Differential Evolution-Based Hyperparameter Optimization for Louvain-Based Graph Clustering in High-Dimensional Data

Mustafa Mahmood Zaidan[1], Muntadher Khamees[1] and Sameera Abubaker Saeed[2]
[1]*Department of Computer Science, College of Science, University of Diyala, 32001 Baqubah, Diyala, Iraq*
[2]*Department of Computer Science, Faculty of Computers and Information Technology, Hadhramout University, 50511 Mukalla, Yemen*
mmzk1986@gmail.com, alkarawis@uodiyala.edu.iq, smarei@hu.edu.ye

Keywords: Differential Evolution, Louvain Community Detection, Graph Construction Optimization, Resolution Parameter Tuning, High-Dimensional Clustering.

Abstract: The Louvain community detection and other graph-based clustering techniques are appealing to high-dimensional data since they have the ability to infer complicated structural relationships using similarity graphs. Nevertheless, their results are very sensitive to the choice of graph construction and the modularity resolution parameter which can give unstable partitions or under-clustering in the case of manually set hyperparameters. To overcome this shortcoming, this paper suggests a hybrid clustering model that combines the Louvain-based graph clustering with Differential Evolution (DE) to jointly optimize the hyperparameters. The framework proposed will optimize neighborhood size, embedding dimension, pruning quantile, and resolution simultaneously by using a single objective function that balances the modularity, silhouette score, and cluster balance, and a soft cluster-count regularization term is employed to encourage meaningful cluster granularity without enforcing a strict constraint on the number of clusters. In practice, to achieve the DE optimization step, it is done once offline on a representative subsample, with final clustering applied to the entire dataset using the optimized parameters. The results of experiments on five benchmark datasets (MFEAT, COIL-20, HAR, MNIST, and ISOLET) demonstrate that the proposed framework is stable in terms of outperforming the traditional clustering approaches and a fixed-parameter graph baseline. To illustrate, in MFEAT ARI increased by 0.52 to 0.65 and ACC increased by 0.54 to 0.67 whereas in ISOLET the detected cluster count increased by 10.0 to 24.7, which was much closer to the reference value of 26. The findings indicate a practical utility of the framework in high-dimensional clustering and the methodological importance of integrating joint graph-hyperparameter optimization with soft cluster-count regularization to achieve better clustering quality and control cluster-granularity.

1 INTRODUCTION

Among the wide range of available techniques, the graph-based clustering has obtained a lot of attention due to the ability to naturally represent complex and non-convex data distributions by similarity graphs. In particular, modularity-based community detection methods such as Louvain are very popular due to their efficiency and ability to handle a large scale network [1], [2].Despite their widespread use, graph-based methods for clustering are quite sensitive to the construction of the similarity graph. Choices such as neighborhood size, similarity metric, pruning strategy, embedding dimension and the resolution parameter can have a significant effect on the final clustering result. Prior research has shown that the construction of graph and the parameter setting play a critical role in the determination of the quality of clustering [3]-[5]. In addition, the choice of resolution parameters in methods based on modularity has a direct influence on the community structure that it detects and may lead to scale-dependent biases [6], [7]. The current approaches, though, are mainly based on predetermined settings or consider the graph-construction parameters and resolution tuning independently. Consequently, a gap still exists in creating a single framework that optimizes these factors together in addition to decreasing the under-clustering propensity of Louvain-based clustering. It is novel in that it combines Differential Evolution with Louvain-based graph clustering to co-optimize graph-construction hyperparameters and resolution along with a soft cluster-count regularization mechanism, in a single objective formulation. To overcome these drawbacks, in this article, we propose a hybrid graph-based clustering framework which includes community detection (Louvain community detection) and an optimization algorithm (Differential Evolution (DE)) for automated hyperparameter optimization [8]. Instead of tuning graph construction parameters manually, in our strategy neighborhood size, embedding dimension, pruning quantile and resolution is simultaneously optimized in a joint objective function [8] that balances quality of structure with geometrical compactness [1], [9], [10]. In addition we propose a soft cluster-count regularisation mechanism that encourages a meaningful level of cluster granularity, without imposing strict constraints [11], [12]. The contributions of this study are mainly as follows:

- We have developed a unified optimization framework that adjusts the parameters required for graph construction as well as the modularity resolution parameter in the Louvain clustering algorithm.
- We introduce a soft cluster-count regularization strategy to reduce under-clustering without imposing rigid constraints on K.
- We demonstrate consistent performance gains over traditional clustering methods and fixed-parameter graph baselines across five benchmark datasets with varying sizes and dimensionalities.

