CE888 Assignment 1

Ogulcan Ozer

***Abstract*—The traditional unsupervised machine learning algorithms like k-Means, PCA and other linear factor models are proven to be very useful for many important applications. But in this day and age we are able to collect very complex, high dimensional data and these simple but efficient algorithms are failing to learn or represent it [1, p. 151] In this work we are going to use autoencoders to reduce the dimensions of three different datasets and try to extract good features with and without softmax activation function. Then finally, we will cluster the data by using the extracted features and compare the results.**

I. Introduction

UNSUPERVISED learning is a branch of machine learning that deals with unlabeled data to find patterns, groups and relations. This sub section of machine learning is useful to describe and understand unknown data, which also helps us understand the properties of future data samples. Right now we are collecting data much faster than we can process it. And most of the time the data we acquire is unlabeled, and these huge amounts of data are complex, noisy and it might include unnecessary information.

There are different ways to reduce the complexity of a given data and learn good features from it. In this work, we will look at autoencoders, one of those procedures used to learn features from a given data. Autoencoders are a part of representational learning, which can be trained with or without labels. They play a key role in deep learning applications. They are used in dimensionality reduction and feature extraction, and together with deep learning they gained a lot of traction after Hinton proposed a method to train deep belief networks one layer at a time [2].Another addition to the autoencoder will be the softmax activation function. Since softmax gives us the probability distribution over-the classes of our data- the output vector [1, pp. 178-181], we can use the output of our encoder for cluster assignment. The motivation for using autoencoders is, in what they do they are very similar to traditional feature selection algorithms, but they are also capable of learning good features from complex and high dimensional data. This aspect of autoencoders and unsupervised learning in general is important. As LeCun states in [3], people are not paying attention to unsupervised learning because of the success of the supervised learning in recent years. But it will be far more important in the future because of how living beings perceive and learn, they do not need labels to recognize the world around them.

In this paper we are going to look at previous work (Section II), look at different methods and present their results to show how they compare. In Section III, we are going to present our goals and what we want to accomplish. Then we will talk about the datasets that we are going to use for the task and give detailed information about them. The planned experiments and analyses for the task will be mentioned in Section IV. Evaluation methods that are going to be used after the experiments will be presented in Section V.

II. Background

Clustering is an important part of data analysis, that is used to naturally group the samples in order to extract information or make decisions [4]. Autoencoders were discovered in 1985 and presented in [5]. As the technology and our computational power improved immensely in the last decade, they became more and more feasible. After Hinton's solution [2] these simple network models became very popular, now they are used in the state of the art machine learning models, breaking records and in some tasks performing better than humans [6]. But most of the success and interest to the autoencoders can be attributed to the studies on supervised learning.

In this study, we will be looking at the other side and see how autoencoders perform compared to some of the well known traditional models. These algorithms will be k-Means [7] and DBSCAN [8].

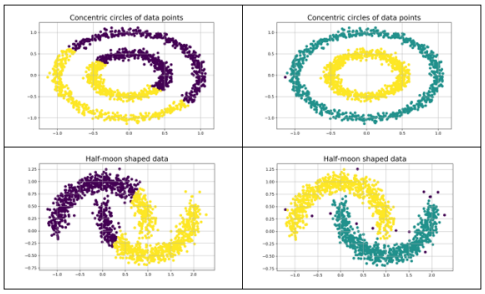

Fig. 1. Clusters created by k-Means and DBSCAN. First Column : k-Means, Second Column : DBSCAN
Adapted from CE888 lab6 results.

As it can be seen in Figure 1, k-Means algorithm can not cluster non-linear data. Results in [9] also confirm that, but it also shows us that the k-Means can be significantly improved if the dimensions of the data is reduced before the clustering procedure. In [10] a multilayer network is used to map high dimensional, non-linear features to low-dimensional space which is fine-tuned to work with KNN. They achieved 1.01% error rate using 7 nearest neighbours on MNIST dataset.

