifies the possibility that the data points belong to another cluster other than cluster 2. Cluster 2, on the other hand, contains more data point and shows no sign of negative silhouette value. In Figure 4.35, data points with negative silhouette value can be seen at cluster 2 and cluster 3. This situation suggests the possibility that the data points belong to cluster other than cluster 1, cluster2 and cluster 3. K=2 in Figure 4.35 has lesser data point with negative silhouette value compared to K=1 in Figure 4.34. In spite of that, K=2 is still selected as the outcome as it has the highest average silhouette. Gap statistics is a more superior verification method in comparison with both Elbow analysis and Silhouette analysis. The evaluation focuses on the within cluster dispersion of data points. The desirable number of cluster is obtained when the dispersion is minimum. Based on Figure 4.36 below, the gap value at K=3 falls within the allowable range of error (One Standard Error) of K=4. Being the least number of cluster that satisfies the One Standard error, K=3 is selected as the outcome. To investigate further, the point at K=2 is far below the allowable range of error of K=3 making it undesirable.

Table 4.10: Silhouette value for dataset of year 2014

Cluster (K)	Silhouette value
2	0.5367
3	0.3702
4	0.3039
5	0.3261
6	0.3185
7	0.2855

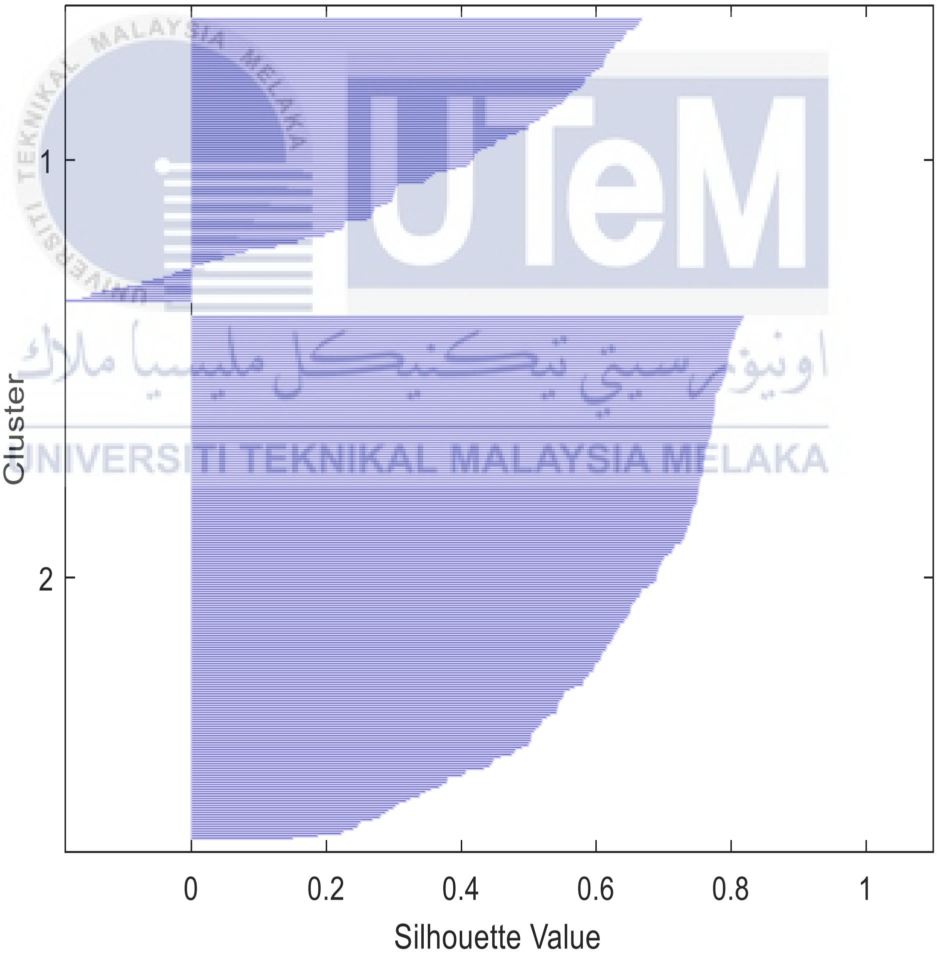


Figure 4.34: Silhouette plot for 2 clusters in year 2014

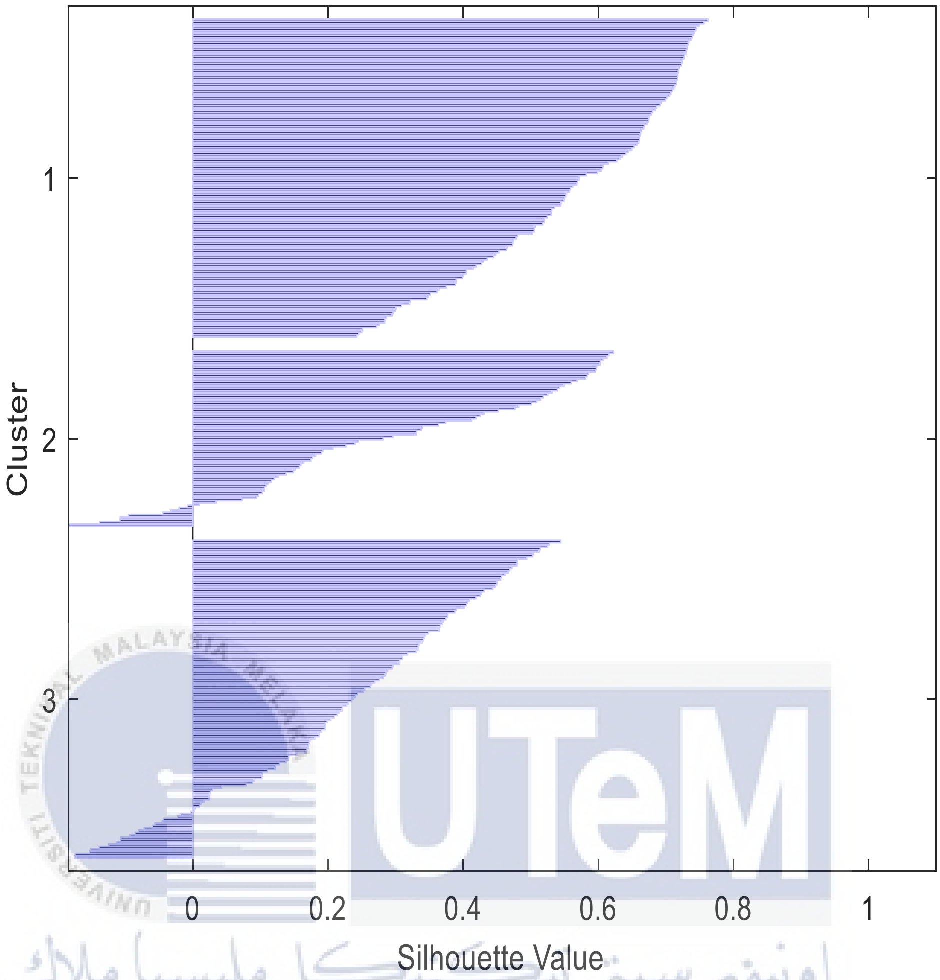


Figure 4.35: Silhouette plot for 3 clusters in year 2014

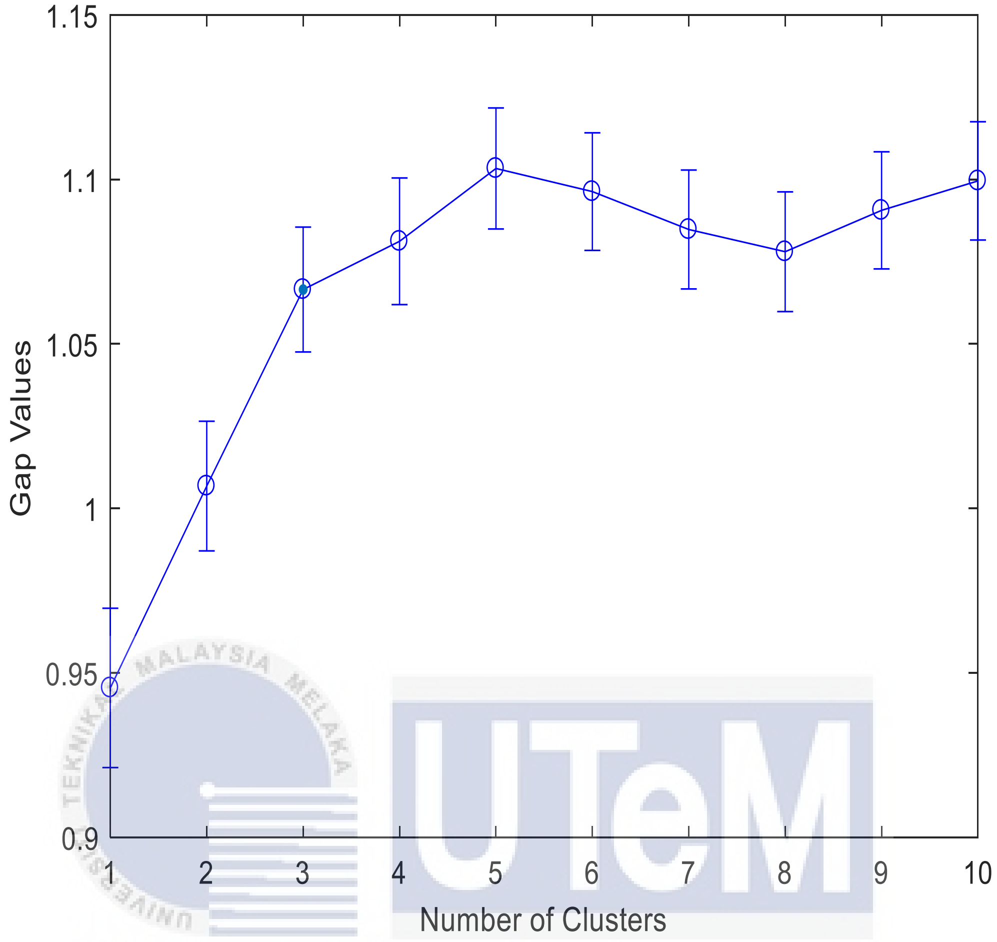


Figure 4.36: Gap statistics analysis in year 2014

In 2015, the highest average silhouette value comes from K=2 as seen in Table 4.11. Being the best choice, the overall negative silhouette value of K=2 in Figure 4.37 is higher than K=3 in Figure 4.38. This drawback suggests that more data points in clusters of K=2 belong to other cluster which is not desirable. The weakness in Silhouette analysis is revealed in this situation where mean silhouette only considers the positive silhouette value while neglecting the negative silhouette value. This analysis focuses more on the exactness of each data point in the cluster. Hence, K=3 is a better choice as it considers both negative and positive side of the graph. In gap analysis, the least number of cluster with gap value that satisfies One Standard Error of the higher number of cluster, K+1 is K=1. As seen in Figure 4.39, the range of allowable error is exceptionally big compared to other Gap statistics analysis graph in this project. Thus, most of the points are within the allowable range of the higher number of cluster.

