Tracking The Sensitivity of The Learning Models Toward Exact and Near Duplicates

Menna Ibrahim Gabr[1], Yehia Helmy[2], Doaa S. Elzanfaly[3]
Business Information Systems (BIS)-Dept.-Faculty of Commerce and Business Administration, Helwan University, Egypt[1, 2]
Faculty of Information and Artificial Intelligence, Helwan University, Egypt[3]
Faculty of Computer and Informatics, British University in Egypt, Egypt[3]

***Abstract*—Most real-world datasets contaminated by quality issues have a severe effect on the analysis results. Duplication is one of the main quality issues that hinder these results. Different studies have tackled the duplication issue from different perspectives. However, revealing the sensitivity of supervised and unsupervised learning models under the existence of different types of duplicates, deterministic and probabilistic, is not broadly addressed. Furthermore, a simple metric is used to estimate the ratio of both types of duplicates regardless of the probability by which the record is considered duplicate. In this paper, the sensitivity of five classifiers and four clustering algorithms toward deterministic and probabilistic duplicates with different ratios (0% - 15%) is tracked. Five evaluation metrics are used to accurately track the changes in the sensitivity of each learning model, MCC, F1-Score, Accuracy, Average Silhouette Coefficient, and DUNN Index. Also, a metric to measure the ratio of probabilistic duplicates within a dataset is introduced. The results revealed the effectiveness of the proposed metric that reflects the ratio of probabilistic duplicates within the dataset. All learning models, classification, and clustering models are differently sensitive to the existence of duplicates. RF and Kmeans are positively affected by the existence of duplicates which means that their performce increase as the percentage of duplicates increases. Furthermore, the rest of classifiers and clustering algorithms are sensitive toward duplicates existence, especially within high percentage that negatively affect their performance.**

Keywords*—*Deduplication; deterministic duplicates; probabilistic duplicates; supervised learning models; unsupervised learning models; evaluation metrices

I. INTRODUCTION

Data quality has been an active research area that affects different domains. Many data quality dimensions have been addressed through the literature[1]–[5], Data Duplication has been considered as one of the most intriguing dimensions. Data duplication is defined as multiple representation of the same real world object or a measure of undesirable duplicates within a certain field, record or dataset[6]. The duplication can be found in two different types, the Deterministic and the Probabilistic duplications[7], [8]. The Deterministic (exact) duplication, where two records or more are identical and Probabilistic (near/fuzzy) duplication, where multiple records are nonidentical and refer to the same real world entity[9], [10].

Duplicates can occur due to two main causes. The intra source duplicates, and the inter source duplicates[11]. The intra source duplicates occurs when a single data object can be entered many times into the same database. Whereas the duplicates that appear while merging multiple data source are called inter source duplicates[11]. The process of removing the intra source duplicates is called deduplication[12] which is the main scope of this paper. Whereas removing duplicates from inter source duplicates is called Record Linkage[13].

Data quality dimensions are assessed to evaluate by how much the data is qualified for a task at hand. Dimensions are measured either objectively or subjectively. Subjective measurements are based on consumers' opinions like questionnaires, and surveys. Whereas Objective measurements are used to give a simple ratio between the undesirable outcomes and the total[14]. For example, to calculate the percentage of duplicates in the dataset, the following equation can be used (number of duplicate records/Total number of records). This simple metric can be perfectly used within exact duplicates, however, in the case of near duplicates the probability of these duplicate records should be considered to reflect the true percentage of duplicates within a dataset.

The research effort in duplication area is diverse. However, little work has focused on clarifying the effect of both data duplicate types on the analysis results. In the domain of android malware [15], has addressed the sensitivity of the supervised and unsupervised learning models due to the existence of near duplicates. Thus, in this paper an initiative is taken to clarify the effect of both types of duplicates on classification and clustering learning models.

This paper investigates the impact of the deterministic and the probabilistic duplicates on the results of descriptive (clustering task), and predictive (classification task) data analytics. Five classification models namely, Decision Tree (DT), Support Vector Machine (SVM), Naïve Bayes (NB), Linear Discriminant Analysis (LDA) and Random Forest (RF) are used to clarify the effect of the deterministic duplicates with different ratios. The sensitivity of the five classifiers has been evaluated using three evaluation metrics, Accuracy[16], Matthews Correlation Coefficient (MCC)[17] and F1-Score[18]. While the impact of probabilistic duplicates is investigated through four clustering algorithms namely, The Partition Around Medoids (PAM), Clustering for Large Application (CLARA), K-means and Density-Based Spatial Clustering of Applications with Noise (DBSCAN). Furthermore, two evaluation indices namely the Average Silhouette Coefficient (ASC)[19] and DUNN Index (DI)[20]

are used to track the sensitivity of the four clustering algorithms.

Since that the simple metric doesn't consider the probability of the near duplicates, thus a metric is introduced to estimate the true percentage of near duplicates within a dataset.

The remainder of this paper is organized as follows, Section II, reviews the state of the art and presents the related studies. Where the used clustering evaluation indices are described in Section III. The experimental framework and results are discussed in Section IV. Section V concluded the paper.

II. Related Work

The field of duplication and its treatment methods is immense in the literature. It is addressed from different perspectives and domains, started from an overview of the duplication and techniques[21]–[23], improving the detection techniques[24], [25], evaluating the impact of the duplicates[26], and proposing new frameworks[27], [28] and methods[29], [30] to effectively enhance the detection process. However, little research addresses the impact of duplicates on the analysis results. Some studies are presented below.

In [15], they examined the effect of duplication on the supervised and unsupervised learning in the domain of android malware detection. From their perspective, duplication in this domain means that data samples (e.g., the APK, the DEX code, etc.) appear many times within a corpus. They declared that duplication has limited impact on the supervised classification model, however, it has a significant effect on the unsupervised clustering model.

In [31], two of the benchmark datasets are cleaned from near duplicate images. Deep leaning models are tested against the datasets before and after removing near duplicates. The results revealed a decrease in the classification accuracy by 9% to 14% which means that the duplicates' existence can give more inflated results.

