CE888 Assignment 2

Julien Gergi Sarkis

Abstract—Clustering is one of the fundamental techniques in machine learning and in artificial intelligence, in this report, three different datasets are chosen from UCB archive, the datasets will be loaded and inspected, after the inspection of the datasets we will cluster and evaluate all of them using K-means clustering technique, then an autoencoder will be trained on the data and it will be used as inputs to the clustering algorithms. However, the same autoencoder will be trained but with a variation on the middle layer so we can use SoftMax features. In the end an evaluation of our method will be done using a variation of cluster sizes.

Index Terms—Neural Network, Clustering, Machine learning, Data science, Autoencoders, Completeness score.

1 Introduction

Why using clustering technique? Clustering is the essential data driven application domains. It has been researched for grouping algorithms and for functions related to distance. [1]

Clustering is an important visualization and data analysis tool to use, in the studies about clustering, many different aspects has been researched, such as the right distance metric, the way to group instance to clusters and many other. [1]

In this paper, we will be using three datasets from archive, they will be loaded and inspected using jupyter notebook with the programming language python with the use of many data science and machine learning libraries. All the datasets will be clustered using K-means clustering and later evaluated. The first dataset will be Gesture Phase Segmentation, it has features from seven videos of people gesticulating, each video has two files: a raw file with in each frame the position of hands, wrists, head and spine of the user; the other file is a processed file with the acceleration and velocity of the wrists and hands. The second dataset is Grammatical Facial Expression dataset, it has 18 videos, in each video, a user in front of the camera does five sentences in Brazilian sign language five times which requires the use of grammatical facial expressions, the dataset has 36 files, 18 datapoint file and 18 target files. The third dataset is Human Activity Recognition (HAR), the type of movement will be classified in six categories: Walking, walking downstairs, walking upstairs, sitting, standing and laying.
After loading and inspecting each data set, before splitting the data randomization will be done then we will be extracting the feature and label vector and finally splitting the data into test and train. After that normalization of the dataset is done for simpler parameter selection and finally K Means clustering is done to the data. After the clustering is done, we will be training an Autoencoder on the data, and the autoencoder features will be used as inputs to clustering algorithms, then an attempt to change the middle layer to use SoftMax features. As all data science and machine learning tasks, evaluation of the work will be done.

In this paper, description of similar work and background knowledge about clustering, deep models and autoencoders will be shown, then I will write about the methodology used where I will describe the analysis achievements and the methods used to achieve them and the data sets will be described. Moreover, I will write about the experiment done, the challenges and then a discussion will be written to evaluate the methods followed by a conclusion.

2 Background Study

2.1 Similar Work

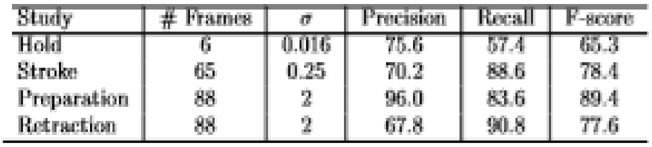
A lot of work has been done using the same datasets and there is a lot of similar work on clustering and on autoencoders; we will see some similar work done on the subject.

In the past years, there has been more studies on gesture, which goal is to analyze the use of some parts of the body to help with the communication. The research on gesticulation is studied in many fields and it is studied by capturing a video of humans doing gesture while speaking and then analyze it. Since machine learning is good at knowing pattern it has been used for this purpose. In a paper, spatial temporal information and machine learning has been used for the segmentation of gesture unit, in this work a video was the input for the gesture unit segmentation which can be looked at also as a classification problem. The software used was based on Microsoft Kinect sensor. The performance of the experiments made, first experiment had 70% of frames for training and the rest for testing, the second experiment used one video for testing and

- *Gergi Sarkis Julien, Student in the CSEE department in the University of Essex. Jg18196@essex.ac.uk*

another for training. The experiments done were assisted by the software MATLAB. Avoiding overfitting, every 10 training epochs the models were tested. For the first experiment, models with data vector had the best result, for the second experiment the data with trajectory and position had better performance than the data with acceleration and velocity. [2]

Another study on gestures was made utilizing support vector machine, also in this study a video with sequence of frames is the input, and the goal was to classify the gestures to classes like rest, stroke and others, in figure one you can find results for this study. [3]