Table 4.11: Silhouette value for dataset of year 2015

Cluster (K)	Silhouette value
2	0.4784
3	0.3818
4	0.3498
5	0.3449
6	0.2709
7	0.2631

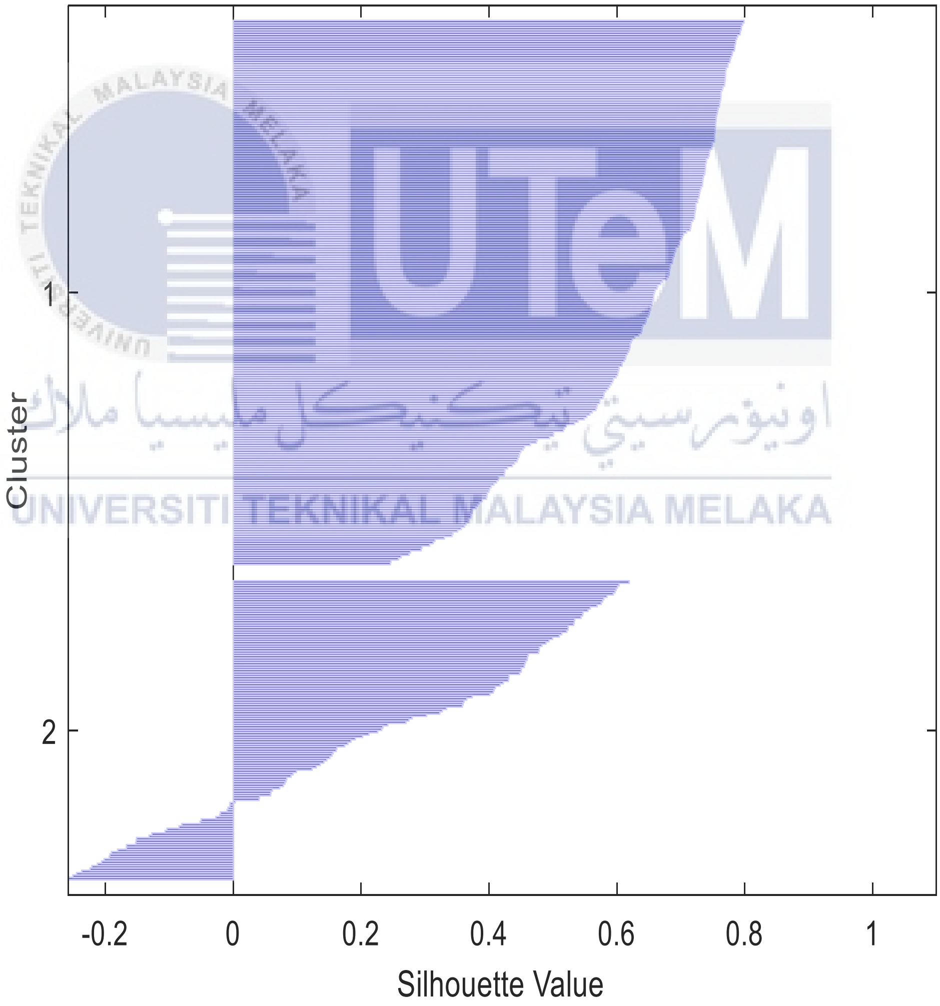


Figure 4.37: Silhouette plot for 2 clusters in year 2015

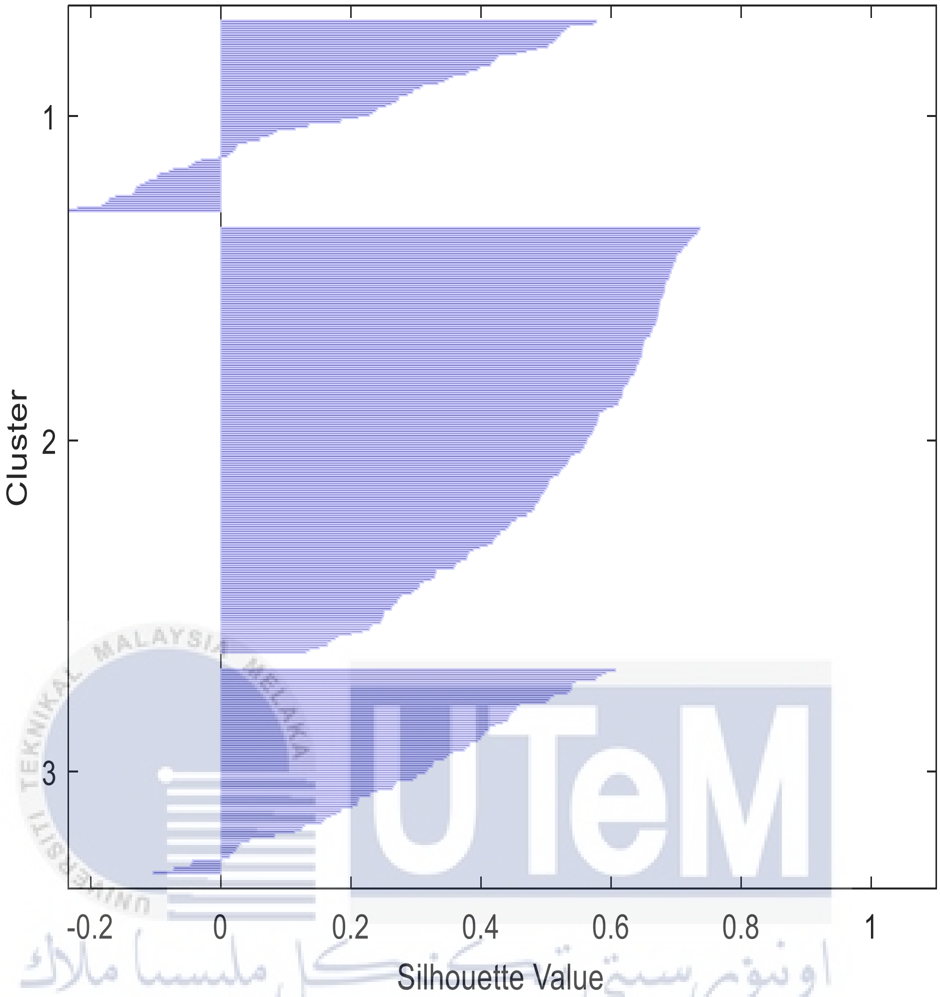


Figure 4.38: Silhouette plot for 3 clusters in year 2015

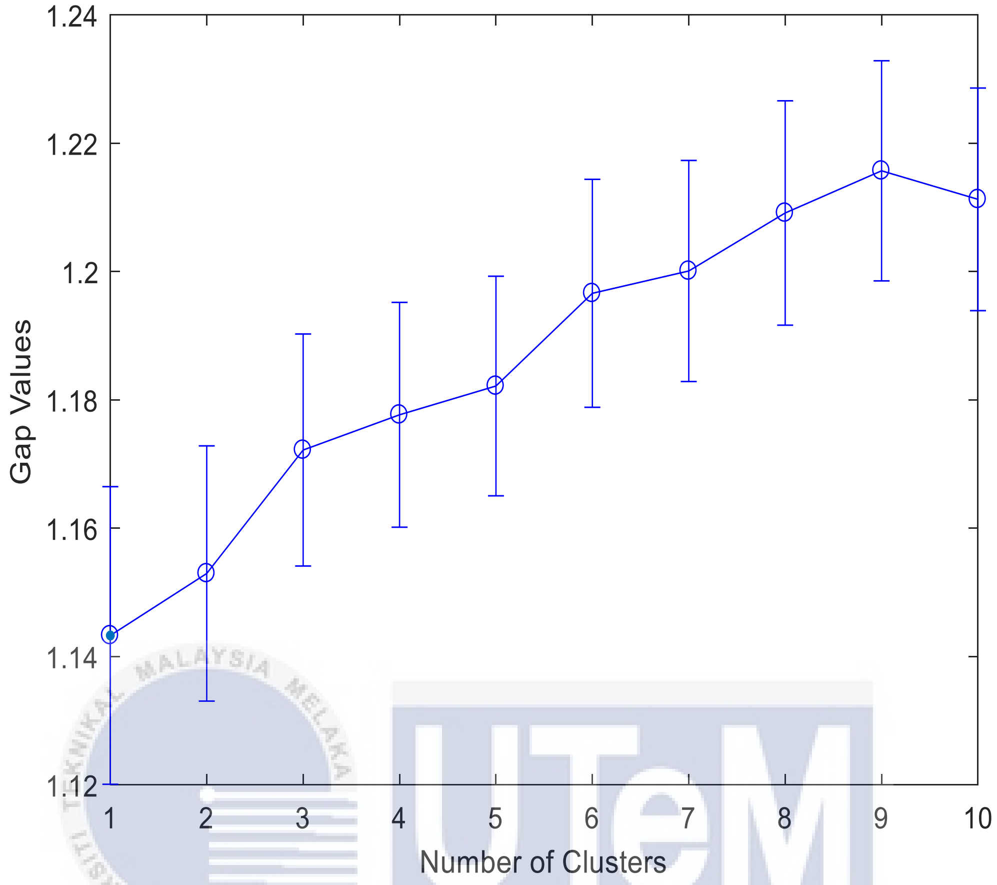


Figure 4.39: Gap statistics analysis in year 2015

In year 2016, K=2 has the highest mean silhouette value as seen in Table 4.12 below. Conducting evaluation of graph in Figure 4.40 and Figure 4.41 below, a clear cut difference can be observed in the negative silhouette side favoring K=2. Cluster 1 in Figure 4.41 has greater negative silhouette value than the overall negative silhouette value in Figure 4.40. Thus, K=2 is selected as the proper result. In gap statistics analysis, the least number of clustering satisfying the One Standard Error is K=3. It is evidently clear that the gap value of K=3 falls within the allowable range of error of K=4. To support the decision, it is apparent that the gap value of K=3 is considerately near to the gap value of K=4 in Figure 4.42.