Furthermore in[32], the author examined the impact of code duplication while evaluating machine learning models. He declared that code duplication has a negative effect on the machine learning models' performance which sometimes inflated by 100%. So, he recommended removing any exact or near duplicates to have more reliable and accurate results.

In the image processing field[33], the near images between training and test sets are removed to improve the quality of machine learning results. Four classifiers, RF, DT, SGD, and perceptron classifiers are used, and their performance is recorded. There is a slight decrease in the accuracy of four classifiers after removing near images which means that duplicates can give deceptive performance.

The authors[10], introduced a new technique to detect the near duplicates. Their technique doesn't depend on columns to detect duplicates, but on the metadata that describes the datasets. Their experiments declared around 95% accuracy rate.

While [34] propose a new record linkage deduplication framework through six steps that detects and visualizes duplicates in the datasets.

Within another study[35] the natural language preprocessing and machine learning are used to detect the duplicates with 90% for area under the curve.

The probabilistic duplicates detection approaches, such as similarity-based derivation, and decision-based derivation, are presented in[36]. To effectively detect duplicates they examine the adaptation of search space reduction, like using Blocking and Sorted Neighborhood methods which effectively reduces the record pair comparisons.

A new duplication detection framework is proposed[37]. It depends on metric functional dependencies (MFDs) to enhance the detection accuracy. Their experimental results on three real datasets show an improvement of 25% and 34% in precision and recall respectively.

Most of the studies have focused on introducing the duplicates detection techniques and reducing the search space of the records comparison. However, few studies investigated the effect of duplicates on the analytical results. Most of them tackled the problem of probabilistic (near) duplicates[38]–[43] due to its complexity than the exact ones[44].

III. Clustering Evaluation Indices

A. Average Silhouette Coefficient

Average Silhouette Coefficient is a measure of the separation distance between clusters, (1). It is a graphical display of how well each data point is clustered. The silhouette coefficient ranges from −1 to +1, higher values that are closer to 1, indicate more coherent clusters[45].

$$s = \frac{b-a}{\max\{a,b\}} \tag{1}$$

Where (a) is the mean of the intra-cluster distance, the average dissimilarity of data points in the same cluster. And (b) is the mean value of the nearest-cluster distance, the cluster with the smallest average dissimilarity.

B. DUNN Index

The Dunn Index quantifies the ratio between the smallest distance between cases in different clusters and the largest distance within a cluster, (2). A high DI means better clustering since observations in each cluster are closer together, while clusters themselves are further away from each other[46].

$$D = \min_{i=1,\dots,k}\left\{\min_{j=i+1,\dots,k}\left[\frac{d(C_i,C_j)}{\max_{l=1,\dots,k} diam(C_l)}\right]\right\} \tag{2}$$

Where k is the number of clusters, Ci is the ith cluster, d (Ci, Cj) is the distance between cluster Ci and Cj, and diam (Ci, Cj) is the diameter between the two clusters[46].

IV. Experimental Design

The main target is clarifying the effect of the presence and absence of duplication on data analytics. The two types of

duplication are investigated, the deterministic and the probabilistic duplication. The effect of the deterministic duplicates is examined through five different classifiers: DT and RF follow decision tree manner, SVM follows linear algorithms that separates between classes with a hyperplane, NB works using Probabilistic technique, and LDA works on discriminating between classes by maximizing distance and minimizing scale between them. The five classifiers are tested against a dataset with different ratios of deterministic duplicates (5%, 10% and 15%). For more accurate tracking of the changes in the performance of each classifier, three evaluation metrics are used, Accuracy, F1-Score and MCC.

Due to the lack of the labelled benchmark datasets within probabilistic duplicates, unsupervised learning is applicable in this case. The sensitivity of four clustering algorithms toward probabilistic duplicates is evaluated through different ratios of probabilistic duplicates (Zero, 3.06% and 5.54%). The PAM, CLARA and K-means are partitioning based clustering. Where DBSCAN is a density-based clustering. The performance of the four clustering algorithms is validated using Average Silhouette Coefficient and DUNN Index. The full description of the used datasets, experimental steps, and results are presented below.

A. Datasets

For a classification task, a synthetic dataset obtained from UCI Machine Learning Repository [47] is used. Whereas two benchmark datasets[48] are used in clustering task. Table I shows more description about these datasets.

TABLE I. DESCRIPTION OF THE DATASETS

Dataset Name	Size	Features	Data Type	Dataset Type
Dry Beans	13,611	17	Numeric	Synthetic
DBLP	2616	5	Mixed	Benchmark
ACM	2294	5	Mixed	Benchmark

B. Experimental Design

A total of 84 experiments were conducted to measure the effect of different types of duplicates on the results of descriptive and predictive data analytics. The detailed experimental steps are presented below.

1) Experimental steps for deterministic duplicates: The sensitivity of five classifiers toward the presence and absence of deterministic duplicates is investigated through 60 experiments, divided into four groups. Group 1, zero duplicates, where the classifiers are tested against the original dataset to report the baseline performance. Group 2, the 5% duplicates, where 5% of the deterministic duplicates are inserted into the original dataset then the sensitivity of the classifiers are measured using the three-evaluation metrics, Accuracy, F1-Score and MCC. Group 3, the 10% duplicates, and Group 4, the 15% duplicates, follow the same structure as Group 2 except that the percentage of the deterministic duplicates is changed before testing the sensitivity of the classifiers toward the duplicates. The 60 experiments, 15 for each Group (5 classifiers x 1 dataset x 3 evaluation metrics), are conducted to clearly report the changes of each classifier's performance. The upper part of Fig. 1 shows a general description of the four groups of experiments.

2) Experimental steps for probabilistic duplicates: In this section the sensitivity of four clustering algorithms toward the presence and absence of probabilistic duplicates is measured through 24 experiments. The experimental steps have been divided into 3 groups, 8 experiments in each group (4 clustering algorithms x 1 dataset x 2 evaluation indices). Group A, the DBLP dataset has been integrated with ACM dataset to be one dataset with 0% of probabilistic duplicates. The four clustering algorithms are tested against the integrated dataset and have been evaluated using the Average Silhouette Coefficient and DUNN Index. In Group B, the percentage of the duplicates increased to 3.06%, then the sensitivity of the four clustering algorithms was tested and evaluated. In Group C, the ratio of probabilistic duplicates increased to 5.54%. Then the performance of the four clustering algorithms is measured and reported using the evaluation indices. In the three groups of experiments, Group A, B, and C, the percentage of near duplicates is measured using our proposed metric. The lower part of Fig.1 represents a general description of the three groups of experiments.

