minimizing the sum of dissimilarities between each object and its corresponding reference point. The algorithm randomly chooses the k objects in dataset D as initial representative objects called medoids. A medoid can be defined as the object of a cluster, whose average dissimilarity to all the objects in the cluster is minimal i.e. it is a most centrally located point in the given data set. Then for all medoid, after every assignment of a data object to particular cluster the new medoid is decided. The problem is K-medoids does not generate the same result with each run,

Algorithm: Document Clustering: *k*-medoids
Input:
A Collection of Documents $\{D_i\}$,
Number of Representatives *K*.
Output:
A set of medoid documents $D_{C1},\ldots, D_{Ck.}$

1. Randomly select *k* documents as the initial cluster centers.
2. For each document D_i, do, Assign its membership to the cluster C_j that has the largest similarity. sim(D_i, D_{cj});
3. Find the most centrally located document in each cluster.
4. Repeat 2 & 3 till small change in total sum of similarity.
5. Return.

C. Single Link Algorithm

Single Link algorithms uses bottom-up strategy. It compares each point with each point. In this, initially, every object belongs to the different cluster. With iteration, we merge the closest clusters, till some condition is satisfied. Fig (3) explains this algorithm.

- The similarity between a pair of clusters:
- The similarity between the most similar pair of documents, one of which appears in each cluster
- Each cluster member will be more similar to at least one member in that same cluster than to any member of another cluster
- Single-link clustering tends to produce a small number of large, poorly linked clusters

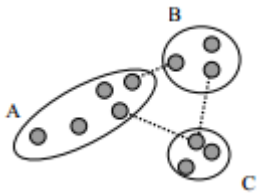

Figure 2: Clustering in single link algorithm

We combine the two clusters whose shortest distance is the smallest: A and B

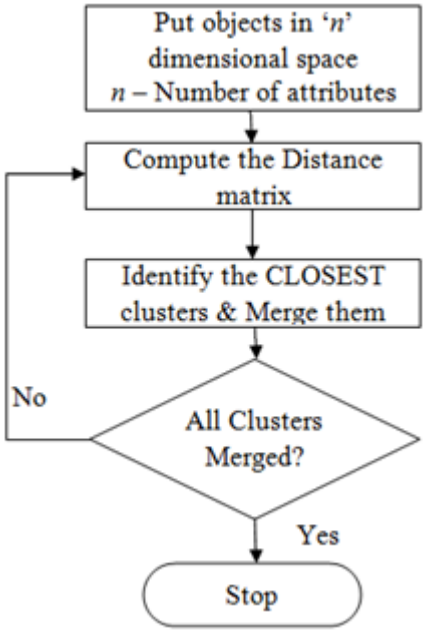

Figure 3: Single Link Algorithm flowchart

D. Complete Link Algorithm

Complete Link algorithm is almost identical to the single link algorithm. The only difference is that, complete link algorithm chooses the distant pair of clusters to merge with iteration. Fig (4) shows this algorithm.

- The similarity between the least similar pair of documents from the two clusters
- Each cluster member is more similar to the most dissimilar member of that cluster than to the most dissimilar member of any other cluster
- Complete-link clustering produces a larger number of small, tightly linked clusters.

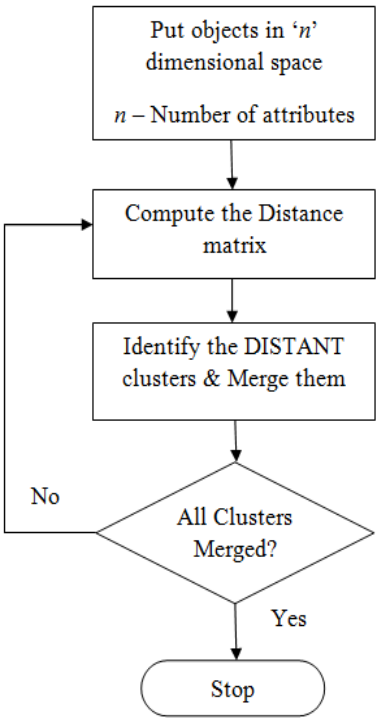

Figure 4: Complete Link algorithm flowchart

We combine the two clusters whose longest distance is the smallest: B and C as shown in the fig (5).

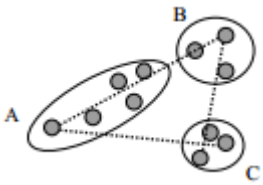

Figure 5: Clustering in Complete Link algorithm

E. Average Link Algorithm

In Average link algorithm, the distance needed to merge the clusters is the average distance between all the objects from one cluster to every object from other cluster. Fig (5) shows the flow of algorithm.