Table 4.12: Silhouette value for dataset of year 2016

Cluster (K)	Silhouette value
2	0.5646
3	0.3303
4	0.3758

Cluster (K)	Silhouette value
5	0.2837
6	0.3095
7	0.3064

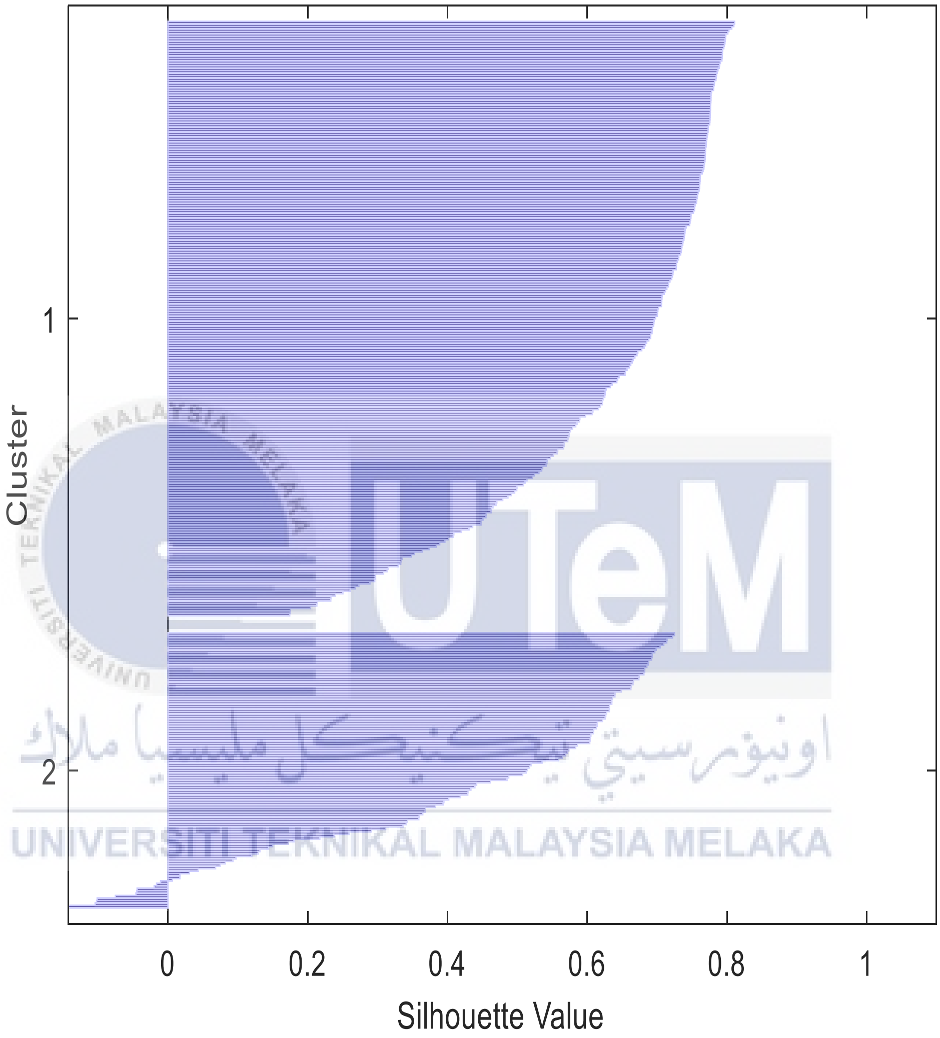


Figure 4.40: Silhouette plot for 2 clusters in year 2016

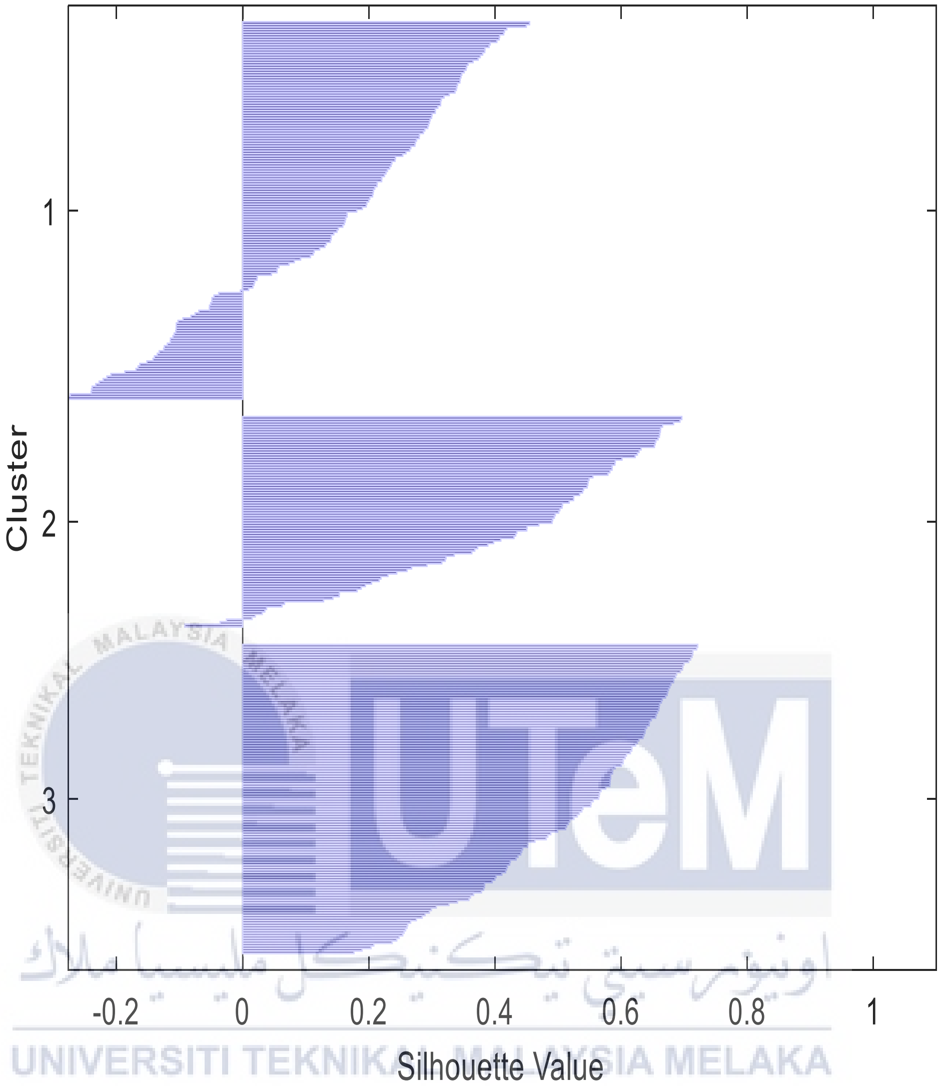


Figure 4.41: Silhouette plot for 3 clusters in year 2016

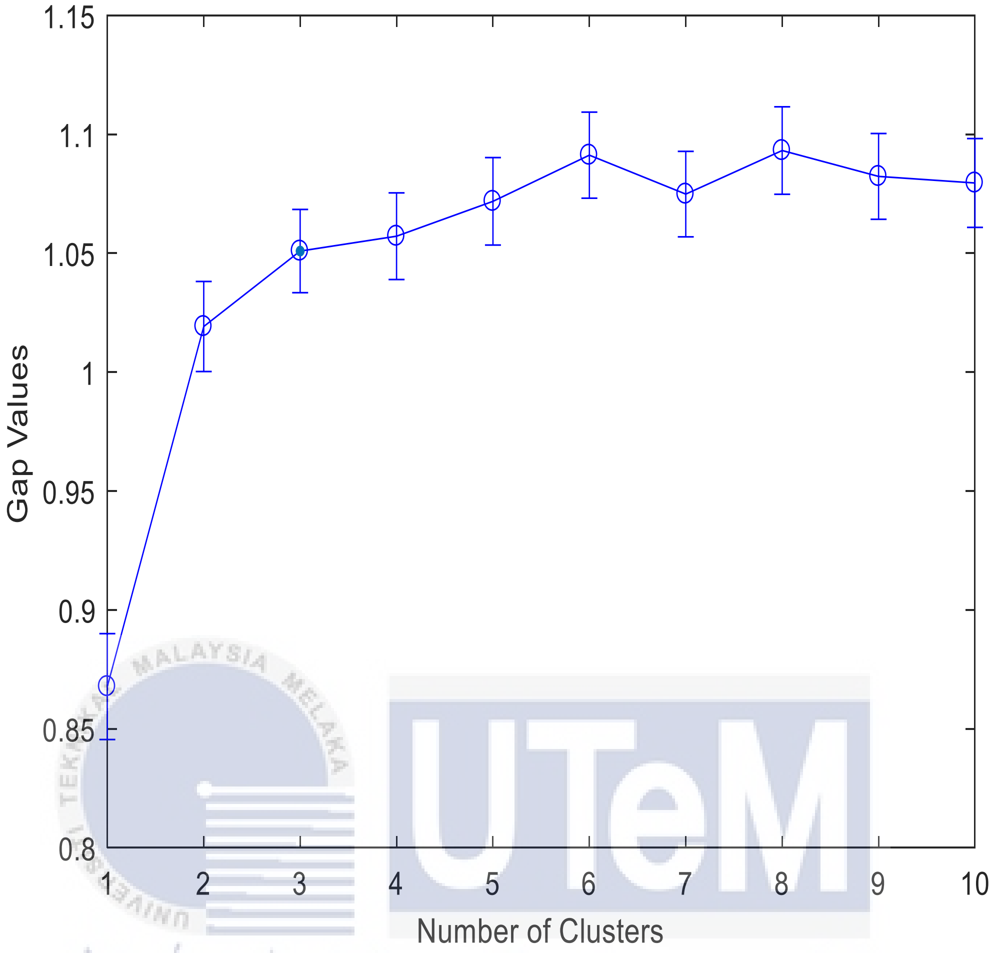


Figure 4.42: Gap statistics analysis in year 2016

Using the total irradiance data from year 2014, 2015 and 2016, the best number of cluster is K=2 as seen in Table 4.13. K=2 in Figure 4.43 remains undisputed as it has the highest overall positive silhouette value while having the least overall negative silhouette value in comparison with K=3 in Figure 4.44. In gap analysis, the correct number of cluster is chosen to be K=3. In Figure 4.45 below, gap value of K=3 is slightly lesser that the gap value of k=4. This makes K=3 a more accurate choice because it does not fall near to the border of One Standard Error of K=4.