2 RELATED WORK

Among the modularity lead techniques, Louvain community detection is one of the most efficient scalability algorithms for large networks [1], [13]. Broad surveys have also pointed out the computational advantages of the graph clustering methods over the traditional partition-based algorithms [5], [14], but the effectiveness of the methods is strongly dependent on the way the graph is constructed and on how the resolution parameter is selected.

Recent attempts have been made to systematically explore the impact of neighborhood size, measures of similarity and pruning strategies on clustering results, confirming that graph construction has a decisive role in the obtained results [3]. Despite this understanding, there exists a great variety of practical implementations with fixed parameter settings and these can lead either to under-clustering or unstable community structures in different datasets. Differential Evolution (DE) in particular had shown good results in continuous global optimization problems [8]. Some researches have been done to apply the DE for the community detection problem or the parameter tuning of density-based clustering algorithms [15], [16]. However, these typically try to optimize the parameters of clustering separately from each other, without joint optimization of the choices for graph construction and modularity resolution, in a single and integrated manner. In contrast, in the proposed method the weighted graph construction, maximization of modularity and DE based hyperparameter optimization are integrated in a unified objective formulation. Furthermore, a soft cluster-count regularization mechanism is introduced in order to mitigate the sensitivity on the resolution parameter while there are no rigid constraints on the number of clusters.

3 PROPOSED METHOD

We propose a hybrid graph based clustering framework which combines weighted community detection and Differential Evolution (DE) for automatic hyperparameter optimization [8], [17].

The overall framework is made up of two main phases. The first is an offline optimization phase, in which Differential Evolution is used to jointly optimize the hyperparameters (k, d, q, r) with soft cluster-count regularization [8]. The second is an online execution phase, which is the feature representation, the weighted KNN graph construction, the connectivity enforcement and the Louvain-based community detection and post-processing [1], [3]. An overview of the proposed framework can be seen in Figure 1.

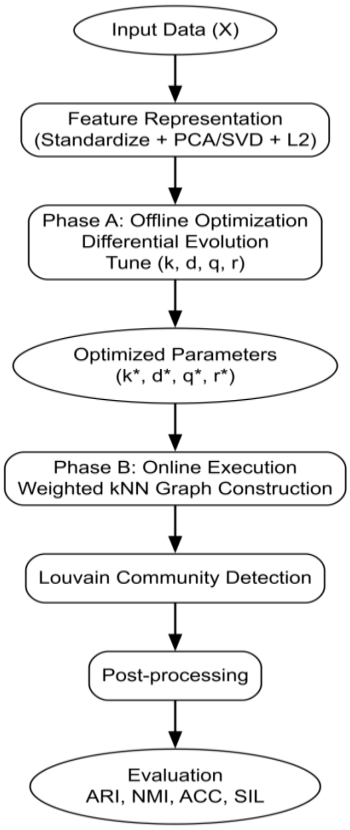

Figure 1: Overview of proposed DE-optimized Louvain graph clustering framework.

3.1 Weighted KNN Graph Construction

Given the normalized embedding $Z \in \mathbb{R}^{nxd}$, we construct the weighted k-nearest neighbor (kNN) graph with the cosine or euclidean distance. Distances are converted into similarity weights using Kernel bounded Cosine or Exponential.Two such symmetrization strategies are considered:

1) Symmetric kNN: Both the edges are averaged.
2) mutual kNN Only reciprocal neighbor relations are maintained.

After building the graph, some edges turn out to be very weak. These weak links are removed based on a quantile threshold q, since keeping them does not really help the structure. At the same time, we make sure that no node becomes completely isolated; if that happens, at least one connection is kept. This step mainly helps to reduce the sensitivity of the graph to noisy or insignificant relations. All graph-related operations are carried out using the NetworkX library.

3.2 Connectivity Enforcement

Edge pruning may cause disconnected components, under an adverse impact on optimization of modularity [1]. In order to maintain the coherence of the structure, disconnected parts are interconnected by light-weighted bridging edges according to minimum distance between components in the embedding space.