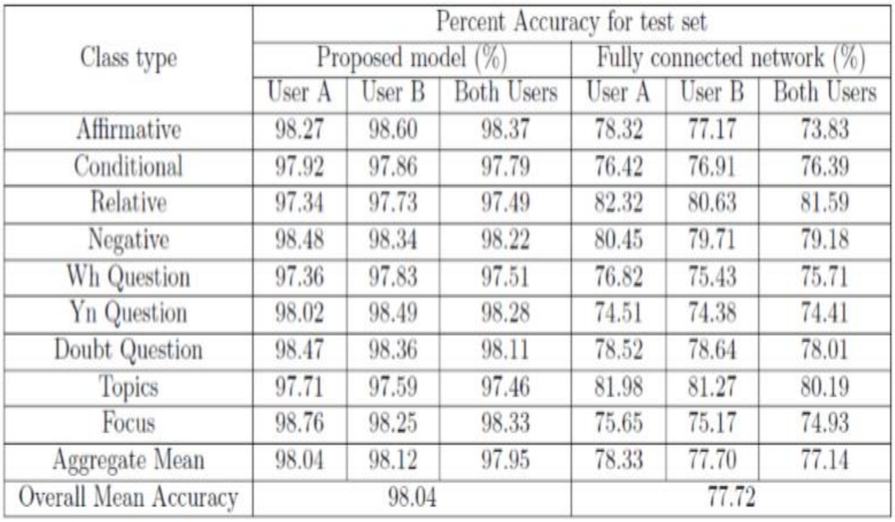


Study	# Frames	σ	Precision	Recall	F-score
Hold	6	0.016	75.6	57.4	65.3
Stroke	65	0.25	70.2	88.6	78.4
Preparation	88	2	96.0	83.6	89.4
Retraction	88	2	67.8	90.8	77.6

Figure 1 Results for SVM [3]

Another experiment, on grammatical facial expression recognition using deep neural network; it is about sign language which is a communication for deaf people. The experiment focuses on the Brazilian sign language libras trying to classify hand signs to their original meaning while in the same time focusing on emotions followed by the signs, which means facial expressions accompanied by the sign language is important to the communication. These facial expressions are the grammatical facial expressions which are useful in a lot of applications. the dataset they used consists of 225 videos with five sessions of recording, each session one sentence is recorded, in total the dataset has 27965 frames. They started with z score standardization, then a customized feed-forward deep neural network is made, it has two layers that are hidden with normal input and output layers. After initialization, tuning and network training, the system had an overall accuracy of 98.04% a table in figure two show the accuracy of the experiment. [4]

Figure 2 Comparison of the accuracy in the experiment [4]



Class type	Percent Accuracy for test set					
	Proposed model (%)			Fully connected network (%)		
	User A	User B	Both Users	User A	User B	Both Users
Affirmative	98.27	98.60	98.37	78.32	77.17	73.83
Conditional	97.92	97.86	97.79	76.42	76.91	76.39
Relative	97.34	97.73	97.49	82.32	80.63	81.59
Negative	98.48	98.34	98.22	80.45	79.71	79.18
Wh Question	97.36	97.83	97.51	76.82	75.43	75.71
Yn Question	98.02	98.49	98.28	74.51	74.38	74.41
Doubt Question	98.47	98.36	98.11	78.52	78.64	78.01
Topics	97.71	97.59	97.46	81.98	81.27	80.19
Focus	98.76	98.25	98.33	75.65	75.17	74.93
Aggregate Mean	98.04	98.12	97.95	78.33	77.70	77.14
Overall Mean Accuracy	98.04			77.72		

2.2 Background knowledge

Clustering is the action of organizing objects that are similar to each other in groups, which is in machine learning and artificial intelligent one of the most used tasks. In the last ten years, many clustering algorithms has been introduced and successfully used in big real-world data and tasks. There are two types of clustering: Similarity-based clustering and Feature-based clustering. The first one is based on distance matrix which is basically a N x N matrix which calculates the distance between pairs of the samples N. the most known similarity-based clustering is the Spectral clustering. A feature-based is based on N x D matrix as an input, where D is the dimension of the feature and N is the number of samples. The most used and popular feature-based clustering is the K-means which will be used in this assignment. The goal of K-means clustering is dividing all the samples into K clusters. [5]

Clustering is an enormous field to focus on in the area of data analysis specially that these days we have a large amount of big and multidimensional data. All the focus in this field has made clustering a well-studied task. Living in the era of big data where the work is made utilizing large datasets, clustering methods should adapt to these conditions. Grouping technique are the most famous way of clustering with the most common method K means which is a grouping method. The grouping method work by partitioning the data into similar partitions, the number of parts divided are normally decided by the user. [6]