The study [11] gives a lot of insight on autoencoders and dimensionality reduction. They introduce the pre-training notion, which involves an effective way of initializing good weights and training the network layer by layer. Then they suggest fine-tuning the deep network as a whole. This practice can also be seen in [12] and [9], these studies also suggest pre-training. On the other hand, [13] concludes that the pre-training procedure does not improve the accuracy. Their experiments show that the pre-training, over multiple DNNs and

15 different datasets, on average reduce the accuracy. Another study [14] learns overcomplete representations of features for data reconstructions. It is shown in their results that this sparse, energy based autoencoder with linear layers can reproduce the MINST dataset with 0.7% error, while not requiring any preprocessing. [15] also follows a similar approach, in which a deep sparse network trained in unsupervised fashion to recognize faces, humans and objects. Finally, the last study we looked at implemented and compared different models (Figure 2). As a result [16] states that the best result obtained came from k-Means clustering.

TABLE I
TEST RECOGNITION ACCURACY (AND ERROR) FOR NORB DATASET. ADAPTED FROM : [16]

Algorithm	Accuracy (error)
Conv. Neural Network	93.4% (6.6%)
Deep Boltzmann Machine	92.8 (7.2%)
Deep Belief Network	95.0% (5.0%)
Deep neural network	97.13% (2.87%)
Sparse auto-encoder	96.9% (3.%)
Sparse RBM	96.2 (3.8%)
K-means clustering (4000 features)	97.21% (2.79%)

III. METHODOLOGY

Our main goal in this study is to correctly cluster data using unsupervised machine learning techniques. To accomplish this task we will go through three different subtasks. First, we will use traditional clustering algorithms to see how they perform on our datasets. In the second part, we will use autoencoders to learn good features from our datasets, then we will pass the-extracted features- outputs of the autoencoder to a clustering algorithm to see the difference in their performance. Finally, we will change the output activation function of the autoencoder to the softmax activation function, which will give us the probability distribution of the output vector. Then we will use the maximum weighted class as our cluster assignment.

Before starting the experiments the datasets will be examined and processed appropriately for each sub-task. For the first sub task, depending on the complexity and dimensions of the given data, a suitable clustering algorithm will be chosen. For the second sub-task, features of the datasets will be standardized. As LeCun concludes in [17], networks can learn better and faster if the inputs are centered around zero with equalized covariance. Another consideration for sub-tasks two and three is using ensemble of autoencoders. Since the optimization of the randomly initialized weights is difficult [11], we can have multiple autoencoders in our model to mitigate the effects of random weight initialization. For the last sub-task, since we will be using softmax as our activation function, the targets of our datasets will be converted to one hot coded versions.

The three datasets chosen to be clustered in this study are Modified NIST, Human Activity Recognition Using Smartphones (HAR) and High Time Resolution Universe Survey 2 (HTRU2).

MNIST is a image dataset of hand written digits. It is a modified version of the NIST dataset collected from Census Bureau employees and high-school students and processed by LeCun et.al..The original NIST set was normalized, so that each sample would be centered according to the weights of it's pixels in a 28x28 box. Images were also converted to grey-scale and shuffled to make it easy to use in machine learning applications [18]. The dataset consists of 60.000 training images and 10.000 test images. Since each sample is represented as a 28x28 pixel binary image, it can be said that samples have 784 features. MNIST dataset is frequently used in machine learning research[–], which makes it easier for researchers to compare their models and solutions to the others in the field.

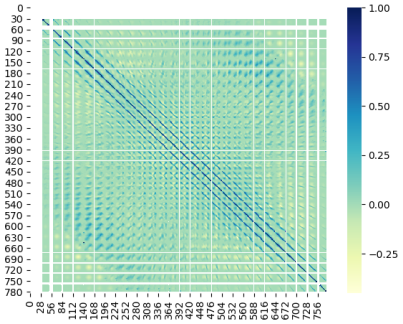

Fig. 2. Corraletion heatmap of MNIST features.

HAR is a dataset of multiple classes of human activities. It was created by recording different classes of activities of 30 volunteers by using the accelerometer and the gyroscope in a smartphone. These sensors were used to record the linear acceleration and angular velocity on three axes at 50 Hz. Volunteers were also recorded by a camera for the labeling of the activities(walking, walking upstairs, walking downstairs, sitting, standing, laying) [19]. The dataset consists of 7352 training samples and 2947 test samples. Each sample has 561 features which are- extracted- estimated from the recorded accelerometer and gyroscope signals.These features are normalized, scaled and shifted to be within [-1,1]. Subject id labels for the features and mentioned raw accelerometer and gyroscope signals are also provided with the dataset. Dimensions of the subject ids are the same with their training and testing parts. On the other hand, the raw signals are seperated into nine different files grouped under three axes, which are body acceleration (X,Y,Z), body gyroscope (X,Y,Z) and total acceleratin (X,Y,Z).