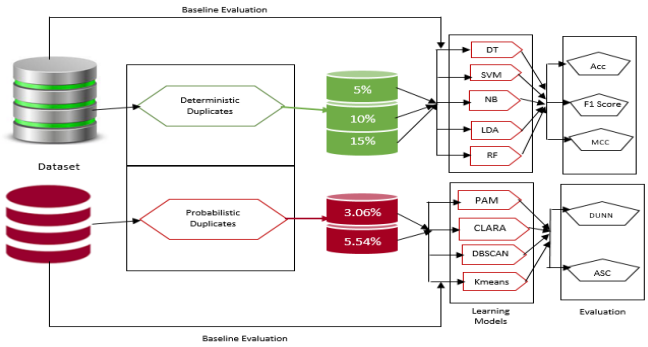


Fig. 1. Framework of all duplication experiments.

C. Experimental Results

The experimental results are addressed in two sections, the first section includes the results of the effect of deterministic duplicates on predictive analytics. Whereas the second section interprets the effect of probabilistic duplicates on descriptive analytics.

1) Results of deterministic duplicates effect: This section takes place into four subsections, the first subsection presents the baseline performance. The three other subsections show the experimental results of Group 2, 3 and 4 when having different ratios of deterministic duplicates.

a) The Zero Duplicates Experiment: The five classifiers are tested against the original dataset (zero duplicates) and their performance is reported and evaluated using three evaluation metrics as shown in Table II. The best classifier performance is recorded for RF followed by SVM, DT, LDA and NB respectively when evaluated by MCC, Accuracy and F1-Score measures.

b) The 5% Duplicates Experiment: In this group of experiments, 5% of deterministic duplicates are added to the original dataset. This duplicate percentage is measured using the simple equation, (3), that measures the ratio of duplicates.

$$Duplication = \frac{Total\ Number\ of\ Duplicate\ Rows}{Total\ Number\ of\ Rows} \quad (3)$$

The sensitivity of the five classifiers toward these duplicates is tested and recorded by the evaluation metrics. It is obviously clear that the performance of the five classifiers increased across all evaluation metrics compared with the baseline performance, Table III. 5% of duplicates positively affected the classifiers, which means that the five classifiers are positively sensitive to the existence of deterministic duplicates in this case. Higher performance means higher sensitivity. In this case, SVM reported higher sensitivity across all metrics, then comes RF, followed by DT and LDA. Whereas NB has the lowest sensitivity compared to other classifiers.

However, the performance of the five classifiers increased, this performance doesn't reflect the reality, true performance as in Table II, thus it can't be trusted or even used in making decisions.

c) The 10% Duplicates Experiment: Following the same steps in Group 2 experiment, the original dataset is contaminated by 10% of deterministic duplicates calculated by the same Eq., (3). As shown in Table IV, the performance of the five classifiers is still increasing after adding the 10% of duplicates compared with their performances in the previous two Groups of experiments, Table (II and III). The results declared that the five classifiers are still positively sensitive to the presence of deterministic duplicates. Furthermore, the performance of the five classifiers is still unreliable and can't be trusted. SVM and RF reported higher positive sensitivity in this case. NB has the lowest positive sensitivity, where DT and LDA have middle sensitivity toward 10% of exact duplicates.

d) The 15% Duplicates Experiment: The sensitivity of the five classifiers is measured and evaluated after adding 15% of deterministic duplicates into the original dataset. Table V shows that the performance of all classifiers except RF drastically decreased under the baseline performance. This indicates that DT, SVM, NB, and LDA are negatively sensitive to the existence of deterministic duplicates with large percentages more than 10%. However, the performance of RF is still increasing through the three-evaluation metrics (MCC, F1-Score, and Accuracy) compared with its performance across all previous experiments. In the case of negative sensitivity, NB has the highest sensitivity, then DT and SVM. Whereas LDA has the lowest negative sensitivity in this case.

Following are some observations after executing the four Groups of experiments:

- The performance of the RF classifier is positively affected by the presence of deterministic duplicates under small and large ratios, Fig. 2(a).
- As shown in Fig. 2(b), (c), (d), and (e), the DT, SVM, LDA and NB are positively sensitive to deterministic duplicates with certain percentages of duplicates ranging from 5% to 10%.
- However, they are negatively sensitive when having deterministic duplicates of more than 10%, Fig. 2(b), (c), (d), and (e).
- In general, the presence of deterministic duplicates limited to 10% has a positive effect on a classification task. As this percentage increases the effect of the deterministic duplicates differs based on the classifier used.
- Neither the positive nor the negative classifiers' sensitivity are good results. The classifiers in both cases are giving deceptive performance that doesn't reflect the reality. Hence, any decision taken based on a dataset contaminated by deterministic duplicates with any ratio is a completely wrong decision which can have negative implications on any business.

TABLE II. THE BASELINE PERFORMANCE

Dataset	Metrics	DT	SVM	NB	LDA	RF
Dry Beans	MCC	89.51%	91.00%	87.45%	87.86%	90.72%
	F1-Score	91.30%	92.50%	89.50%	89.70%	92.30%
	Acc.	91.31%	92.53%	89.54%	89.72%	92.31%

TABLE III. THE SENSITIVITY OF THE CLASSIFIERS WITH 5% OF DETERMINISTIC DUPLICATES

Dataset	Metrics	DT	SVM	NB	LDA	RF
Dry Beans	MCC	90.10%	91.78%	88.10%	88.57%	91.41%
	F1-Score	91.70%	93.20%	90.00%	90.40%	92.90%
	Acc.	91.72%	93.17%	90.04%	90.37%	92.87%