According to scikit-learn, K-means clustering groups data by splitting samples in n groups which has same variance. This method needs that the number of groups to be specified. It has been used in many applications in different fields. Theoretically, it splits a set of N samples X into K cluster C; each known by the mean of the sample in the groups or clusters C, these means are described by centroids of the clusters. This K mean algorithm goal is to choose a centroid which will have a minimized inertia. [7]

Deep generative models are known to be successful in studying coherent latent representations for the data which are continuous for example images, artworks and audio. But they are not good with discrete data as they face many challenges. From chancing facial expressions in images to real musical improvisation to the creation of artwork which look as realistic as real artwork, Generative machine learning models proved to be successful. [8]

Deep clustering uses the deep neural networks to study feature representation which is advisable for clustering. In neural network the famous deep clustering algorithms are stacked autoencoders, it needs layer-wise training before tuning; it is made from completely connected layers but they are incompetent with images. [9]

Autoencoding is an algorithm that compresses data, they are data specific meaning that they exclusively compress data that are similar to the data they have been trained on. However, autoencoders are lossy, saying that because the decompressed outputs are less than the inputs. Moreover, they are automatically learned from examples of data which is a positive property because it is simple to train particular instances of algorithms that has a good performance on a particular type of input. [10]

According to keras, building an autoencoder three things are needed, both an encoding and a decoding function and a distance function between the loss of information in the compressed data and the decompressed. In figure three you can see a representation on the autoencoder.

Variational autoencoder gives a formulation where the encoder z is taken as a latent variable in a generative model which is probabilistic. [8]

Conventional autoencoder are made from two layers, corresponding to an encoder and a decoder, the goal is to search for a code in each input with minimizing the mean squared errors that is between the output and the input for all the samples. [9]

Deep latent variable models, prepared utilizing generative adversarial networks or variational autoencoders both are now vital strategy for portrayal of continuous structures and they have proved important advance in learning representation of high dimensional continuous data such as images. [11]

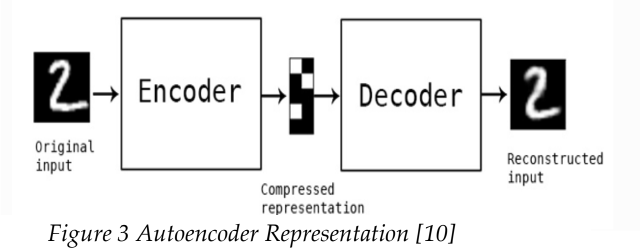


Figure 3 Autoencoder Representation [10]

Autoencoders have an important role in unsupervised machine learning and in deep neural networks. They are simple circuits that learns, with the goal to change inputs into outputs with less distortion, although they are simple, but they have a fundamental role in machine learning. Recently, they are playing a role in deep architecture where they are stacked and trained in a unsupervised way, then the top layer is trained in a supervised way followed by fine tuning the entire system. [12]

Clustering of data is simple in pattern recognition, but if the original data is not well distributed because of the variance being large, it is hard for the common clustering methods to have a good performance. For this problem, the autoencoder is a good solution for this kind of problem, the autoencoder gives a nonlinear mapping function by learning the encoder and the decoder. The autoencoder is not commonly used for clustering which requires some optimization which has two parts, distance between data and the cluster center in the space, and reconstruction error, during this phase the centers of cluster and the **data** representation are constantly updated. [13]

Deep learning has been used in application in many fields such as speech recognition and image classification, it has been an important topic in machine learning and artificial intelligence, multiple methods and algorithms have been studied and successfully used in real tasks. Because spectral clustering and autoencoder are alike in theory, autoencoders are better in efficiency and more flexible. [14]

3 Dataset Description

In this section, a description of the datasets used in the project will be shown. We are using three different datasets downloaded from https://archive.ics.uci.edu/ml/datasets.php and all the descriptions are taken from a readme file that is downloaded with the dataset; The first dataset is Gesture Phase Segmentation dataset which is made from features extracted from seven videos with humans doing gestures, the second dataset used is grammatical facial expression which is made from eighteen videos of users doing Brazilian sign language, the third dataset is Human Activity Recognition which is made from a group of 30 people each person performed six activities while wearing a smartphone on his waist.

3.1 Gesture Phase Segmentation Dataset

This dataset was created in June 2014 by Renata Cristina Barros Madeo, Priscilla Koch Wagner, and Sarajane Marques Peres.