Our last dataset HTRU2 was obtained by Bates [20] using ANNs to first filter out the bad candidates,then manually inspecting the outputs of the ANN.Each sample of HTRU2 is a set of features extracted from the data acquired by large radio telescopes during High Time Resolution Universe Survey. HTRU2 has 8 features and consists of 17,898 total samples. Only 1,639 of them are positive pulsar samples and the 16,259 of them are negative samples. The first four features are simple statistics of the pulsar profiles and rest of the features were obtained from DM-SNR curve [21].These compact features

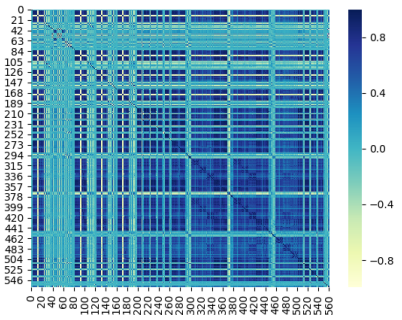

Fig. 3. Corraletion heatmap of HAR features.

were obtained from 90.000 labeled pulsar candidates by using Pulsar Feature Lab [22].

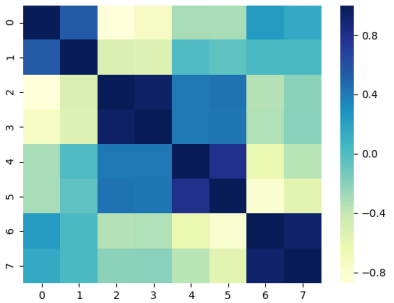

Fig. 4. Corraletion heatmap of htru features.

TABLE II
DATASET STATISTICS.

Datasets	Data Points	Features	Classes
MNIST	70.000	784	10
HAR	10.299	561	6
HTRU2	17,898	8	2

IV. EXPERIMENTS

In the experiments the implementation of the tasks and data analysis will be done using python. In the imlementation of the traditional clustering algorithms, sckit-learn library will be used because of its easy to use nature and rich functions. For the rest of the tasks, tensorflow and keras libraries will be used to implement autoencoders, since tensorflow converts the python code into C code in runtime, it is fast and it can run on GPUs, making it one of the best neural network library. Finally, the data analysis and plotting will be done using matplotlib and seaborn libraries.

A. Clustering Algorithms

In the core part of the clustering experiments, k-Means and DBSCAN algorithms will be used. These algorithms will be first compared to each other, then their results will be analysed, evaluated and recorded for future comparisons. If there is enough time to deviate from the core path, agglomerative clustering algorithm will also be used and these clustering algorithms will be tested again after applying dimensionality reduction methods like PCA/kernel PCA, LDA to the datasets. Given enough time, we will try these dimension reduction algorithms to confirm [23], in which it is concluded that the linear autoencoders with squared loss functions behave like PCAs. Otherwise, the experiments will continue with the autoencoders.

B. AutoEncoder and Softmax

Rest of the tasks share a similar model. Both of them will be using autoencoders to extract features. Implementation priority will be given to the main parts. First, an autoencoder will be trained and optimized to learn good features, then the output of these features will be passed to the mentioned clustering algorithms in the first task. Then, in the second part the output activations will be swapped with softmax activation function for clustering.

After implementing the minimum requirements, if there is enough time, various additional experiments will be performed. First, we will use linear autoencoder with one hidden layer for clustering to compare the performance to clustering with PCA. Then, we will perform the same experiment for non-linear autoencoder and kernel PCA. After the comparisons, we will try methods like Dropout [24] and denoising, to prevent one to one mapping and promote generalization [12]. Finally we will experiment with ensemble of autoencoders for more robust results.

C. Similar Work

In the study [11], autoencoders are compared to non-linear PCA and other dimensionality reduction methods. And it is stated that the autoencoders out perform PCA, logistic PCA, latent semantic analysis (LSA) and locar linear embedding techniques. The results of the reconstructions of MNIST digits from the study can be seen below.

Fig. 5. Top to bottom: Random test images, autoencoder reconstruction, logistic PCA reconstruction and standart PCA reconstruction.
Adapted from : [11]

Another study [9] uses autoencoders with dropout to extract good features for clustering. In their model, softmax is applied to the resulting output vector from the autoencoder and then they optimize the network by comparing the soft assignment to the target clusters. Their results on MNIST can be seen below.