TABLE IV. THE SENSITIVITY OF THE CLASSIFIERS WITH 10% OF DETERMINISTIC DUPLICATES

Dataset	Metrics	DT	SVM	NB	LDA	RF
Dry Beans	MCC	90.42%	92.11%	88.51%	89.02%	91.77%
	F1-Score	92.00%	93.40%	90.40%	90.70%	93.10%
	Acc.	92.02%	93.44%	90.39%	90.73%	93.15%

TABLE V. THE SENSITIVITY OF THE CLASSIFIERS WITH 15% OF DETERMINISTIC DUPLICATES

Dataset	Metrics	DT	SVM	NB	LDA	RF
Dry Beans	MCC	88.70%	90.20%	86.40%	87.40%	92.13%
	F1-Score	90.50%	91.80%	88.60%	89.20%	93.40%
	Acc.	90.53%	91.78%	88.57%	89.24%	93.43%

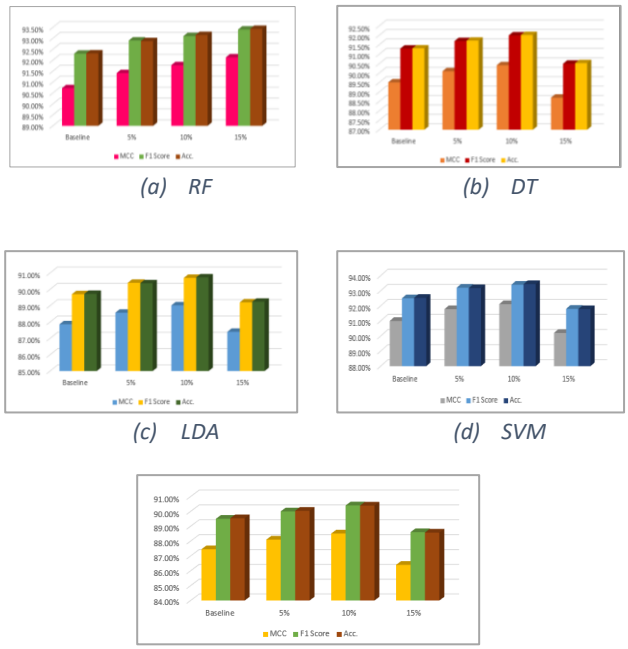


Fig. 2. Performance of the five classifiers with different ratios of deterministic duplicates.

2) Results of probabilistic duplicates effect: The experimental results on the effect of probabilistic duplicates are presented into three subsections. In the three subsections, the performance of the clustering algorithms with different ratios of probabilistic duplicates ranged from 0% to 5.54% is reported and evaluated using the Average Silhouette Coefficient and DUNN Index, both distance functions, Manhattan, and Euclidean functions, are used to calculate DUNN Index.

a) Group A Experiments: In this experimental group, the ACM and DBLP datasets are merged to have one cleaned integrated dataset. Then the four clustering algorithms are applied, and their quality is measured. Table VI shows the baseline performance of all clustering algorithms with zero duplicates. Generally, the performance of all clustering models is low, this is due to the dimensionality reduction problem. As stated in the literature [49]–[51], reducing the dimensions of the dataset improves the clustering results, however this is not the scope of this paper but showing the effect of near duplicates on the performance of the clustering models within different ratios.

TABLE VI. THE SENSITIVITY OF THE CLUSTERING ALGORITHM WITHOUT PROBABILISTIC DUPLICATES

Evaluation Indices	PAM	CLARA	Kmeans	DBSCAN
DUNN- Manhattan	0.018	0.017	0.016	0.096
DUNN- Euclidean	0.029	0.026	0.021	0.149
AVG.silhouette Coefficient	0.2	0.26	0.159	0.076

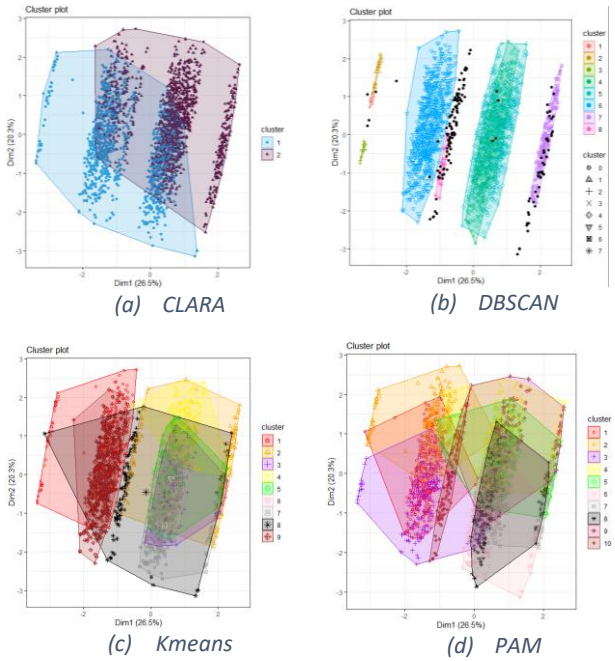


Fig. 3. Performance of the four clustering algorithms with zero probabilistic duplicates.

The evaluation done by DUNN index using the two distance functions shows that DBSCAN has the higher performance, followed by PAM then, CLARA whereas Kmeans reported the lowest performance. However, this is not the case when evaluating the quality of the four algorithms with Average Silhouette Coefficient. CLARA reported highest value, followed by PAM then Kmeans whereas DBSCAN has the lowest performance in this case. The baseline performance of the four clustering algorithms is depicted in Fig. 3.

The clusters created using DBSACN and CLARA are clearly depicted, Fig. 3(a) and (b). Whereas the clusters created by PAM and Kmeans, as shown in Fig. 3(c) and (d), overlapped, and cannot be detected clearly.

b) Group B Experiments: In this group of experiments, the integrated dataset is contaminated by 3.06% of probabilistic duplicates. The following example is given to explain how this percentage is calculated.