Table 4.13: Silhouette value for dataset of year 2014, 2015 and 2016

Cluster (K)	Silhouette value
2	0.5275
3	0.3581
4	0.3509

Cluster (K)	Silhouette value
5	0.3161
6	0.2678
7	0.2655

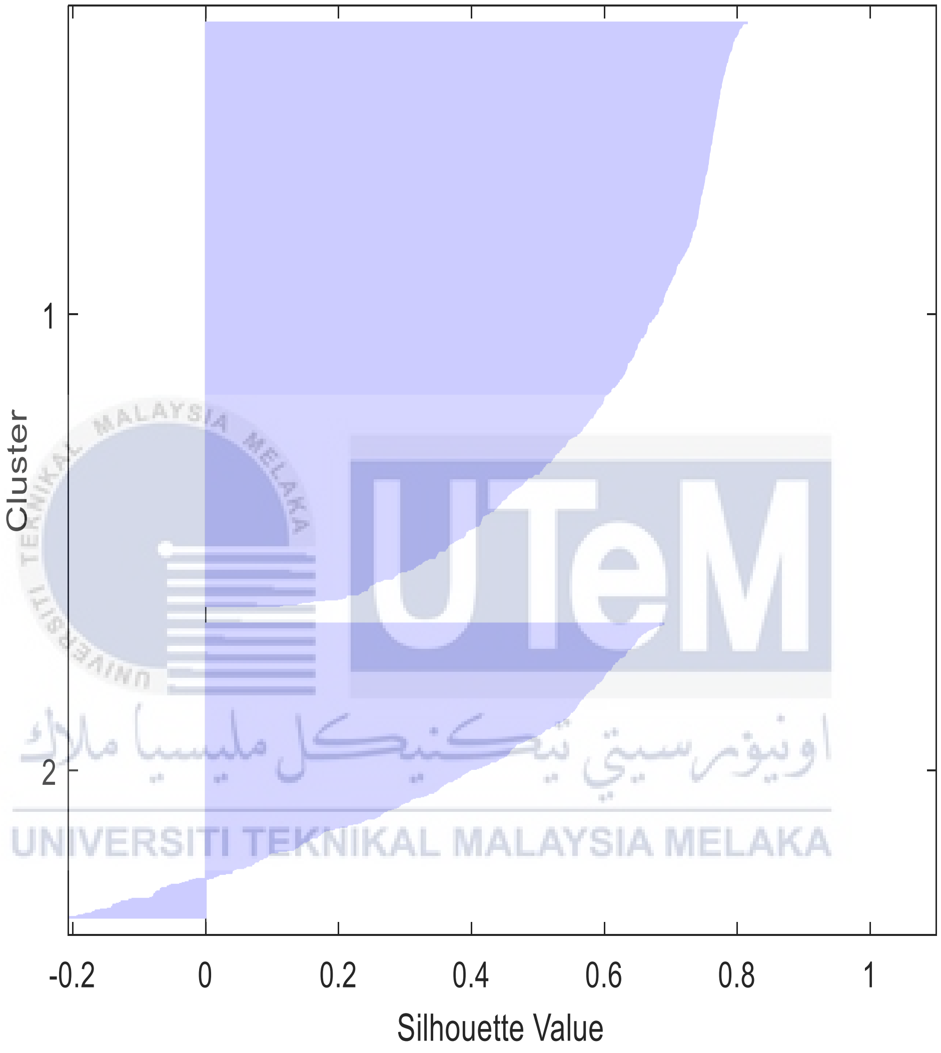


Figure 4.43: Silhouette plot for 2 clusters in year 2014, 2015 and 2016

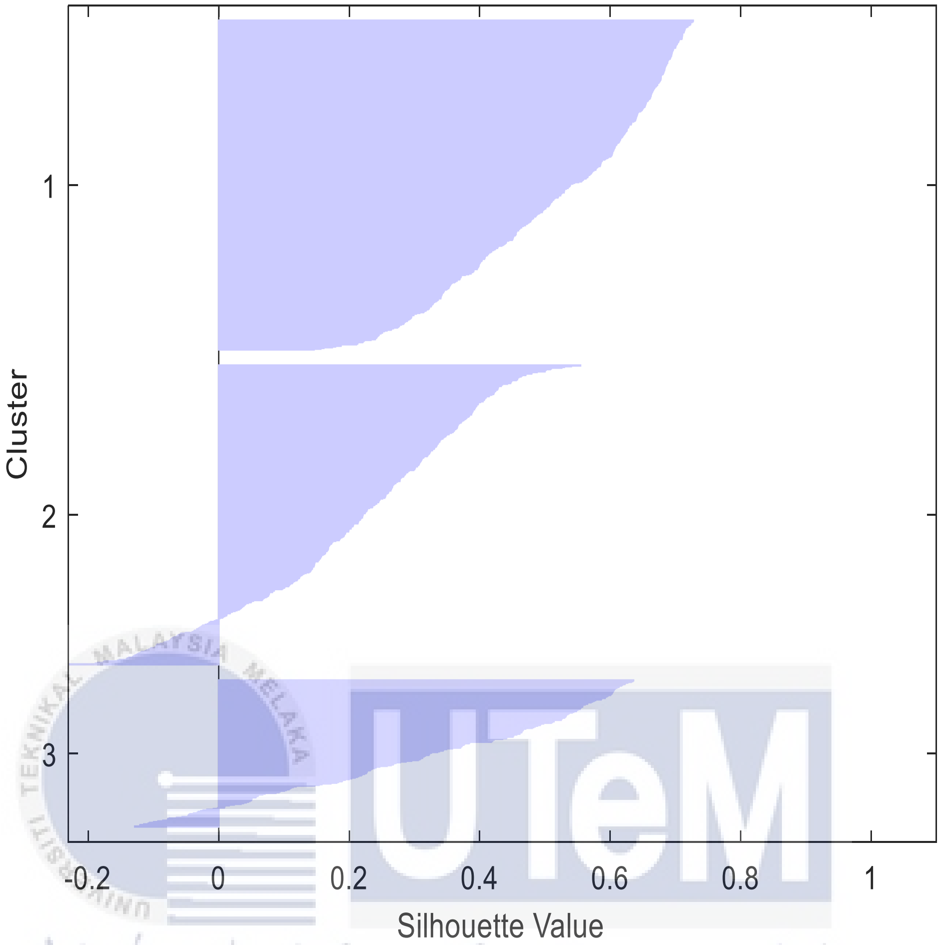


Figure 4.44: Silhouette plot for 3 clusters in year 2014, 2015 and 2016

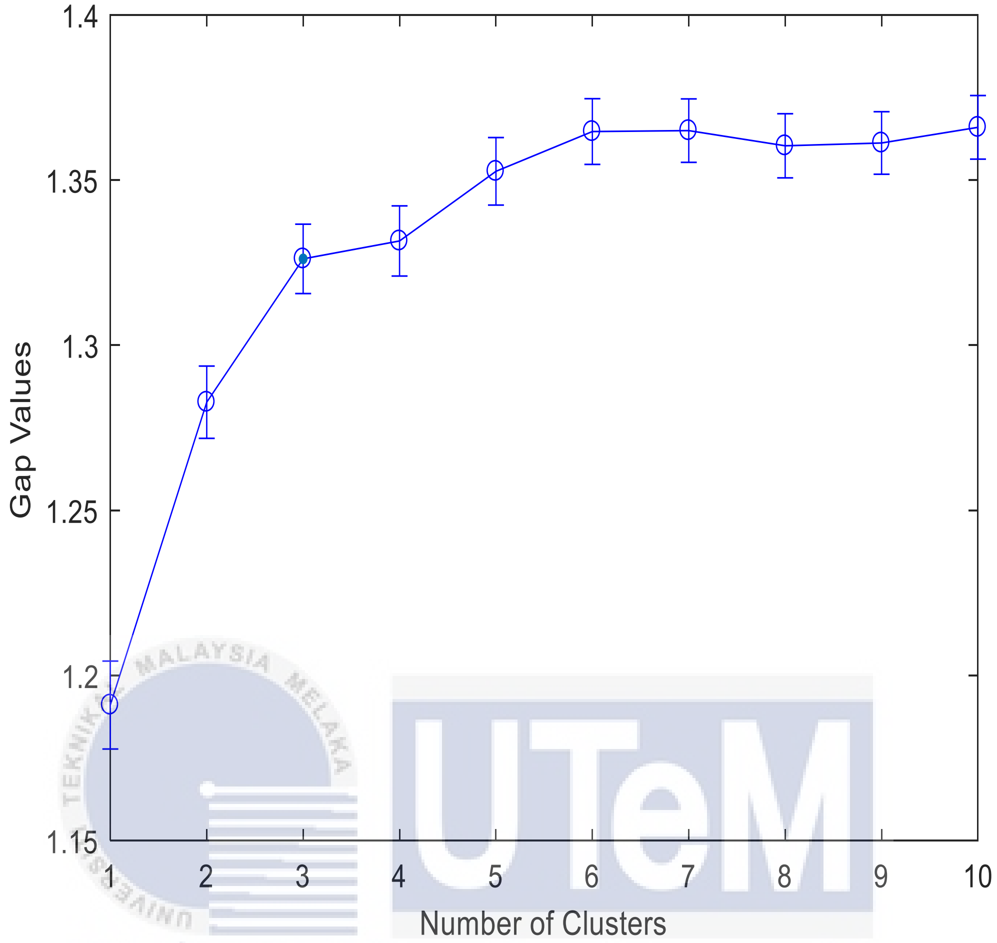


Figure 4.45: Gap statistics analysis in year 2014, 2015 and 2016

After a series of analysis, K=3 is selected for both Weka and Matlab software with consideration of the priority of each verification method. The size of dataset is influential to the final result. To further verify the validity of the final outcome, K=3, manual classification of irradiance profile is done in the next section.

4.3.3 Manual Classification of Irradiance Profile

Manual classification is crucial in clustering process. It helps to identify whether the data points in a cluster is accurate. All data points in a cluster are expected to show similar trend. Beside, manual classification also aids in identifying the category of cluster. Due to the fact that a larger dataset would derive result with greater precision, manual classification only focuses on three years' dataset. With the help of simple random sampling, the irradiance profile of ten days pertaining to the same cluster is chosen at random to avoid bias. In Weka, the chosen days are tabulated in Table 4.14 below. By looking at the irradiance profiles, cluster 3 in Figure 4.48 can be categorized as "clear sky" The trend in cluster 3 has the highest stability with minimum fluctuation. It has the highest point at irradiance more than 1000 Wm^{-2}. Comparing irradiance profiles of cluster 1 and cluster 2 in Figure 4.46 and Figure 4.47 below respectively, cluster 1 has the highest tendency to be categorized as "cloudy day". The trend in cluster 1 has more fluctuation with higher magnitude compared to the