3.3 Clustering and Post-processing of Graph

Louvain modularity maximization with resolution parameter r [1], [4] is used to identify communities. This parameter determines granularity of clusters and thus influences the communities identified. Once the clustering is done, very small clusters ($size < max(3{,}0.5\% \ x \ n)$) are reallocated to the nearest large cluster in the embedding space. This consolidation process is added to minimize over-fragmentation and enhance robustness.

3.4 DE-Based Hyperparameter Optimization

To optimize the four hyperparameters of the proposed framework, that is $\theta = (k, d, q, r)$ where:

- k is neighborhood size;
- d is embedding dimension;
- q is pruning quantile;
- r is resolution parameter.

Search ranges are: $k \in [8{,}80]$, $d \in [10{,}160]$, $q \in [0{,}0.4]$, $r \in [0.05{,}0.9]$.

DE is taken due to its performance in the context of the continuous global optimization [8]. Graph construction and modularity resolution are co-optimized in the proposed framework with one objective function.

Fitness Function:

$$F(\theta) = w_1 Q(\theta) + w_2 S(\theta) + w_3 B(\theta) - P(\theta)$$

Q, S, and B denote modularity [1], silhouette score [10], and a normalized entropy-based cluster balance term, respectively.

The penalty term P penalizes the disconnected graphs, excessive number of clusters and degenerate small clusters. Weights are the same for all the data sets and is fixed, i.e. $w1 = 0.6$, $w2 = 0.3$ and $w3 = 0.1$. This way a balance is achieved between the structural quality and the geometric compactness.

Soft Clustering Count Regularization. A tolerance range $[(1 - alpha) * C\_target, (1 + alpha) * C\_target]$ is defined and deviations outside this band are penalized (quadratically). In our experiments $\alpha = 0.15$, allowing $a \pm 15\%$ margin. This mechanism is unsupervised flexible and optimized for decent granularity of clusters. Effectiveness of the proposed framework is judged based on the evaluation of the performance on benchmark data in Section 4.

4 EXPERIMENTAL SETUP

4.1 Datasets and Baselines

We test the proposed framework on five benchmark data sets and these data sets are publicly available on UCI Machine Learning Repository [18] : MFEAT, HAR, ISOLET, and MNIST and a COIL-20 data set from the Columbia Object Image Library[19]. As in Table 1, these datasets vary between size, dimensionality and number of classes and therefore, provide a balanced evaluation of medium and large-scale clustering scenarios. Ground-truth labels are only used for purposes of external evaluation, neither in construction of the graph, optimization of parameters, nor decision making in clustering

Table 1: Dataset statistics.

Dataset	Samples	Features	Classes
MFEAT	2000	649	10
COIL-20	1440	1024	20
HAR	10299	561	6
MNIST	70000	784	10
ISOLET	6006	617	26

The performance is clustered after judging on the basis of:

- Adjusted Rand Index (ARI) [20].
- Normalized Mutual Information (NMI) [21].
- Clustering Accuracy (ACC).
- Silhouette Score (SIL) [10].

These metrics provide complementary perspectives of partition agreement, geometric cohesion and label agreement. We benchmark the proposed approach against four sample baselines:

1) KMeans.
2) Gaussian Mixture Model (GMM).
3) Agglomerative clustering with average linkage and cosine distance.
4) Graph Baseline (optimization of the Louvain no DE).

KMeans, GMM and Agglomerative clustering is used on the standard configurations in scikit-learn [22].Graph Baseline Theoretically, the Graph Baseline is the same weighted kNN construction pipeline as in Section 3 but with more fixed hyperparameters: $k = 35$, $d = 40$, $q = 0.10$, $r = 0.25$. These values were selected as constant mid-range values in the range of search and reflect values commonly used as the size of neighborhoods and the value of resolutions in graph clustering algorithms in Louvain [1], [5].