A dataset is given with the following:

Total Number of Records = 2,667 Unique Records = 2,388

Duplicate Records = 279 Probabilistic Match

The Data Uniqueness of the given dataset using the old metric is:

$$Uniqueness = \frac{\text{Total Unique Rows}}{Total\ Number\ of\ Records} = 2388/2667$$

$$= 0.895$$

So, the data duplication will be:

$$Duplication = 1 - Uniqueness$$
$$= 1 - \frac{Total\ Unique\ Rows}{Total\ Rows}$$
$$= \frac{Total\ Number\ of\ Duplicate\ Rows}{Total\ Number\ of\ Rows}$$

= 1-0.895 = 0.105=10.5%

However, the total number of duplicate records in the above equation refers to the deterministic duplicates and it does not consider the probability or the similarity by which the record is considered duplicates. To consider the probabilistic duplicate records, a new metric is proposed.

$$Data\ Duplication = \frac{1}{N}\sum_{i=1}^{N} P_i \quad (4)$$

Where N is the total number of records, i is the record number, and Pi is the probability that I record has a duplicate. So, if the i record is unique then Pi = 0, if it has a deterministic duplicate, Pi = 1, and if it is probably duplicate, then Pi is the similarity measure between this record and its linkage.

Data Duplication = (0.2* 162 + 0.4*105+ 0.6*12) / 2667 = 81.6/2667 = 0.0306.

This means that the uniqueness of the integrated dataset = (1-0.0306) = 0.969.

Thus the 10.5% of duplicates measured using old metric doesn't reflect the similarity of the probabilistic records, whereas the proposed metric considers the probability of probabilistic duplicates while measuring, thus the true percentage of near duplicates in this case is 3.06%.

Fig. 4. shows different effects of near duplicates on clustering algorithms as the ratio of duplicates increased. Generally, DBSCAN is a robust algorithm as its performance doesn't change from its baseline. Whereas Kmeans is positively affected by such increased ratio. PAM and CLARA give different performances depending on the evaluation indices used. A detailed description of the sensitivity for each clustering algorithm is presented in Table VII.

The following observations can be derived from this group of experiment:

- DBSCAN has robust performance when evaluated by DUNN index. While it is negatively affected by the increase of the duplicates when Average Silhouette Coefficient used, which means that its performance degraded from its baseline performance. It decreased by 0.022%.
- The performance of Kmeans is increased by around 0.006% when evaluated by DUNN index. Kmeans has a positive sensitivity in this case. But this is not the case while using Average Silhouette Coefficient, Kmeans is negatively affected by near duplicates, its performance decreased by 0.012%.
- PAM shows different sensitivity, robust and negatively affected by near duplicates, within DUNN index using both distances Manhattan and Euclidean respectively. Also, Average Silhouette Coefficient shows negative sensitivity for PAM.
- CLARA has different sensitivity, positive, robust, and negative sensitivity, when evaluated by Manhattan, Euclidean, and Average Silhouette Coefficient respectively.

c) Group C Experiments: The ratio of probabilistic duplicates increased to 5.54% within the integrated dataset using the new metric, (4) within the given dataset:

Total Number of Records = 2,947 Unique Records = 2,389

Duplicate Records = 558 Probabilistic Match

Thus, Data Duplication = (0.2* 324 + 0.4*210+ 0.6*24)/2,947 = 163.2/2,947 = 0.0554

The four clustering algorithms are tested against the integrated dataset and their sensitivity is assessed using the evaluation indices. In Fig. 5, it is obviously clear that the performance of the four clustering algorithms except Kmeans is low. This means that the clustering algorithm, CLARA, PAM, and DBSCAN are negatively affected by the new ratio of probabilistic duplicates, Fig. 5(a), (b), and (d). However, Kmeans is positively affected by such increase in the duplicate's ratio, Fig. 5(c).

TABLE VII. THE SENSITIVITY OF THE CLUSTERING ALGORITHM WITH 3.06% OF PROBABILISTIC DUPLICATES

Evaluation Indices	PAM	CLARA	Kmeans	DBSCAN
DUNN- Manhattan	0.018	0.022	0.021	0.096
DUNN- Euclidean	0.023	0.026	0.028	0.149
AVG.silhouette Coefficient	0.19	0.2	0.147	0.054

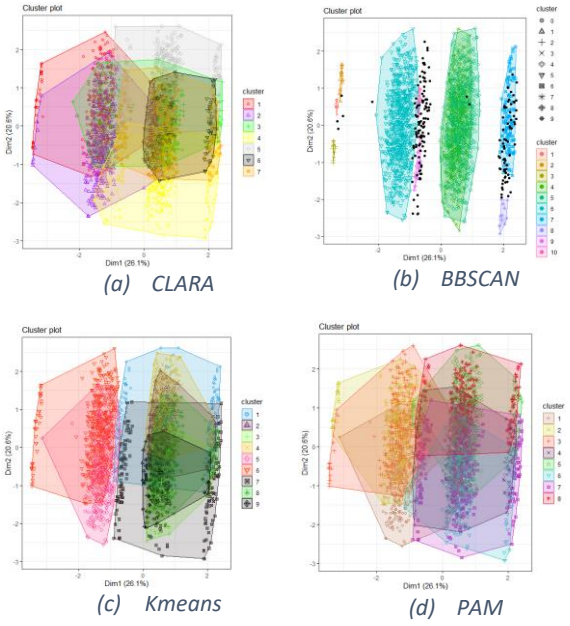


Fig. 4. Performance of the four clustering algorithms with 3.06% of probabilistic duplicates.

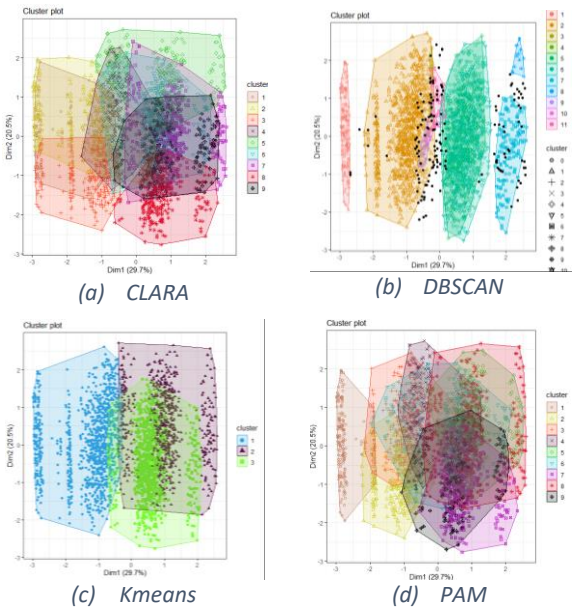


Fig. 5. Performance of the four clustering algorithms with 5.54% of probabilistic duplicates.