This dataset is made from seven videos with people doing gestures, each video has two files, a Raw file with the position of wrists, hands head and spine of the person in each frame, the second file contains the velocity and acceleration of the hands and wrists. It was captured using Microsoft Kinect sensor, three persons were reading comic strips and then telling the stories in front of the Kinect sensor. Using this way, they have obtained a timestamp meaning an image of each frame and a file that has positions with coordinates x, y, z of the left hand, left wrist, right hand, right wrist, spine and head.

It is made from fourteen files, seven of them are raw files and the rest are processed files; one pair for each video. The data is named A, B, C each of them corresponding to a user and the numbers 1, 2, 3 are the stories that were told. The raw files are made from the information collected from Microsoft Kinect, and the processed files are made of scalar and vectoral velocity and acceleration of the hands and wrists.

The number of instances of the videos are:

- A1 = 1747 frames
- A2 = 1264 frames
- A3 = 1834 frames
- B1 = 1073 frames
- B3 = 1423 frames
- C1 = 1111 frames
- C3 = 1448 frames

The raw files have eighteen numeric attributes, a class attribute and a timestamp. The other processed files have thirty-two numeric attributes and a class attribute. These attributes are the x, y, z coordinate for each hand, wrist and the head and spine in the raw files. And in the processed files it is the x, y, z coordinate of the vectoral and scalar velocity of each hand, wrist and the head and spine. There are no missing attribute values in this dataset. [15]

3.2 HTRU2

This dataset was created in 2013 by Thornton D by utilizing a DMs between zero to two thousand cm -3pc.

This dataset reports a sample of pulsar candidates. HTRU2 was taken during a High Time Resolution Universe (HTRU) analysis that was made by Thornton D.

It has 19898 total instances, with eight features each of them has the label of either positive or negative. All the instances will be predicted to have either of those labels positive or negative.

Each candidate is reported by eight variables and one class variable. The first four are simple statistics collected from the integrated pulse profile. The rest of them are obtained from Dm-SNR curve. The attributes are:

1. Mean of the integrated profile.
2. Standard deviation of the integrated profile.
3. Excess kurtosis of the integrated profile.
4. Skewness of the integrated profile.
5. Mean of the DM-SNR curve.
6. Standard deviation of the DM-SNR curve.
7. Excess kurtosis of the DM-SNR curve.
8. Skewness of the DM-SNR curve.
9. Class

The dataset has 16259 negative instances of 17898 and the rest, 1639 are positives. The negative examples are spurious instances caused by noise, and the rest are real pulsar instances. These instances are supervised by annotators.

HTRU2 dataset contains two files, a CSV file which we used in this project and an ARFF file which is used normally as data mining tool. Candidates are in both files in different rows, the rows describe the variables and the last entry is the class label which is zero if negative and one if positive. [16]

3.3 Human Activity Recognition

It is created by Jorge L. Reyes-Ortiz, Davide Anguita, Alessandro Ghio, Luca Oneto and Xavier Parra in the year of 2012.

The human activity recognition dataset is done with thirty users ranging for nineteen to forty-eight years old. Every person did six activities, walking, walking up the stairs, walking down the stairs, sitting, standing and laying down, they did all those activities while having a smartphone on their waist. Using the smartphone, they collected three axial linear velocity and angular acceleration.

Each recording has captured tri-axial and body acceleration, tri-axial angular velocity, a 561 features vector in the domain of time and frequency, an activity label and an identifier of the user.

The dataset contains information about the features, a list containing the features, label for the activity name a training and testing set and labels.

Noting that in this dataset, the features are normalized and in the range of -1 and 1, every feature vector on the text file is a row. [17]

4 Methods Used

In this section, we will talk about the methods used and the machine learning and artificial intelligence algorithms utilized for the dataset, we will evaluate the results collected from those algorithms and a discussion will done on each dataset because every dataset is composed from different type of data.

Every dataset will be written about separately with each step done and all the code were in python using jupyter notebook.

4.1 Gesture Phase Segmentation Dataset

This was the first dataset worked on in this assignment after reading and understanding the description of the dataset and what it is contained of, the work of the project can be started on this data.

Firstly, after configurating and importing the needed libraries, loading and reading the dataset on jupyter notebook after that I removed the timestamp and the phase from the unprocessed raw data then visualized the some of the data read from the dataset, checking their shape, columns and what they are containing. To make everything more convenient, I renamed the ‘Phase’ in the va3 files into ‘phase’ then encoded and estimated the number of instances of the different labels. Before splitting the data into train and test, shuffling of the data is done for reducing variance and to have less overfitting, after that I extracted the feature and label vector, and split the data into 60% training and 40% testing. Furthermore, there were nan values in the data, so they were filled with the mean, after that normalization of the data is made for simpler selection of the parameter.