trend in cluster 2. A fast-moving and less dense cloud would give rise to a trend fluctuating between high crest and low trough at higher frequency. Cluster 2, on the other hand, has lower fluctuation magnitude. Most of the trends in cluster 2 linger closely while having lower overall irradiance. Thus, cluster 2 can be categorized as "overcast".

Table 4.14: Selection of simple random sampling (DAY)

Cluster number	Amount of data point	Selection of simple random sampling (DAY)
1	320	472, 1008, 857, 122, 1016, 147, 955, 663, 326, 470
2	236	319, 1057, 670, 601, 246, 991, 876, 1010, 350, 614
3	506	729, 453, 213, 589, 409, 168, 80, 550, 800, 562

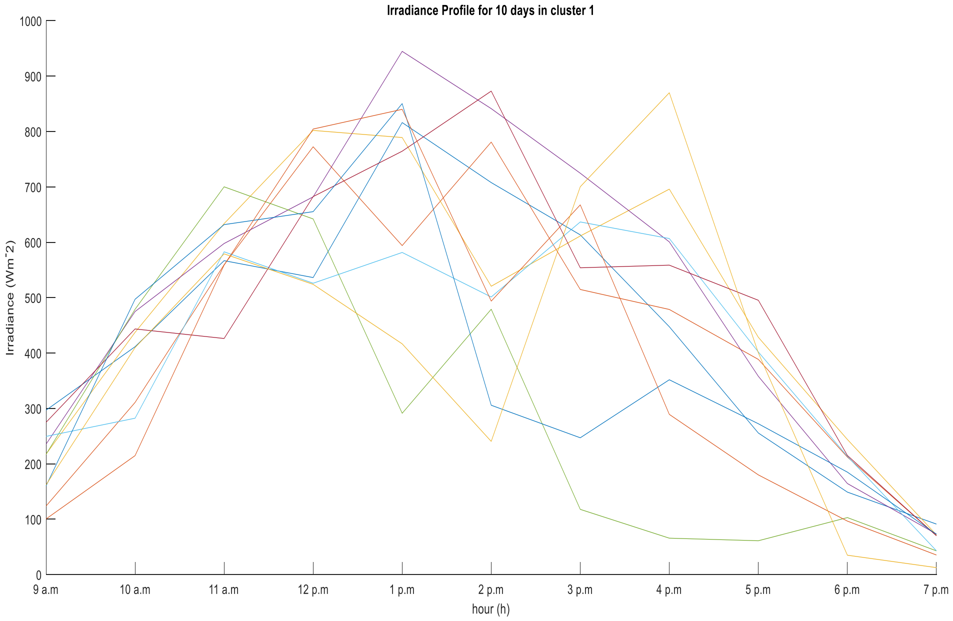


Figure 4.46: Irradiance profile of cluster 1 in Weka for year 2014, 2015 and 2016

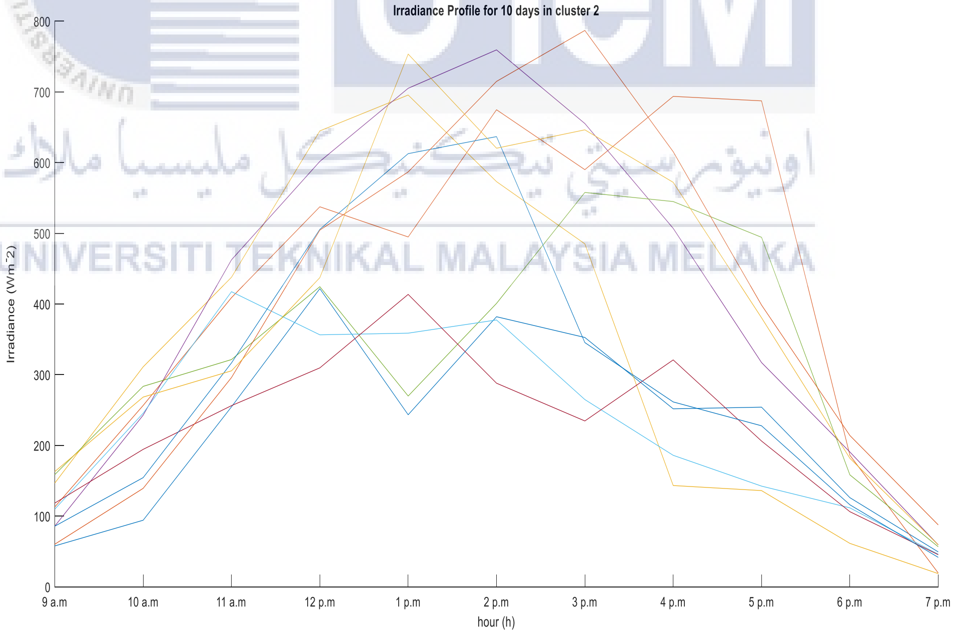


Figure 4.47: Irradiance profile of cluster 2 in Weka for year 2014, 2015 and 2016

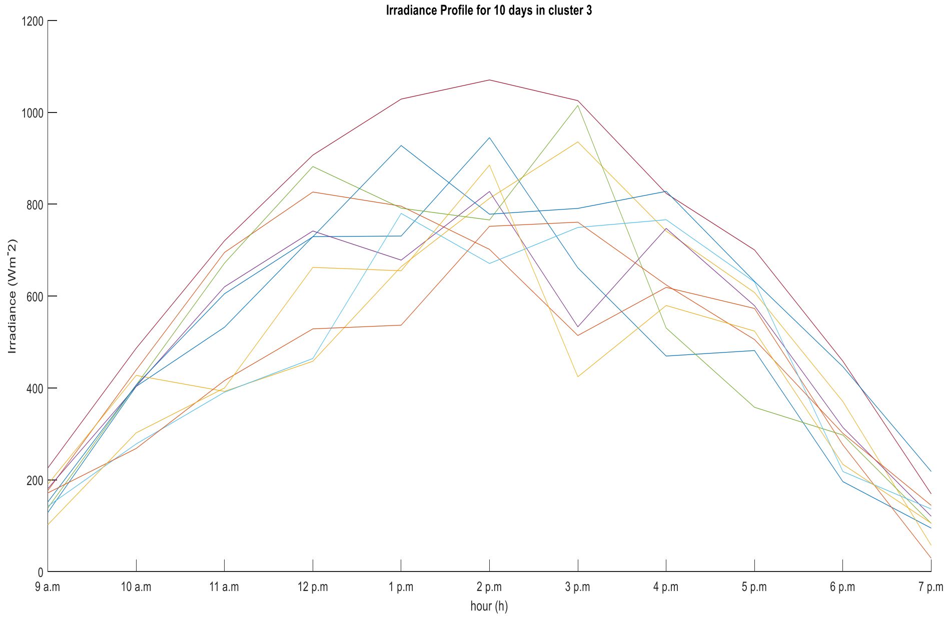


Figure 4.48: Irradiance profile of cluster 3 in Weka for year 2014, 2015 and 2016

In Matlab, the chosen days are tabulated in Table 4.15 below. On close inspection, the irradiance profiles in cluster 1 in Figure 4.49 can be categorized as "clear sky" The trend in cluster 1 has the highest stability with minimum fluctuation. It has multiple trends lingering at the highest point with irradiance of about 1000 Wm^{-2}. Comparing irradiance profiles of cluster 2 and cluster 3 in Figure 4.50 and Figure 4.51 below respectively, cluster 3 has the highest tendency to be categorized as "cloudy day". The trends in cluster 3 have more fluctuation with higher magnitude compared to the trends in cluster 2. At the same time, cluster 3 has higher overall irradiance value than that in cluster 2 while being lower that the overall irradiance in cluster 1. Cluster 2, on the other hand, has the least overall irradiance compared to the rest of the profile. Thus, cluster 2 can be categorized as "overcast" as the high magnitude fluctuation only happen at lower irradiance value.