TABLE VIII. THE SENSITIVITY OF THE CLUSTERING ALGORITHM WITH 5.54% OF PROBABILISTIC DUPLICATES

Evaluation Indices	PAM	CLARA	Kmeans	DBSCAN
DUNN- Manhattan	0.011	0.012	0.023	0.045
DUNN- Euclidean	0.018	0.017	0.035	0.073
AVG. silhouette Coefficient	0.2	0.21	0.196	0.008

Based on the quantitative results presented in Table VIII, we can conclude that DBSCAN is highly sensitive toward a high ratio of duplicates, PAM came in the second rank of sensitivity, whereas CLARA has the lowest sensitivity compared with other algorithms. In this case the performance of Kmeans increases as ratio of probabilistic duplicates increased too. Thus, Kmeans is the only algorithm that has a positive sensitivity toward near duplicates.

Fig. 6 and Fig. 7 show the sensitivity observation of the four clustering algorithms after conducting all groups of Experiments, Group A (zero duplicates), Group B (3.06% duplicates), and Group C (5.54% duplicates).

1) Sensitivity measured by DUNN index

a) PAM Algorithm has a negative sensitivity toward the existence of probabilistic duplicates which means that its performance decreases as the percentage of the duplicates increases Fig. 6(a).

b) Fig. 6(b) shows that CLARA has a negative sensitivity only in the case of a high percentage of probabilistic duplicates, 5.54%. The performance of CLARA remains the same or slightly increases (by 0.005%) when the percentage of duplicates increased from zero to 3.06%.

c) DBSCAN has the same performance as CLARA, its performance is negatively affected by high ratios of the probabilistic duplicates, Fig. 6(c). Its performance remains the same when having zero and 3.06% of these duplicates.

d) On the other flip of the coin, comes Kmeans. Kmeans has a positive sensitivity to the existence of probabilistic duplicates with different ratios. Fig. 6(d) shows that the performance of Kmeans increases as the ratio of probabilistic duplicates increases too.

2) Sensitivity measured by average silhouette coefficient

a) As depicted in Fig. 7(a), the performance of PAM algorithm decreased from its baseline when 3.06% of duplicates were inserted into the dataset. Then it started to increase again as a larger ratio of probabilistic duplicates inserted. This indicates that PAM has no clear performance.

b) CLARA is a negatively sensitive clustering algorithm to the presence of near duplicates, Fig. 7(b). Its performance decreases when probabilistic duplicates exist. However, the performance of CLARA during different ratios of duplicates (3.06% and 5.54%) is almost the same, only 0.01% difference.

c) In Fig. 7(c), the performance of DBSCAN algorithm is negatively affected by the existence of probabilistic duplicates. The performance of DBSCAN clearly decreased when the ratio of the duplicates increased. Thus, we can conclude that DBSCAN is the highly sensitive algorithm toward probabilistic duplicates.

d) Fig. 7(d) declared that generally Kmeans is considered as a positively sensitive algorithm. Compared with the baseline, its performance increased (by 0.037%) as high ratios of probabilistic duplicates increased too.

Neither the positive nor the negative sensitivity of the clustering models toward probabilistic duplicates is good performance. The same as concluded from deterministic experiments, any decision taken in these cases will cause severe harm to any business either financially wise or management wise.

The experimental results agreed with what mentioned in different studies [15], [31]–[33], that the existence of both types of duplicates is somehow has an effect either positive or negative on the performance of the learning models.

Based on the quantitative results for each experiment, the existence of duplicates has a significant effect on the performance of supervised learning models, whereas it has a moderate impact on the performance of unsupervised learning models, which is the opposite of what mentioned in [15]. Fig. 8 shows the sensitivity of the learning models toward duplicates.

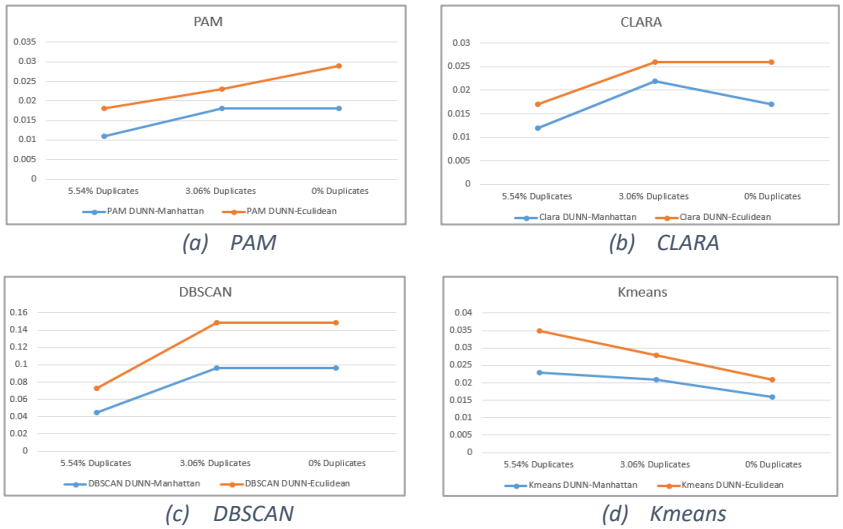


Fig. 6. The sensitivity of the four clustering algorithms using DUNN index.