Each cluster member has a greater average similarity to the other members of its cluster than it does to all members of any other cluster

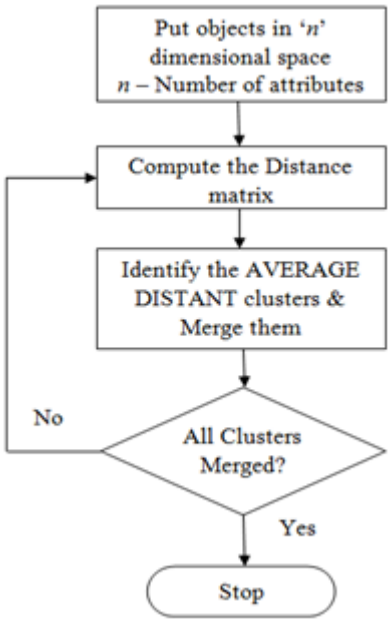

Figure 6: Average Link Algorithm flowchart

F. Cluster-based Similarity Partitioning Algorithm (CSPA)

If two objects belong to the same cluster, they considered as the similar object, if not, then considered as dissimilar. This is the simple logic behind CSPA. In similarity matrix of this algorithm, similarity of object is denoted by 1, and otherwise 0. Thus, for every clustering, an *n*n* binary similarity matrix is created.

3. Literature Review

There are just a couple of studies reporting the utilization of grouping calculations in the Machine Legal sciences field. Basically, the vast majority of the studies portray the utilization of excellent calculations for grouping information e.g., Desire Boost (EM) for unsupervised learning of Gaussian Mixture Models, K-implies, Fuzzyc-implies (FCM), and Orchestrating toward oneself Maps (SOM). These calculations have well-known properties and are generally utilized as a part of practice. For example, K-means and FCM can be seen as specific instances of EM [CM Bishop]. Calculations like SOM [16], in their turn, by and large have inductive inclinations like K-means, however are generally less computationally proficient. In [3], SOM-based calculations were utilized for bunching records with the point of settling on the choice making procedure performed by the analysts more effective. The records were bunched by considering their creation dates/times and their augmentations. This sort of calculation has additionally been utilized as a part of [12] with a specific end goal to bunch the results from essential word looks. The underlying presumption is that the bunched results can build the data recovery productivity, on the grounds that it would not be important to survey all the reports found by the client any longer.

A coordinated environment for mining messages for scientific examination, utilizing order and grouping calculations, was introduced in [13]. In a related application space, messages are gathered by utilizing lexical, syntactic, structural, and area particular gimmicks [7]. Three grouping calculations (K-means, Bisecting K-means and EM) were utilized. The issue of grouping messages for legal investigation was additionally tended to in [15], where a Piece based variation of K-means was connected. They got results were investigated subjectively, and the creators inferred that they are fascinating and helpful from an examination point of view. All the more as of late [8], a FCM-based system for mining affiliation tenets from scientific information was depicted. The writing on Machine Criminology just reports the utilization of calculations that accept that the quantity of groups is known and altered from the earlier by the client. Went for unwinding this presumption, which is frequently improbable in viable applications, a typical approach in different areas includes evaluating the quantity of groups from information. Basically, one affects diverse information segments (with distinctive quantities of groups) and afterward evaluates them with a relative legitimacy record to gauge the best esteem for the quantity of bunches [1], [4], and [11]. This work makes utilization of such routines, consequently conceivably encouraging the work of the master inspector who in practice would scarcely know the quantity of bunches from the earlier.

Clustering algorithms are studied for several years, along with the literature on the subject is huge. Therefore, we thought we would choose some (six) representative algorithms in an effort to show the potential of the proposed approach, namely: the partitioned K-means [1] and K-medoids [10], the hierarchical Single, Complete or Average Link [14], along with the cluster ensemble algorithm referred to as CSPA [2]. These algorithms were run with some other mixtures of their parameters. Thus, like a contribution of our own work, we compare their relative performances to the studied application domain; using five real-world investigation cases conducted with the Brazilian Federal Police Department. So as to make the comparative research into the algorithms more realistic, two relative validity indexes (Silhouette [10] and its particular simplified version [6]) are familiar with estimate the sheer numbers of

clusters automatically from data. It really is well-known that the sheer numbers of clusters is a significant parameter of several algorithms and it also can be quite a priori unknown. In terms of we understand, however, the automated estimation of the sheer numbers of clusters hasn't been investigated within the computer Forensics literature.