TABLE III
CLUSTERING ACCURACY ON DIFFERENT SUBSAMPLES OF MNIST
ADAPTED FROM: [9]

Method	0.1	0.3	0.5	0.7	0.9
k-Means	47.14%	49.93%	53.65%	54.16%	54.39%
AE + k-Means	66.82%	74.91%	77.93%	80.04%	81.31%
DEC(proposed)	70.10%	80.92%	82.68%	84.69%	85.41%

V. DISCUSSION

At the end of the experiments, the evaluation of each model's cluster assignments will be calculated by using the V-measure. Since it takes into account both completeness and homogenity of the clusters, it will be our choice of evaluation [25]. On the other hand we will also use the Fowlkes-Mallows score. Because it allows us to compare the performance of our three different methods [26]. Also as an addition, we plan to evaluate the contingency matrix of each model, then perform McNemar's test between the models to see the significance of their difference.

This study will improve our understanding of autoencoders and it will give us a chance to observe how they perform compared to the traditional methods. It is assumed that the task of optimizing and fine tuning the encoders will be difficult and time consuming. Another benefit will be the comparison of the softmax clustering to the rest of the methods. Downside of this study will be understanding why the autoencoder does what it does, since it is a black box model. Luckily this is out of the scope of this study.

In a broader perspective, these experiments will improve our understanding of clustering, data manipulation and unsupervised learning in general. Understanding the data, its complexity and dimensions are not trivial yet very important. Even though Machine Learning and Data Science fields can provide many models and tools, it is difficult to get good results if we are not familiar with the data/problem we are dealing with.

VI. CONCLUSION

We briefly explained the purpose and the tasks of the project. While going through the literature, importance of unsupervised learning was understood. Also it was realized that the field needs more attention. Later on, we described our model and the datasets we are going to use. Then the experiments we are going to perform were described with references to the similar work.In the end the evaluation methods that we will use were explained together with what we expect to gain at the end of this work.

We believe that this work will provide a lot of insights and practical experiences about clustering, autoencoders and data analysis. With the addition of improving our data presentation skills, such as drawing graphs and plots through software. This will be a good study to put what we learned in theory to practice, do the experiments for ourselves and compare our results to the ones we found.

APPENDIX A

See page 6 for the GANTT chart of the assignment.

REFERENCES

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[2] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[4] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.

[5] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[6] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[7] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[8] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[9] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *International conference on machine learning*, 2016, pp. 478–487.

[10] R. Salakhutdinov and G. Hinton, "Learning a nonlinear embedding by preserving class neighbourhood structure," in *Artificial Intelligence and Statistics*, 2007, pp. 412–419.

[11] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[12] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of machine learning research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[13] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik, "Deep neural nets as a method for quantitative structure–activity relationships," *Journal of chemical information and modeling*, vol. 55, no. 2, pp. 263–274, 2015.

[14] C. Poultney, S. Chopra, Y. L. Cun *et al.*, "Efficient learning of sparse representations with an energy-based model," in *Advances in neural information processing systems*, 2007, pp. 1137–1144.

[15] Q. V. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. S. Corrado, J. Dean, and A. Y. Ng, "Building high-level features using large scale unsupervised learning," *arXiv preprint arXiv 1112.6209*, 2011.

[16] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 215–223.

[17] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[18] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[19] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones." in *Esann*, 2013.

[20] S. Bates, M. Bailes, B. Barsdell, N. Bhat, M. Burgay, S. Burke-Spolaor, D. Champion, P. Coster, N. D'Amico, A. Jameson *et al.*, "The high time resolution universe pulsar survey—vi. an artificial neural network and timing of 75 pulsars," *Monthly Notices of the Royal Astronomical Society*, vol. 427, no. 2, pp. 1052–1065, 2012.

[21] M. Keith, A. Jameson, W. Van Straten, M. Bailes, S. Johnston, M. Kramer, A. Possenti, S. Bates, N. Bhat, M. Burgay *et al.*, "The high time resolution universe pulsar survey–i. system configuration and initial discoveries," *Monthly Notices of the Royal Astronomical Society*, vol. 409, no. 2, pp. 619–627, 2010.