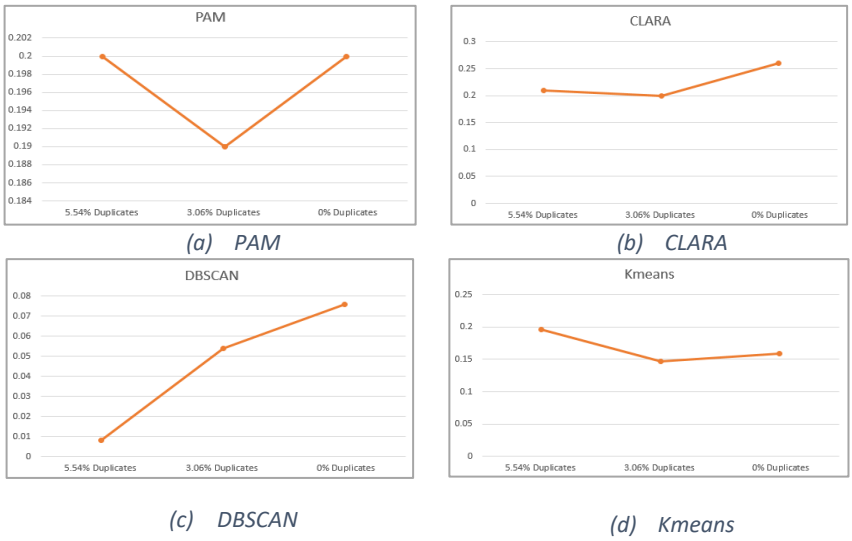
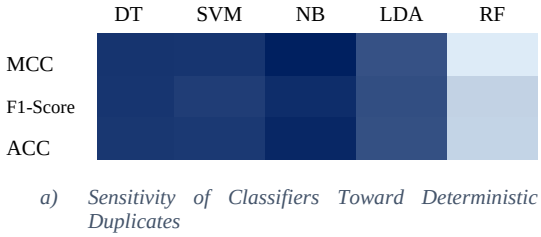
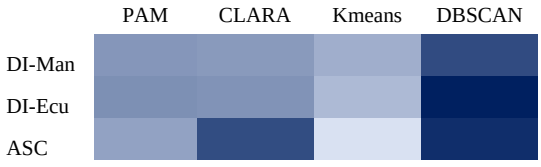


Fig. 7. The sensitivity of the four clustering algorithms using average silhouette coefficient.



a) Sensitivity of Classifiers Toward Deterministic Duplicates



b) Sensitivity of Clustering Models Toward Probabilistic Duplicates

Fig. 8. The sensitivity of all learning models toward different types of duplicates.

In Fig. 8. the Dunn index using Manhattan function is referred to as (DI-Man), whereas Dunn index using Euclidean function as (DI-Ecu). The Average silhouette Coefficient is referred to as (ASC). Furthermore, the accuracy is shortened in (ACC).

The difference in the average performance from the baseline performance is calculated and depicted in Fig. 8. It is clearly noted that DT, SVM, NB, and LDA are highly sensitive toward the existence of deterministic duplicates. Whereas RF is positively sensitive in this case. Thus, generally deterministic duplicates have a severe influence on most of the classifiers.

However, most of the clustering models are moderately sensitive toward the existence of probabilistic duplicates. Only DBSCAN is extremely sensitive toward near duplicates. Thus, the existence of probabilistic duplicate has a moderate influence on the unsupervised learning models.

The existence of both types of duplicates can cause significant harm to the analysis results and then the whole business. Thus, it is highly recommended to remove any duplicates from the dataset before putting it through the processing phase.

V. CONCLUSION

Cleaning the data from quality issues like missing values, inconsistencies, duplication, etc. is an essential step if accurate decision is needed. Thus, in this paper the sensitivity of supervised (DT, SVM, RF, NB, and LDA) and unsupervised (DBSCAN, Kmeans, CLARA, and PAM) learning models toward the existence of two types of duplicates, probabilistic and deterministic, with different ratios (0%-15%) is investigated. The results of these models are validated using five evaluation metrics, MCC, F1-Score, Accuracy, Average Silhouette Coefficient, and DUNN Index. Three datasets are used through 84 experiments. The experimental results can be concluded as follows. First, both types of duplicates have an influence on the sensitivity of the learning models, which differs based on the learning model itself. Second, small percentages of deterministic duplicates have a positive impact on the five classifiers. This declares that the performance of five classifiers increased when exact duplicates exist with small ratios. Third, RF is the only algorithm that has a positive sensitivity toward exact duplicates with ratios more than 10%, whereas negative sensitivity is reported for the rest of classifiers under high ratio of exact duplicates. Fourth, Kmeans clustering algorithm has a positive sensitivity when having near duplicates with either small or large ratios. Fifth, generally the rest of clustering algorithms are negatively sensitive toward near duplicates especially with high percentages. Sixth, the proposed duplicate metric proved its effectiveness in measuring the true percentage of near duplicates within a dataset.

REFERENCES

[1] N. Micic, D. Neagu, F. Campean, and E. H. Zadeh, “Towards a Data Quality Framework for Heterogeneous Data,” in Proceedings - IEEE International Conference on Internet of Things, IEEE Green Computing and Communications, IEEE Cyber, Physical and Social Computing, IEEE Smart Data, iThings-GreenCom-CPSCom-SmartData, 2018, vol. Janua, pp. 155–162.

[2] J. Alipour and M. Ahmadi, “Dimensions and assessment methods of data quality in health information systems,” Acta Medica Mediterr., vol. 33, no. March, pp. 313–320, 2017.

[3] Y. Kodra, M. Posada, D. Paz, A. Coi, M. Santoro, F. Bianchi, F. Ahmed, Y. R. Rubinstein, and J. Weinbach, “Data Quality in Rare Diseases Registries,” in Rare Diseases Epidemiology: Update and Overview, Springer Cham, 2017, pp. 149–164.

[4] T. Romanian and E. Journal, “Data Quality Dimensions to Ensure Optimal Data Quality,” Rom. Econ. J., no. 63, pp. 89–103, 2017.

[5] M. Rehman, “Role of FCBF Feature Selection in Educational Data Mining,” Mehran Univ. Res. J. Eng. Technol., vol. 39, no. 4, pp. 772–778, 2020.

[6] M. I. Gabr, Y. M. Helmy, and D. S. Elzanfaly, “Data Quality Dimensions, Metrics, and Improvement Techniques,” Futur. Comput. Informatics J., vol. 6, no. 1, pp. 25–44, 2021.

[7] T. Avoundjian, J. C. Dombrowski, M. R. Golden, J. P. Hughes, B. L. Guthrie, J. Baseman, and M. Sadinle, “Comparing Methods for Record Linkage for Public Health Action: Matching Algorithm Validation Study,” JMIR Public Heal. Surveill., vol. 6, no. 2, pp. 1–12, 2020.