4.2 Implementation Details

The experiments are all written in Python and made with scikit-learn [22], SciPy and NetworkX. In the case of the proposed approach, Differential Evolution (DE) [8] is used in acquiring an optimal value of the four hyperparameters (*k, d, q, r*), depending on the fitness formulation presented in Section 3.The fitness weights are also fixed on all the datasets, $w\_mod = 0.6$, $w\_sil = 0.3$ and $w\_bal = 0.1$. The values offer a tradeoff between structural quality (modularity), geometrical compactness (silhouette) and cluster balance. DE optimization is carried out once in the case of the offline run in fixed 800 instances as a subsample of the total population of all the datasets (maxiter = 15, popsize = 8), such that the optimization budget will be identical across all benchmarks. The stage of DE is done once offline on a subsample of 800 cases whereas the online clustering runtime is comparable to the graph baseline. Final clustering is retrieved on the entire dataset based on the parameter acquired. In benchmark evaluation soft cluster-count regularization is enabled with $\alpha = 0.15$ and $\lambda\_c\_target = 3.0$. This introduces a ±15% tolerance band around the known reference cluster count and assists to optimize the process without possessing strict limits on the value. The average performance of 5 independent runs is all reported result. Individual measurements of runtime are made of:

1) classification of time on all the data;
2) DE optimization time of the subsampled data.

This division relies on practical deployment environments, in which the parameter tuning is in an offline phase, and clustering is also conducted during inference. All the experiments were conducted on a usual desktop computer CPU (Intel Core i7, 16GB RAM).

Table 2: Performance comparison (average of 5 runs) clustering.

Dataset	Method	ARI	NMI	ACC	SIL	C	Clustering Time (s)	Offline DE Time (s)
MFEAT(K=10)	KMeans	0.43	0.42	0.47	0.21	–	0.18	
	GMM	0.38	0.49	0.54	0.21	–	0.48	
	Agglo (cosine)	0.41	0.50	0.54	0.24	–	0.55	
	Graph Baseline	0.52	0.54	0.54	0.24	8.0	0.95	
	Graph + DE (Ours)	0.65	0.66	0.67	0.26	10.2	1.05	110
COIL-20 (K=20)	KMeans	0.45	0.43	0.38	0.12	–	0.14	
	GMM	0.32	0.30	0.27	0.10	–	0.36	
	Agglo (cosine)	0.40	0.43	0.38	0.22	–	0.44	
	Graph Baseline	0.47	0.51	0.50	0.16	18.5	0.78	
	Graph + DE (Ours)	0.56	0.62	0.60	0.16	19.8	0.85	105
HAR (K=6)	KMeans	0.36	0.52	0.50	0.14	–	2.6	
	GMM	0.33	0.48	0.47	0.13	–	7.8	
	Agglo (cosine)	0.38	0.55	0.52	0.24	–	9.5	
	Graph Baseline	0.44	0.59	0.54	0.21	5.0	14.8	
	Graph + DE (Ours)	0.49	0.60	0.58	0.23	5.2	16.2	135
MNIST (K=10)	KMeans	0.44	0.56	0.58	0.18	–	2.2	
	GMM	0.40	0.55	0.52	0.14	–	6.9	
	Agglo (cosine)	0.37	0.54	0.48	0.12	–	8.3	
	Graph Baseline	0.44	0.58	0.53	0.15	8.0	15.5	
	Graph + DE (Ours)	0.47	0.59	0.62	0.16	9.8	17.0	140
ISOLET (K=26)	KMeans	0.20	0.45	0.28	0.08	–	1.4	
	GMM	0.18	0.42	0.26	0.07	–	4.1	
	Agglo (cosine)	0.22	0.45	0.30	0.06	–	5.2	
	Graph Baseline	0.26	0.51	0.34	0.10	10.0	6.8	
	Graph + DE (Ours)	0.32	0.55	0.41	0.12	24.7	7.5	120

Note: Time represents, Clustering + DE Optimization. DE is a single offline cost (subsample n=800). C = cluster count that is detected (ground truth K in parentheses). Performance averages of 5 independent runs are taken as all values.

5 RESULTS AND DISCUSSION

Table 2 displays the clustering performance of each of the methods on the 5-benchmark data. Performance averages of 5 independent runs are taken as all values.