After making the data ready and based on the literature review done on the subject K-mean clustering will be done to the data which is an unsupervised machine learning technique that groups together sets of objects where objects in the same group or cluster are more similar to the other objects in the other clusters. This technique is proved to be successful in many fields like image analysis, pattern recognition and many more. However, K means is a centroid-based type of clustering that groups the objects in clusters based on which is closer to a centroid distance of the cluster. [18]

However, after completing the K-mean clustering technique you can see in the table below a result of the evaluation scores.

n_clusters	10	100	1000	3951
Adjusted rand score	0.028256	0.060753	0.023806	0.0044599
Homogeneity score	0.057486	0.30425	0.56637	0.86382
Completeness score	0.049774	0.11806	0.15012	0.18097

We can see that the more clusters we have the higher the homogeneity score is, the adjusted rand score increased in the first increase of the clusters but then started decreasing after 1000 clusters, and the completeness score increased with the increase in the number of clusters; we can say that the best number of clusters to use in this experiment is 1000.
However, after looking at these results we can say that clustering that is made on the data gave us poor result; further improvement in the method used could be done to increase the effectiveness of the algorithm used.

4.2 Human Activity Recognition Dataset

The second dataset that I've work on during this assignment is the human activity recognition dataset. After further understanding of the data and exploration of the features and what the data is contained of, I've started coding.

At first, the usual configuration is done, importing the libraries, reading and processing the data. In this dataset I've coded a function to visualize the data which we must see clearly. Before training, the data should be standardized for the values of the mean. After that, I've reduced the PCA to decrease dimensions and lower computation. After finishing with preparing the data, I've split it into train and test, in this dataset, the split is 70% training and 30% testing.

After the splitting of the data, I've applied K-means clustering to it and you could see the results of the unsupervised method in the table below, 5, 10,15 and 20 clusters were tried.

n_clusters	5	10	15	20
Adjusted rand score	0.33191	0.36209	0.38968	0.36209
Homogeneity score	0.38281	0.74196	0.6925	0.74196
Completeness score	0.99005	0.49059	0.50974	0.49059

From the table, we can see that when the number of clusters increased the adjusted rand score increased until a certain number of clusters after that it started decreasing, the homogeneity on the other hand was varying up and down with the number of clusters and the completeness score was decreasing as the number of clusters decreased. So, in this dataset I would say that the best number of clusters that were tested is five clusters. This dataset proved better result than the previous dataset with using thousand clusters as the number of clusters, it resulted in better adjusted rand, homogeneity and completeness score.

4.3 HTRU2

The HTRU2 dataset is a replacement for the dataset Grammatical Facial Expressions which was replaced because of unresolved problems that happened while trying to work on.

This dataset consists of only two files, in this assignment we utilized only one of it which is the csv file; it has 9 columns, one of them is the class that we need to predict, the system will be predicting two outcomes either positive or negative.

After configurating, importing the needed libraries, and loading the data. a visualization of the data is done to see and know furthermore about our dataset. Before the data is being split, shuffling of the data is done; the dataset is then split into train and test with 80% train and 20% testing.

We then cluster what we have, in this dataset we will try 2, 5, 15 and 20 clusters and compare their result in a table below

n_clusters	2	5	15	20
Adjusted rand score	0.61085	0.12628	0.035984	0.029631
Homogeneity score	0.45636	0.52709	0.66948	0.68335
Completeness score	0.38011	0.12381	0.085047	0.079383

In this dataset, the adjusted rand score decreased dramatically when the number of clusters increased, but the homogeneity score increased when increasing the number of clusters, on the other hand, the completeness score decreased also dramatically as the number of clusters increased. For this experiment with this dataset, the best number of clusters would be two, because it had the best performance. The algorithm was trying to predict two labels either positive or negative.

5 Discussion

After the literature review and the background study on the previous work done on the subject and the experiments done with the datasets, the clear thing that I've discovered is that the world of data is vast and there is an algorithm for nearly every possible shape, type and size of data.

There are libraries for python language for all the methods and algorithms regarding machine learning and data science which makes it easy and fast to work on a dataset.

The datasets used in this assignment did not have good results when evaluated, the reason of that is the data being very big, complicated, and probably the type of data had an effect on the results, it required more work to fix and simplify, on each dataset I've tried a method for improving the data before fitting the clustering and evaluating, and each resulted in different numbers, the first dataset had very poor results, the other two dataset had better results but still not the best results.