Table 4.15: Selection of simple random sampling (DAY)

Cluster number	Amount of data point	Selection of simple random sampling (DAY)
1	506	67, 171, 359, 396, 576, 781, 839, 859, 864, 968

Cluster number	Amount of data point	Selection of simple random sampling (DAY)
2	235	152, 154, 234, 245, 489, 617, 696, 697, 708, 953
3	321	74, 77, 506, 574, 575, 618, 673, 822, 949, 987

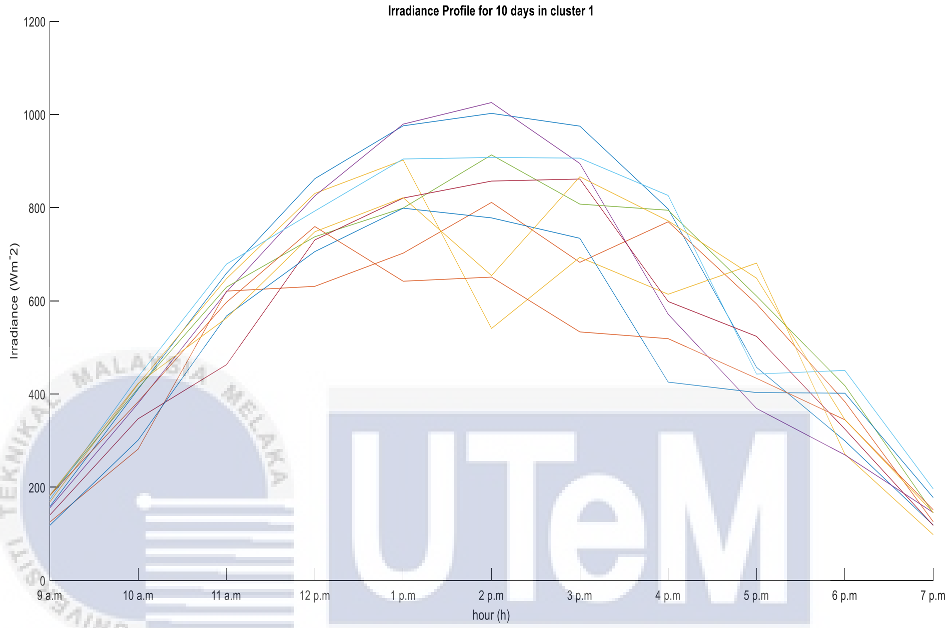


Figure 4.49: Irradiance profile of cluster 1 in Matlab for year 2014, 2015 and 2016

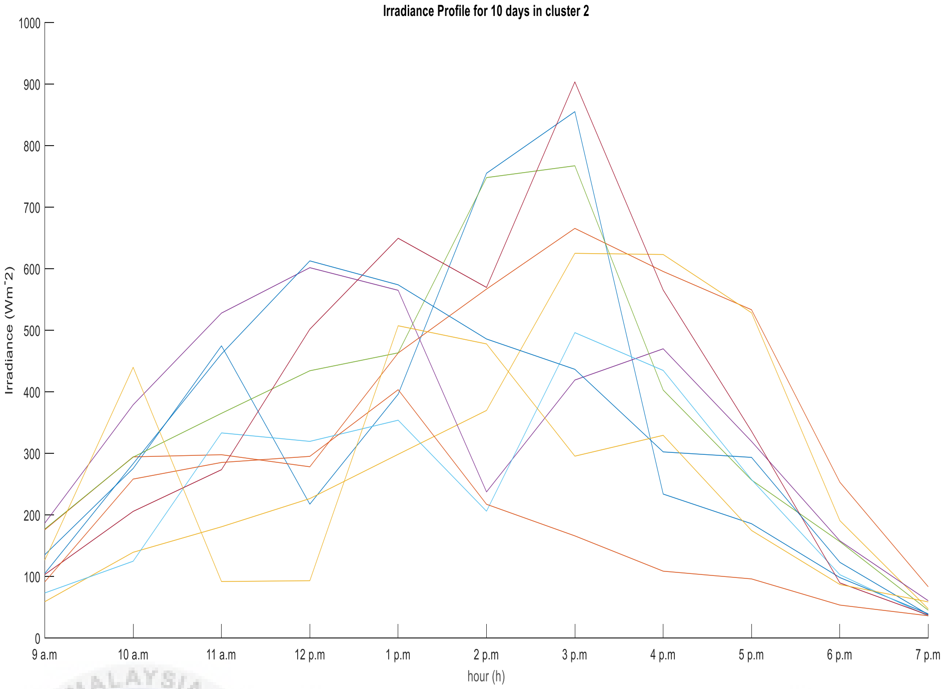


Figure 4.50: Irradiance profile of cluster 2 in Matlab for year 2014, 2015 and 2016

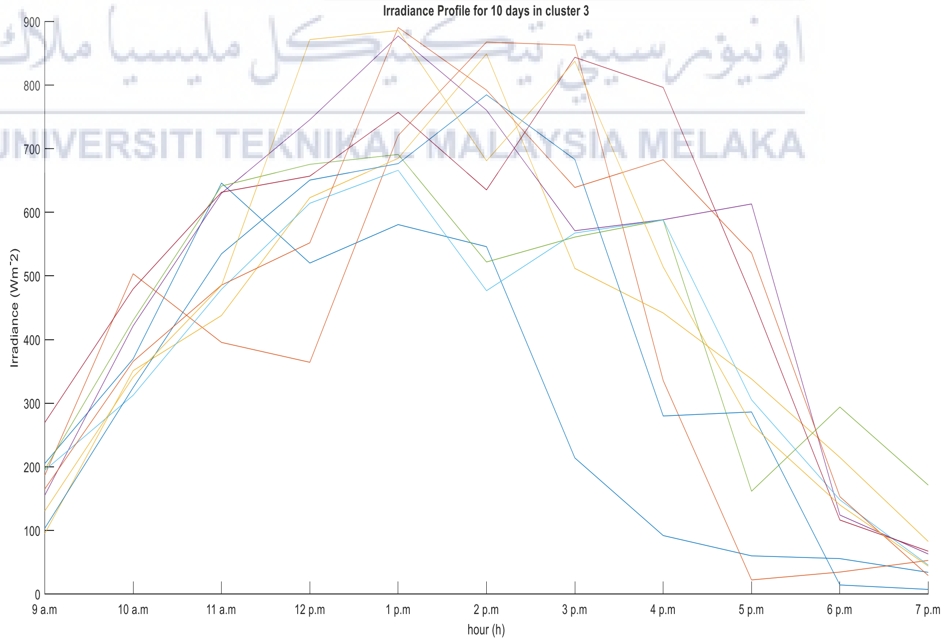


Figure 4.51: Irradiance profile of cluster 3 in Matlab for year 2014, 2015 and 2016

It is comparatively easier to identify the trends of “clear sky” while the trends for “cloudy” and “overcast” are ambiguous. Irradiance profiles with result from Matlab are less controversial than Weka’s.

4.4 Summary

To conclude, both Self Organizing Map and K-Means clustering methods are able to come to a consensus that the ideal number of cluster is 3 in preliminary clustering. Moving forward to part 2, the final clustering, the same result can be drawn from the analysis despite the large increment in data size. In final clustering, the result generated from the additional software, Weka, is proved useful. Acquiring similar result from both software helps in strengthening the confidence level of the final outcome. From the manual classification, the trends of irradiance profile in each cluster is logical and similar to a certain extent. Putting preliminary clustering result and final clustering result side by side, the only difference lies in the outcome of silhouette analysis where the optimum number of cluster is 3 in preliminary clustering while the optimum number of cluster is 2 in final clustering. The final decision depends heavily on gap statistics analysis and elbow analysis as silhouette analysis shows negligence in negative silhouette value.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

To conclude, the objectives in this project are achieved with minimal deviation. In preliminary clustering, the limitation of each method is not fully tested. The dataset of one month narrows down the possibility of unwanted outcome, giving the whole project a smoother experience. Noise, better known as outlier is considered minimum. The optimum number of cluster through silhouette analysis is identified as three, K=3. In final clustering, an additional software, Weka, is used to cluster the irradiance data along with Matlab. Acquiring a near constant outcome will be extremely difficult due to the increase in outlier number and at the same time, similarity between clusters increases, causing the cluster distinctiveness to drop. Hence, in final part, datasets are examined to eliminate outlier before clustering takes place. The final result is acquired after a thorough comparison between verification methods in both Weka and Matlab software. The optimum number of cluster is found to be the same as the preliminary clustering result at three, K=3. Manual classification in done in final clustering on the largest dataset and categorization of each cluster is successfully made.