5.1 Overview Analysis of the General Performance

The proposed Graph + DE framework tends to be better than the fixed-parameter Graph Baseline in the datasets under consideration. This implies that graph-based clustering can be enhanced by automatic hyperparameter optimization to generate a more suitable graph representation, and a more appropriate clustering granularity, without modifying the underlying Louvain procedure itself. The most evident improvements are on MFEAT and COIL-20, in which the baseline settings are less capable of capturing the underlying structure. ARI goes up on MFEAT, but NMI and ACC also improve, suggesting that the measured increase is not confined to a single measure, but represents a more consistent partition on the whole. The same trend is followed in the COIL-20 where ARI and ACC increase to 0.56 and 0.60 respectively. These findings imply that joint optimization of the graph-construction parameters and a graph-resolution can be especially useful in the case that the fixed settings fail to capture the local structure of the data.

5.2 Work with Large or Complex Data

ISOLET is an even more difficult clustering problem since it has quite a number of classes and a high dimensionality. In this case, graph-construction settings with fixed settings can be less apt to give optimal partitions. The fact that the ARI has improved by 0.26 to 0.32 and ACC has improved by 0.34 to 0.41 means that the optimization strategy proposed can be still useful even in a case when the task of clustering becomes more challenging. This implies that the DE-based search can find parameter combinations that can better maintain neighborhood relations and enhance partition quality in more complex data environments. Hence, the suggested algorithm is not just advantageous on mid-sized datasets yet demonstrates useful scalability on the more challenging high-dimensional reference groups.

5.3 Cluster Count Correction

Less under-clustering is one of the most significant impacts of the proposed framework. The Graph Baseline is not as likely to identify as many clusters as would be predicted in a number of datasets, including MFEAT, with C = 8 rather than K = 10, and ISOLET, with C = 10 rather than K = 26. Following DE optimization with soft cluster-count regularization, the numbers of clusters detected shift significantly towards the reference values, to C = 10.2 on MFEAT and C = 24.7 on ISOLET. It means that the proposed framework is better at cluster granularity, as a joint adjustment of graph construction and resolution is made, and the soft regularization term is used to steer the search towards overly coarse partitions without a strict Exact-K constraint.

5.4 Runtime Considerations

The run time of the optimized model is similar to that of the Graph Baseline on all datasets. In fact, in the case of MFEAT, time clustering gain only a very little value, between 0.95 s to 1.05 s. The DE optimization step is the primary added cost and takes about 105-140 s with respect to the dataset. Nonetheless, this price is paid only once during an offline phase on a subsample of 800 instances, but the ultimate clustering of the entire dataset is computationally similar to the baseline. This trade-off is significant in practice: the algorithm yields more accurate partitions and fidelity to cluster counts and retains the online clustering phase efficient. On the whole, the findings indicate that co-optimization of graph construction and modularity resolution can offer significant returns in terms of performance at a reasonable computational cost.

6 CONCLUSIONS

The present paper described a hybrid graph-based clustering model, which is founded on the Louvain community detection algorithm and the Differential Evolution (DE) algorithm to create dynamic hyperparameter optimization. Rather than having to manually select parameters of the graph construction, the proposed one optimally shapes both the neighborhood size, embedding dimension, pruning level and resolution with a single objective function balanced between the modularity and a geometric compactness.

The experimental study of five benchmark data sets proves the same growth as compared to the traditional clustering algorithms and a fixed parameter graph baseline. The improvements are particularly in the improvement of the under-clustering issue, which traditionally occurs in the modularity-based techniques [1]. Moreover, the proposed soft cluster-count regularization is also beneficial in the sense that it directs the optimization to the amount of cluster granularity that is significant but unconstrained resulting in better structural correspondence to ground-truth labels.

The greatest weakness of the current framework is that it tunes the parameters using a fixed subsample, which may be biased in terms of having an entirely different parameter fit. The adaptive subsampling schemes will also be studied in the future, and the model will be extended to bigger and more dynamic graph clustering environment, as the modern development of fast and large-scale community detection proceeds [5].

REFERENCES

[1] V. D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks,” vol. 10008, 2008, [Online]. Available: https://doi.org/10.1088/1742-5468/2008/10/P10008.

[2] Z. Zhang, P. Pu, D. Han, and M. Tang, “Self-adaptive Louvain algorithm: fast and stable community detection algorithm based on the principle of small probability event,” Physica A, 2018, [Online]. Available: https://doi.org/10.1016/j.physa.2018.04.036.

[3] Z. Ait, E. Mouden, I. El-fedany, and M. Lahmer, “A Systematic Study of Graph Construction and Parameter Sensitivity in Clustering,” vol. 18, no. 8, 2025, [Online]. Available: https://doi.org/10.22266/ijies2025.0930.54.