Building the autoencoder is done using keras library, it is used in this project to be trained on the data and to use the autoencoder middle layer features as inputs for the clustering that is previously done. This technique should give better results to the system.

Problems with my TensorFlow library occurred and I couldn't fix it, with many attempts to fix by searching over the internet for fixes but were failed. And it was not possible to work on the machines in the lab because I was abroad on Easter vacation. For this reason, I couldn't experiment with the autoencoder.

Conclusion

This assignment required to work on multiple datasets and made it possible to explore and evaluate multiple machine learning algorithms and included the use of autoencoders to encode and decode the data but what we used in this project is the middle layer of the autoencoder and utilized it as features for the clustering.

Three datasets were chosen from archive, explored and visualized, clustering techniques were used on the data and each dataset have been compared. Autoencoders have been trained then on the data and the middle layer of the autoencoder have been used as inputs for the clustering.

REFERENCES

[1] J. Xie, R. B. Girshick and A. Farhadi, "Unsupervised Deep Embedding for Clustering Analysis," 2015.

[2] P. Wagner, S. Peres, R. Madeo, C. Lima and F. Freitas, "Gesture Unit Segmentation Using Spatial-Temporal Information and Machine Learning," *Proceedings of the 27th International Florida Artificial Intelligence Research Society Conference,* 2014.

[3] R. C. B. Madeo, S. M. Peres and C. A. d. M. Lima, "Gesture phase segmentation using support vector machines," *Expert Systems with Applications,* vol. 56, pp. 100-115, 2016.

[4] D. Walawalkar, "Grammatical facial expression recognition using customized deep neural network architecture," 2017.

[5] Z. Jiang, Y. Zheng, H. Tan, B. Tang and H. Zhou, "Variational Deep Embedding: An Unsupervised and Generative Approach to Clustering," 2017.

[6] B. Wu and B. M. Wilamowski, "A Fast Density and Grid Based Clustering Method for Data With Arbitrary Shapes and Noise," *IEEE Transactions on Industrial Informatics,* vol. 13, no. 4, pp. 1620-1628, 2017.

[7] "2.3. Clustering — scikit-learn 0.20.3 documentation," Scikit-learn.org, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/clustering.html#k-means. [Accessed 12 April 2019].

[8] M. J. Kusner, B. Paige and J. M. Hernández-Lobato, "Grammar variational autoencoder," vol. 70, pp. 1945-1954, 2017.

[9] X. Guo, X. Liu, E. Zhu and J. Yin, "Deep Clustering with Convolutional Autoencoders," *Neural Information Processing,* pp. 373-382, 2017.

[10] "Building Autoencoders in Keras," Blog.keras.io, 2019. [Online]. Available: https://blog.keras.io/building-autoencoders-in-keras.html. [Accessed 12 April 2019].

[11] J. Zhao, Y. Kim, K. Zhang, . A. M. Rush and Y. LeCun, "Adversarially Regularized Autoencoders," 2017.

[12] P. Baldi, "Autoencoders, Unsupervised Learning, and Deep Architectures," *Proceedings of ICML Workshop on Unsupervised and Transfer Learning,* 2012.

[13] C. Song, Y. Huang, F. Liu, Z. Wang and L. Wang, "Deep auto-encoder based clustering," *Intelligent Data Analysis,* vol. 18, no. 6S, pp. S65-S76, 2014.

[14] F. Tian, B. Gao, Q. Cui, E. Chen and T.-Y. Liu, "Learning Deep Representations for Graph Clustering," *AAAI Conference on Artificial Intelligence,* pp. 1293-1299, 2014.

[15] "UCI Machine Learning Repository: Gesture Phase Segmentation Data Set," Archive.ics.uci.edu, 2019. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Gesture+Phase+Segmentation. [Accessed 19 04 2019].

[16] "UCI Machine Learning Repository: HTRU2 Data Set," Archive.ics.uci.edu, 2019. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/HTRU2. [Accessed 19 04 2019].

[17] "UCI Machine Learning Repository: Human Activity Recognition Using Smartphones Data Set," Archive.ics.uci.edu, 2019. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones. [Accessed 19 04 2019].

[18] "K-Means Clustering with scikit-learn," DataCamp Community, [Online]. Available: https://www.datacamp.com/community/tutorials/k-means-clustering-python. [Accessed 15 04 2019].