[22] R. Lyon, "Pulsar feature lab," 9 2015. [Online]. Available: https://figshare.com/articles/Pulsar_Feature_Lab/1536472

[23] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural networks*, vol. 2, no. 1, pp. 53–58, 1989.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[25] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007.

[26] E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings," *Journal of the American statistical association*, vol. 78, no. 383, pp. 553–569, 1983.

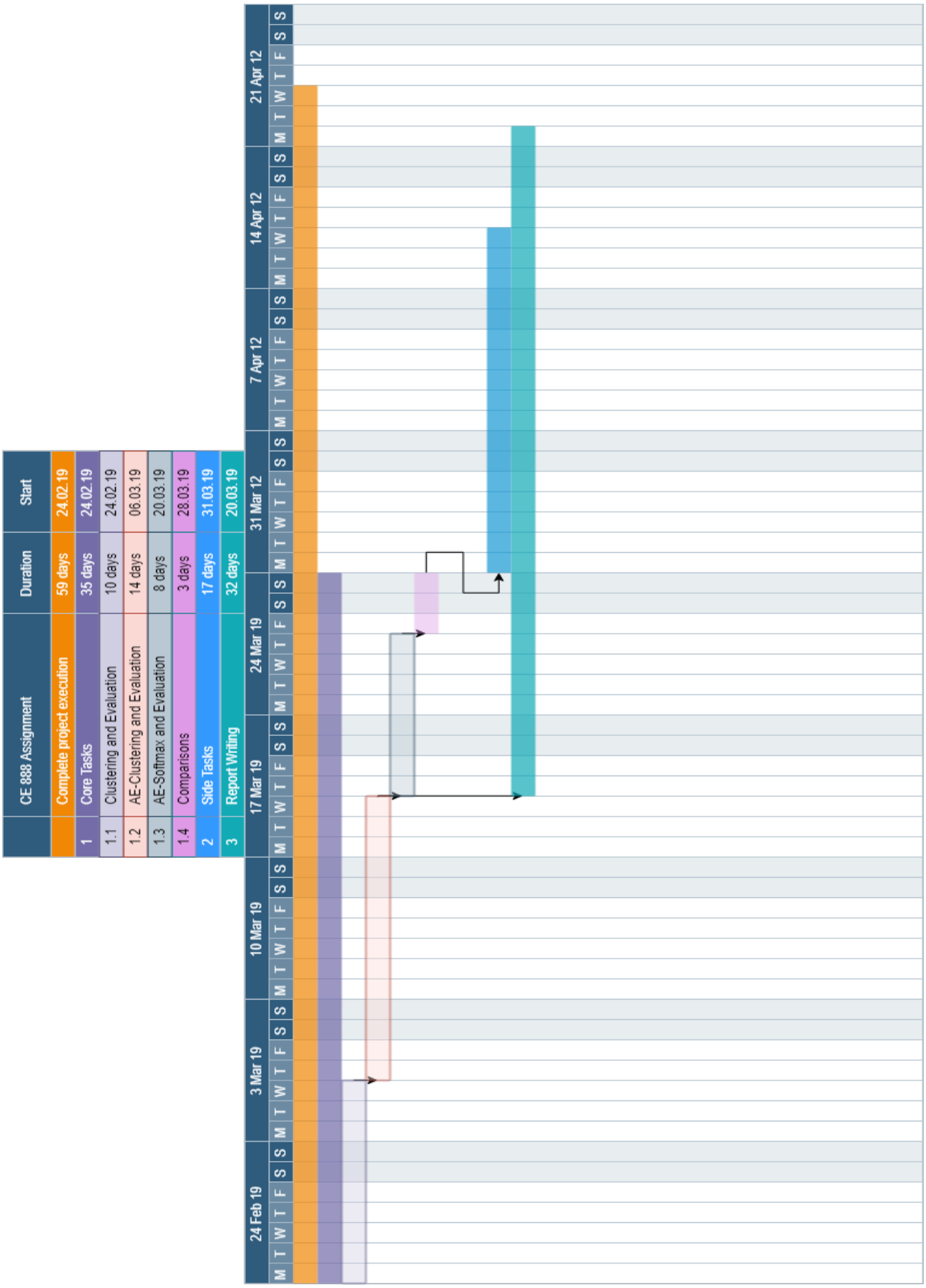

Fig. 6. Assignment GANTT Chart