5.2 Future Works

Some questions have arisen while completing this project. The uncertainties faced in decision making is the root cause. To reduce the uncertainties, I would like to suggest the implementation of larger dataset. Due to data limitation, some data points have to be eliminated as outliers. Some data points may possess extreme value which make them stand out from the rest of the data. The occurrence of extreme value is rare where the value (Irradiance) is either extremely small or big making it a possible outlier during analysis even if the data point is valid. By increasing the data size, the amount of rare extreme value increases. As a result, the possibility of extreme values being eliminated as outlier decreases. If sufficient, this situation would account for extra number of cluster resulting in a total different outcome. In case of the occurrence of extra number of cluster, two possible categories can be added. From the irradiance profile, the "clear sky" category shows minor fluctuation. However, with large amount of extremely high irradiance data, a possible additional cluster may be introduced where the irradiance magnitude remains high throughout the peak hours (9a.m until 2p.m). On the other hand, some data points exhibit unusually high irradiance value after peak hours. This phenomenon, too, could introduce another cluster representing irradiance profile of days where the day or daylight is the longest. Next, making this project more practical with more hands-on experience on the hardware (solar panel) would allow student to discover more about natural variables. These variables may include temperature, humidity, alleviation of the solar panel, wind speed and more. Being able to generate data on our own, a student may gain insight into the capability and importance of machine learning. A final year project combining both supervised and unsupervised machine learning could help student understand the power of each method. Strong understanding of the relation between supervised and unsupervised machine learning could lead to some breakthrough in renewable energy prediction accuracy.

REFERENCES

[1] Dr. Mamta Patel Nagaraja “How do photovoltaics work?” [online].Available: https://science.nasa.gov/science-news/science-at-nasa/2002/solarcells. [Accessed Nov. 18, 2018].

[2] P. M. A. Deshmukh and P. R. A. Gulhane, “Importance of Clustering in Data Mining,” Int. J. Sci. Eng. Res., vol. 7, no. 2, pp. 247–251, 2016.

[3] Drew Robb “Breaking the 60 percent efficiency barrier” [online]. Available: https://www.powerengineeringint.com/articles/print/volume-18/issue-3/features/ccgt-breaking-the-60-per-cent-efficiency-barrier.html. [Accessed Nov. 18, 2018].

[4] Johnzactruba, “The Efficiency of Power Plants of Differnt Types,” *Bright Hub Engineering*, 14-Nov-2018. [Online]. Available: https://www.brighthubengineering.com/power-plants/72369-compare-the-efficiency-of-different-power-plants/. [Accessed: 18-September-2018].

[5] Serm Murmson “The Average Photovoltaic System Efficiency” [online]. Available: https://sciencing.com/average-photovoltaic-system-efficiency 7092.html. [Accessed Nov. 18, 2018].

[6] “What is Machine Learning? A definition,” *Expert System*, 05-Oct-2017. [Online]. Available: https://www.expertsystem.com/machine-learning-definition/. [Accessed: 13-Aug-2018].

[7] “Machine Learning: What it is and why it matters,” *SAS*. [Online]. Available: https://www.sas.com/en_us/insights/analytics/machine-learning.html. [Accessed: 03-Aug-2018].

[8] “Types of Machine Learning Algorithms: Supervised and Unsupervised Learning,” *Netguru Blog on Machine Learning*. [Online]. Available: https://www.netguru.com/blog/types-of-machine-learning-algorithms-supervised-and-unsupervised-learning. [Accessed: 06-Nov-2018].

[9] S. Ray and Business Analytics, “7 Types of Regression Techniques you should know,” *Analytics Vidhya*, 07-Mar-2019. [Online]. Available:

https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/. [Accessed: 2018].

[10] "NVIDIA Blog: Supervised Vs. Unsupervised Learning," *The Official NVIDIA Blog*, 20-Sep-2018. [Online]. Available: https://blogs.nvidia.com/blog/2018/08/02/supervised-unsupervised-learning/. [Accessed: 15-Sep-2018].

[11] M. Sharma, G. N. Purohit, and S. Mukherjee, "Information Retrieves from Brain MRI Images for Tumor Detection Using Hybrid Technique K-means and Artificial Neural Network (KMANN)," in *Networking Communication and Data Knowledge Engineering*, 2018, pp. 145–157.

[12] S. M. Upadhyayula, S. Naish, D. Gunti, S. R. Mutheneni, and R. Mopuri, "Spatial distribution and cluster analysis of dengue using self organizing maps in Andhra Pradesh, India, 2011–2013," *Parasite Epidemiol. Control*, vol. 3, no. 1, pp. 52–61, 2016.

[13] O. Rampado, L. Gianusso, C. R. Nava, and R. Ropolo, "Analysis of a CT patient dose database with an unsupervised clustering approach," *Phys. Medica*, vol. 60, no. November 2018, pp. 91–99, 2019.

[14] Y. Li *et al.*, "Land use pattern, irrigation, and fertilization effects of rice-wheat rotation on water quality of ponds by using self-organizing map in agricultural watersheds," *Agric. Ecosyst. Environ.*, vol. 272, no. September 2018, pp. 155–164, 2019.

[15] N. Chen, B. Ribeiro, A. Vieira, and A. Chen, "Clustering and visualization of bankruptcy trajectory using self-organizing map," *Expert Syst. Appl.*, vol. 40, no. 1, pp. 385–393, 2013.

[16] P. Lin, Z. Peng, Y. Lai, S. Cheng, Z. Chen, and L. Wu, "Short-term power prediction for photovoltaic power plants using a hybrid improved Kmeans-GRA-Elman model based on multivariate meteorological factors and historical power datasets," *Energy Convers. Manag.*, vol. 177, no. October, pp. 704–717, 2018.

[17] S. Kaushik and Saurav, “An Introduction to Clustering & different methods of clustering,” *Analytics Vidhya*, 11-Mar-2019. [Online]. Available: https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/. [Accessed: 2018].

[18] “5 Amazing Types of Clustering Methods You Should Know,” *Datanovia,* 12-Nov-2018. [Online]. Available: https://www.datanovia.com/en/blog/types-of-clustering-methods-overview-and-quick-start-r-code/. [Accessed: 20-Dec-2018].

[19] H. Yu, G. Wen, J. Gan, W. Zheng, and C. Lei, “Self-paced Learning for K-means Clustering Algorithm,” Pattern Recognit. Lett., pp. 1–8, 2018.

[20] G. Gan and M. K. P. Ng, “K-Means Clustering With Outlier Removal,” Pattern Recognit. Lett., vol. 90, pp. 8–14, 2017.

[21] Y. Zhao, Y. Ming, X. Liu, E. Zhu, K. Zhao, and J. Yin, “Large-scale k-means clustering via variance reduction,” Neurocomputing, vol. 307, pp. 184–194, 2018.

[22] S. K. Majhi and S. Biswal, “Optimal cluster analysis using hybrid K-Means and Ant Lion Optimizer,” Karbala Int. J. Mod. Sci., pp. 1–14, 2018.

[23] S. C. Satapathy Joshi, Amit (Eds.), “Information and Communication Technology for Intelligent Systems,” *(ICTIS 2017)*, vol. 2, pp. 325–328, 2017.

[24] W. Gong, R. Zhao, and S. Grünewald, “Structured sparse K-means clustering via Laplacian smoothing,” Pattern Recognit. Lett., vol. 112, pp. 63–69, 2018.

[25] A. Gupta, S. Datta, and S. Das, “Fast automatic estimation of the number of clusters from the minimum inter-center distance for k-means clustering,” Pattern Recognit. Lett., vol. 116, pp. 72–79, 2018.

[26] J. Zhou, Y. Pan, C. L. P. Chen, D. Wang, and S. Han, “K-medoids method based on divergence for uncertain data clustering,” in 2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 - Conference Proceedings, 2016, pp. 2671–2674.

[27] X. S. Xu, X. L. Liang, G. Q. Yang, X. L. Wang, S. Guo, and Y. Shi, "SOMH: A self-organizing map based topology preserving hashing method," Neurocomputing, vol. 236, no. July, pp. 56–64, 2017.

[28] R. Lasri, "Clustering and Classification Using a Self-Organizing," SAI Comput. Conf., pp. 1315–1318, 2016.

[29] D. Indra Kumar and M. R. Kounte, "Comparative study of self-organizing map and deep self-organizing map using MATLAB," in International Conference on Communication and Signal Processing, ICCSP 2016, 2016, pp. 1020–1023.

[30] V. Chaudhary, A. K. Ahlawat, and R. S. Bhatia, "An efficient Self-organizing map learning algorithm with winning frequency of neurons for clustering application," in Proceedings of the 2013 3rd IEEE International Advance Computing Conference, IACC 2013, 2013, no. Step 2, pp. 672–676.

[31] A. Saraswati, V. T. Nguyen, M. Hagenbuchner, and A. C. Tsoi, "High-resolution Self-Organizing Maps for advanced visualization and dimension reduction," Neural Networks, vol. 105, pp. 166–184, 2018.

[32] Nisha and P. J. Kaur, "Cluster quality based performance evaluation of hierarchical clustering method," Proc. 2015 1st Int. Conf. Next Gener. Comput. Technol. NGCT 2015, no. September, pp. 649–653, 2016.

[33] F. Gullo, G. Ponti, A. Tagarelli, and S. Greco, "An information-theoretic approach to hierarchical clustering of uncertain data," Inf. Sci. (Ny)., vol. 402, pp. 199–215, 2017.

[34] S. Park and Y. B. Park, "Photovoltaic power data analysis using hierarchical clustering," in 2018 International Conference on Information Networking (ICOIN), 2018, pp. 727–731.

[35] J. Li and A. Nehorai, "Gaussian mixture learning via adaptive hierarchical clustering," Signal Processing, vol. 150, pp. 116–121, 2018.