[4] S. Li, K. Liu, M. Zheng, and L. Bai, “Multi-view spectral clustering algorithm based on bipartite graph and multi-feature similarity fusion,” Neural Networks, vol. 194, p. 108177, 2026, [Online]. Available: https://doi.org/10.1016/j.neunet.2025.108177.

[5] J. Xue et al., “A comprehensive survey of fast graph clustering,” Vicinagearth, pp. 1-22, 2024, [Online]. Available: https://doi.org/10.1007/s44336-024-00008-3.

[6] J. Reichardt and S. Bornholdt, “Statistical mechanics of community detection,” Phys. Rev. E, vol. 74, no. 1, p. 16110, Jul. 2006, [Online]. Available: https://doi.org/10.1103/PhysRevE.74.016110.

[7] A. Arenas, A. Fernández, and S. Gómez, “Analysis of the structure of complex networks at different resolution levels,” New J. Phys., vol. 10, no. 5, p. 53039, 2008, [Online]. Available: https://doi.org/10.1088/1367-2630/10/5/053039.

[8] R. Storn and K. Price, “Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces,” pp. 341-359, 1997.

[9] S. Fortunato and M. Barthélemy, “Resolution limit in community detection,” Proc. Natl. Acad. Sci., vol. 104, no. 1, pp. 36-41, Jan. 2007, [Online]. Available: https://doi.org/10.1073/pnas.0605965104.

[10] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” J. Comput. Appl. Math., vol. 20, no. C, pp. 53-65, 1987, [Online]. Available: https://doi.org/10.1016/0377-0427(87)90125-7.

[11] R. Tibshirani, G. Walther, and T. Hastie, “Estimating the Number of Clusters in a Data Set Via the Gap Statistic,” J. R. Stat. Soc. Ser. B Stat. Methodol., vol. 63, no. 2, pp. 411-423, Jul. 2001, [Online]. Available: https://doi.org/10.1111/1467-9868.00293.

[12] F. Iglesias, T. Zseby, and A. Zimek, “Clustering refinement,” Int. J. Data Sci. Anal., vol. 12, no. 4, pp. 333-353, 2021, [Online]. Available: https://doi.org/10.1007/s41060-021-00275-z.

[13] V. Blondel, J. L. Guillaume, and R. Lambiotte, “Fast unfolding of communities in large networks: 15 years later,” J. Stat. Mech. Theory Exp., vol. 2024, no. 10, 2024, [Online]. Available: https://doi.org/10.1088/1742-5468/ad6139.

[14] L. Ding, C. Li, D. Jin, and S. Ding, “Survey of spectral clustering based on graph theory,” Pattern Recognit., vol. 151, p. 110366, 2024, [Online]. Available: https://doi.org/10.1016/j.patcog.2024.110366.

[15] C. Liu and Q. Liu, “Community Detection Based on Differential Evolution Using Modularity Density,” 2018, [Online]. Available: https://doi.org/10.3390/info9090218.

[16] O. Ajmal et al., “Enhanced Parameter Estimation of DENsity CLUstEring (DENCLUE) Using Differential Evolution,” Mathematics, vol. 12, no. 17, 2024, [Online]. Available: https://doi.org/10.3390/math12172790.

[17] B. A. Attea et al., “A review of heuristics and metaheuristics for community detection in complex networks: Current usage, emerging development and future directions,” Swarm Evol. Comput., vol. 63, p. 100885, 2021, [Online]. Available: https://doi.org/10.1016/j.swevo.2021.100885.

[18] D. Dua and C. Graff, “UCI Machine Learning Repository,” 2019, [Online]. Available: https://archive.ics.uci.edu/.

[19] H. Nene, S. A. Nayar, and S. K. Murase, “Columbia Object Image Library (COIL-20),” 1996, [Online]. Available: https://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php.

[20] L. Hubert, “Classification,” vol. 218, pp. 193-218, 1985.

[21] A. Strehl, “Cluster Ensembles - A Knowledge Reuse Framework for,” vol. 3, pp. 583-617, 2002.

[22] F. Pedregosa, R. Weiss, and M. Brucher, “Scikit-learn: Machine Learning in Python,” vol. 12, pp. 2825-2830, 2011.