4. Conclusion

We presented an approach that applies document clustering methods to forensic analysis of computers seized in police investigations. Also, we reported and discussed several practical results that can be very helpful for researchers and practitioners of forensic computing. More specifically, inside our experiments the hierarchical algorithms referred to as Average Link and Complete Link presented the best results. Despite their usually high computational costs, we demonstrate that they're particularly suitable for the studied application domain as the dendrograms that they offer summarized views of the documents being inspected, thus being helpful tools for forensic analyzers that analyze textual documents from seized computers. As already observed in other application domains, dendrograms provide very informative descriptions and visualization capabilities of data clustering structures [14]. The partitioned K-means and K-medoids algorithms also achieved accomplishment when properly initialized. Taking into consideration the approaches for estimating the number of clusters, the relative validity criterion referred to as silhouette has proven to be more accurate than its (more computationally efficient) simplified version. Additionally, some of our results suggest that using the file names combined with the document content information may be helpful for cluster ensemble algorithms. Above all, we observed that clustering algorithms indeed tend to induce clusters formed by either relevant or irrelevant documents, thus adding to boost the expert analyzer's job. Furthermore, our evaluation of the proposed approach in five real-world applications shows so it has the potential to speed up the computer inspection process.

Directed at further leveraging the use of data clustering algorithms in similar applications, a promising venue for future work involves investigating automatic approaches for cluster labeling. The assignment of labels to clusters may enable the expert analyzer to spot the semantic content of every cluster more quickly, eventually even before analyzing their contents. Finally, the analysis of algorithms that induce overlapping partitions (e.g., Fuzzy C-Means and Expectation-Maximization for Gaussian Mixture Models) may be worth of investigation.

References

[1] A. K. Jain and R. C. Dubes, "Algorithms for Clustering Data", Englewood Cliffs, NJ: Prentice-Hall, 1988.
[2] A. Strehl and J. Ghosh, "Cluster ensembles: A knowledge reuse framework for combining multiple partitions," J. Mach. Learning Res., 2002.
[3] B. K. L. Fei, J. H. P. Eloff, H. S. Venter, andM. S. Oliver, "Exploring forensic data with self-organizing maps," in Proc. IFIP Int. Conf. Digital Forensics, 2005.
[4] B. S. Everitt, S. Landau, and M. Leese, "Cluster Analysis", London, U.K.: Arnold, 2001.
[5] C. M. Bishop, "Pattern Recognition and Machine Learning", New York: Springer-Verlag, 2006.
[6] E. R. Hruschka, R. J. G. B. Campello, L. N. de Castro, "Evolving clusters in gene-expression data", Inf. Sci., 2006.
[7] F. Iqbal, H. Binsalleeh, B. C. M. Fung, and M. Debbabi, "Mining writeprints from anonymous e-mails for forensic investigation," Digital Investigation, Elsevier, 2010.
[8] K. Stoffel, P. Cotofrei, and D. Han, "Fuzzy methods for forensic data analysis," in Proc. IEEE Int. Conf. Soft Computing and Pattern Recognition, 2010.
[9] L. F. Nassif and E. R. Hruschka, "Document clustering for forensic computing: An approach for improving computer inspection," in Proc. Tenth Int. Conf. Machine Learning and Applications (ICMLA), 2011,
[10] L. Kaufman and P. Rousseeuw, "Finding Groups in Gata: An Introduction to Cluster Analysis",
[11] Hoboken, NJ: Wiley-Interscience, 1990.
[12] L. Vendramin, R. J. G. B. Campello, and E. R. Hruschka, "Relative clustering validity criteria: A comparative overview," Statist. Anal.Data Mining, 2010.
[13] N. L. Beebe and J. G. Clark, "Digital forensic text string searching: Improving information retrieval effectiveness by thematically clustering search results," Digital Investigation, Elsevier, 2007.
[14] R. Hadjidj, M. Debbabi, H. Lounis, F. Iqbal, A. Szporer, and D. Benredjem, "Towards an integrated e-mail forensic analysis framework," Digital Investigation, Elsevier, 2009.
[15] R. Xu and D. C.Wunsch, II, "Clustering", Hoboken, NJ: Wiley/IEEE Press, 2009.
[16] S. Decherchi, S. Tacconi, J. Redi, A. Leoncini, F. Sangiacomo, and R. Zunino, "Text clustering for digital forensics analysis," Computat.Intell. Security Inf. Syst., 2009.
[17] S. Haykin, "Neural Networks: A Comprehensive Foundation", Englewood Cliffs, 1998.