[36] B. Nguyen, F. J. Ferri, C. Morell, and B. De Baets, "An efficient method for clustered multi-metric learning," Inf. Sci. (Ny)., vol. 471, pp. 149–163, 2019.

[37] Z. Nazari, D. Kang, M. R. Asharif, Y. Sung, and S. Ogawa, "A new hierarchical clustering algorithm," ICIIBMS 2015 - Int. Conf. Intell. Informatics Biomed. Sci., pp. 148–152, 2015.

[38] N. Heidari, Z. Moslehi, A. Mirzaei, and M. Safayani, Bayesian distance metric learning for discriminative fuzzy c-means clustering. Elsevier B.V., 2018.

[39] D. Olszewski, "Fraud detection using self-organizing map visualizing the user profiles," *Knowledge-Based Syst.*, vol. 70, pp. 324–334, 2014.

[40] J. Gorricha and V. Lobo, "Improvements on the visualization of clusters in geo-referenced data using Self-Organizing Maps," *Comput. Geosci.*, vol. 43, pp. 177–186, 2012.

[41] P. Sudheera, V. R. Sajja, S. D. Kumar, and N. G. Rao, "Detection of dental plaque using enhanced K-means and silhouette methods," *Proc. 2016 Int. Conf. Adv. Commun. Control Comput. Technol. ICACCCT 2016*, no. 978, pp. 559–563, 2016.

[42] M. Antunes, D. Gomes, and R. L. Aguiar, "Knee/Elbow estimation based on first derivative threshold," *Proc. - IEEE 4th Int. Conf. Big Data Comput. Serv. Appl. BigDataService 2018*, pp. 237–240, 2018.

[43] D. Marutho, S. Hendra Handaka, E. Wijaya, and Muljono, "The Determination of Cluster Number at k-Mean Using Elbow Method and Purity Evaluation on Headline News," *Proc. - 2018 Int. Semin. Appl. Technol. Inf. Commun. Creat. Technol. Hum. Life, iSemantic 2018*, pp. 533–538, 2018.

[44] T. Nguyen, A. Khosravi, D. Creighton, and S. Nahavandi, "Spike sorting using locality preserving projection with gap statistics and landmark-based spectral clustering," *J. Neurosci. Methods*, vol. 238, pp. 43–53, 2014.

[45] K. M. Prabusankarlal, R. Manavalan, and R. Sivaranjani, "An optimized non-local means filter using automated clustering based preclassification through gap statistics for speckle reduction in breast ultrasound images," *Appl. Comput. Informatics*, vol. 14, no. 1, pp. 48–54, 2018.

[46] U. Habib, K. Hayat, and G. Zucker, “Complex building’s energy system operation patterns analysis using bag of words representation with hierarchical clustering,” *Complex Adapt. Syst. Model.*, vol. 4, no. 1, 2016.

[47] “Determining The Optimal Number Of Clusters: 3 Must Know Methods,” *Datanovia*. [Online]. Available: https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/#gap-statistic-method. [Accessed: 26-Mar-2019].

[48] “CS425 Lab,” *Introduction to MATLAB*. [Online]. Available: http://www.cs.uregina.ca/Links/class-info/425/MatlabIntro/lesson.html. [Accessed: 28-Jan-2019].

[49] “Neural Net Fitting,” *Cluster data by training a self-organizing maps network - MATLAB*. [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/neuralnetclustering-app.html. [Accessed: 02-Jul-2018].

[50] Y. Zhang, “An Improved Random Sampling Approach for Large Data Set Mining,” in *Proceedings - 2016 International Conference on Smart City and Systems Engineering, ICSCSE 2016*, 2017, pp. 558–561.

[51] K. R. Prasad and B. E. Reddy, “Assessment of clustering tendency through progressive random sampling and graph-based clustering results,” *Proc. 2013 3rd IEEE Int. Adv. Comput. Conf. IACC 2013*, pp. 726–731, 2013.

[52] L. Xu and F. Zhang, “Fractional Fourier transform estimation of simple randomly sampled signals,” ICSPCC 2016 - IEEE Int. Conf. Signal Process. Commun. Comput. Conf. Proc., no. 4, 2016.

[53] “Silhouette analysis” *Silhouette plot - MATLAB*. [Online]. Available: https://www.mathworks.com/help/stats/silhouette.html. [Accessed: 28-Jan-2019].

[54] “K-means,” *k*. [Online]. Available: https://www.mathworks.com/help/stats/kmeans.html. [Accessed: 29-Aug-2018].

[55] Error Sum of Squares. [Online]. Available: https://hlab.stanford.edu/brian/error_sum_of_squares.html. [Accessed: 02-Jan-2019].

APPENDICES

APPENDIX A GANTT CHART FOR FINAL YEAR PROJECT 1

Project Activities of Final Year Project 1	Week														
	Sept			Oct					Nov				Dec		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Title selection	■														
First meeting with supervisor		■													
Introductory session of the title by supervisor			■												
Research of related studies and journals				■	■	■	■	■	■						
Software exposure						■	■	■	■	■					
Preliminary results										■	■				
FYP 1 Presentation												■			
Submission of FYP 1 final report													■	■	■

APPENDIX B GANTT CHART FOR FINAL YEAR PROJECT 2

Project Activities of Final Year Project 2	Week													
	Feb			Mac				Apr				May		
	1	2	3	4	5	6	7	8	9	10	11	27	28	29
New verification tryout with larger dataset	■	■	■	■										
Verification method analysis				■	■	■	■	■	■	■				
Data analysis and discussion							■	■	■	■	■	■		
Submission of FYP 2 report												■	■	
FYP 2 Presentation														■

APPENDIX C CODING FOR SELF ORGANIZING MAP TOOL

```
%SOM simple script
inputs = IrradianceValue31_transposed;

% Create a Self-Organizing Map
dimension1 = 3;
dimension2 = 3;
net = selforgmap([dimension1 dimension2]);

% Train the Network
[net,tr] = train(net,inputs);

% Test the Network
outputs = net(inputs);

% View the Network
view(net)

% Plots
% Uncomment to activate plots.
% figure, plotsomtop(net)
% figure, plotsomnc(net)
% figure, plotsomnd(net)
% figure, plotsomplanes(net)
% figure, plotsomhits(net,inputs)
% figure, plotsompos(net,inputs)
```

APPENDIX D CODING FOR K-MEANS CLUSTERING

```
%Setting the number of cluster to 2
idx2=K-Means(IrradianceValue31_transposed,2,'Distance','cityblock');
figure()
[silh2,h]=silhouette(IrradianceValue31_transposed,idx2,'cityblock');
h = gca;
h.Children.EdgeColor = [.8 .8 1];
xlabel 'Silhouette Value'
ylabel 'Cluster'

%Setting the number of cluster to 3
idx3=K-Means(IrradianceValue31_transposed,3,'Distance','cityblock');
figure()
[silh3,h]=silhouette(IrradianceValue31_transposed,idx3,'cityblock');
h = gca;
h.Children.EdgeColor = [.8 .8 1];
xlabel 'Silhouette Value'
ylabel 'Cluster'

%Setting the number of cluster to 4
idx4=K-Means(IrradianceValue31_transposed,4,'Distance','cityblock');
figure()
[silh4,h]=silhouette(IrradianceValue31_transposed,idx4,'cityblock');
h = gca;
h.Children.EdgeColor = [.8 .8 1];
xlabel 'Silhouette Value'
ylabel 'Cluster'

%Setting the number of cluster to 5
Idx5=K-Means(IrradianceValue31_transposed,5,'Distance','cityblock');
Figure()
[silh5,h]=silhouette(IrradianceValue31_transposed,idx5,'cityblock');
h = gca;
h.Children.EdgeColor = [.8 .8 1];
xlabel 'Silhouette Value'
ylabel 'Cluster'
```

APPENDIX E CODING FOR CLASSIFICATION OF IRRADIANCE PROFILE

```
A=year141516;
B=3;
Cluster_assignment=kmeans_profiling(A,B)

function [WOW,HeHe,SUMD,Cluster]=kmeans_profiling(X,varargin)
 [m,~]=size(X); %getting the number of samples
if nargin>1, ToTest=cell2mat(varargin(1)); else, ToTest=ceil(sqrt(m)); end
if nargin>2, Cutoff=cell2mat(varargin(2)); else, Cutoff=0.95; end
if nargin>3, Repeats=cell2mat(varargin(3)); else, Repeats=3; end
%unit-normalize
MIN=min(X); MAX=max(X);
X=(X-MIN)./(MAX-MIN);
L=zeros(ToTest,1); %initialize the results matrix
for p=1:ToTest %for each sample
    [~,~,dist]=kmeans(X,p,'emptyaction','drop'); %compute the sum of intra-cluster
distances
    tmp=sum(dist); %best so far

    for cc=2:Repeats %repeat the algo
        [~,~,dist]=kmeans(X,p,'emptyaction','drop');
        tmp=min(sum(dist),tmp);
    end
    L(p,1)=tmp;
end
Var=L(1:end-1)-L(2:end);
PC=cumsum(Var)/(L(1)-L(end));
[r,~]=find(PC>Cutoff);
K=1+r(1,1);
[WOW,HeHe,SUMD]=kmeans(X,K); %rerun one last time with the optimal number of
clusters
HeHe=HeHe.*(MAX-MIN)+MIN;
end
```

APPENDIX F CODING FOR VERIFICATION ANALYSIS AND SIMPLE RANDOM SAMPLING

```
%Mean distance of silhouette
cluster2 = mean(silh1)
cluster3 = mean(silh3)
cluster4 = mean(silh4)
cluster5 = mean(silh5)

%Gap statistics
x=year141516;
GapValue=evalclusters(x,'kmeans','Gap','KList',[1:10],'Distance','sqeuclidean','B',500,'ReferenceDistribution','PCA','SearchMethod','firstMaxSE')
plot(GapValue)

%Random sampling
n=321;
k=10;
y = randsample(n,k)
```