[8] J. A. Oostema, A. Nickles, and M. J. Reeves, “A Comparison of Probabilistic and Deterministic Match Strategies for Linking Prehospital and in-Hospital Stroke Registry Data,” J. Stroke Cerebrovasc. Dis., vol. 29, no. 10, 2020.

[9] Q. Chen, J. Zobel, and K. Verspoor, “Benchmarks for measurement of duplicate detection methods in nucleotide databases,” Database, no. January, 2017.

[10] M. Chevallier, N. Rogovschi, and F. Boufar, “Detecting Near Duplicate Dataset with Machine Learning,” Int. J. Comput. Inf. Syst. Ind. Manag. Appl., vol. 14, no. July, pp. 1–12, 2022.

[11] F. Naumann and M. Herschel, “An Introduction to Duplicate Detection.” 2010, pp. 1–87.

[12] W. Xia, D. Feng, H. Jiang, Y. Zhang, V. Chang, and X. Zou, “Accelerating content-defined-chunking based data deduplication by exploiting parallelism,” Futur. Gener. Comput. Syst., vol. 98, pp. 406–418, 2019.

[13] A. Gkoulalas-Divanis, D. Vatsalan, D. Karapiperis, and M. Kantarcioglu, “Modern Privacy-Preserving Record Linkage Techniques: An Overview,” IEEE Trans. Inf. Forensics Secur., 2021.

[14] R. Vaziri, M. Mohsenzadeh, and J. Habibi, “TBDQ: A pragmatic task-based method to data quality assessment and improvement,” PLoS One, vol. 11, pp. 1–30, 2016.

[15] Y. Zhao, L. Li, H. Wang, H. Cai, T. F. Bissyandé, J. Klein, and J. Grundy, “On the Impact of Sample Duplication in Machine-Learning-Based Android Malware Detection,” ACM Trans. Softw. Eng. Methodol., vol. 30, no. 1, pp. 1–36, 2021.

[16] M. Bekkar, H. K. Djemaa, and T. A. Alitouche, “Evaluation Measures for Models Assessment over Imbalanced Data Sets,” J. Inf. Eng. Appl., vol. 3, no. 10, pp. 27–39, 2013.

[17] D. Chicco, “The Matthews Correlation Coefficient (MCC) is More Informative Than Cohen ’ s Kappa and Brier Score in Binary Classification Assessment,” IEEE Access, vol. 9, no. Mcc, pp. 78368–78381, 2021.

[18] A. Tharwat, “Classication Assessment Methods,” Appl. Comput. Informatics, 2020.

[19] H. Řezanková, “Different approaches to the silhouette coefficient calculation in cluster evaluation,” 21st Int. Sci. Conf. AMSE, no. August, pp. 1–10, 2018.

[20] N. Saini, S. Saha, and P. Bhattacharyya, “Automatic Scientific Document Clustering Using Self-organized Multi-objective Differential Evolution,” Cognit. Comput., vol. 11, pp. 271–293, 2019.

[21] G. Papadakis, D. Skoutas, E. Thanos, and T. Palpanas, “Blocking and Filtering Techniques for Entity Resolution: A Survey,” ACM Comput. Surv., vol. 53, no. 2, 2020.

[22] B. H. Li, Y. Liu, A. M. Zhang, W. H. Wang, and S. Wan, “A Survey on Blocking Technology of Entity Resolution,” J. Comput. Sci. Technol., vol. 35, no. 61772268, pp. 769–793, 2020.

[23] V. Christophides, V. Efthymiou, T. Palpanas, G. Papadakis, and K. Stefanidis, "End-to-End Entity Resolution for Big Data: A Survey," arXiv Prepr. arXiv1905.06397, vol. 1, no. 1, 2019.

[24] Z. A. Omar, M. A. A. Bakar, Z. H. Zamzuri, and N. M. Ariff, "Duplicate Detection Using Unsupervised Random Forests A Preliminary Analysis," 2022 3rd Int. Conf. Artif. Intell. Data Sci. (AiDAS). IEEE, 2022.

[25] A. Hindle and C. Onuczko, "Preventing duplicate bug reports by continuously querying bug reports," Empir. Softw. Eng., vol. 24, pp. 902–936, 2019.

[26] W. Elouataoui, I. El Alaoui, S. El Mendili, and Y. Gahi, "An End-to-End Big Data Deduplication Framework based on Online Continuous Learning," Int. J. Adv. Comput. Sci. Appl., vol. 13, no. September, pp. 281–291, 2022.

[27] Y. Aassem, I. Hafidi, and N. Aboutabit, "Enhanced Duplicate Count Strategy: Towards New Algorithms to Improve Duplicate Detection," ACM Int. Conf. Proceeding Ser., pp. 1–7, 2020.

[28] F. Panse and J. Schildgen, "Similarity-driven Schema Transformation for Test Data Generation," EDBT, pp. 408–413, 2022.

[29] K. Loannis, T. Papenbrock, and F. Naumann., "MDedup: Duplicate detection with matching dependencies," Proc. VLDB Endow. 13.5, pp. 712–725., 2020.

[30] S. Chavhan, P. Patil, and G. Patle, "Implementation of Improved Inline Deduplication Scheme for Distributed Cloud Storage," in 5th International Conference on Communication and Electronics Systems (ICCES). IEEE, 2020, no. Icces, pp. 1406–1410.

[31] B. Barz and J. Denzler, "Do We Train on Test Data? Purging CIFAR of Near-Duplicates," J. Imaging, vol. 6, no. 41, 2020.

[32] M. Allamanis, "The adverse effects of code duplication in machine learning models of code," Onward! 2019 - Proc. 2019 ACM SIGPLAN Int. Symp. New Ideas, New Paradig. Reflections Program. Software, co-located with SPLASH 2019, pp. 143–153, 2019.

[33] C. Geier and A. Science, "Training on test data: Removing near duplicates in Fashion-MNIST.," arXiv Prepr. arXiv1906.08255 (2019)., no. 2019, pp. 1–4.

[34] O. Azeroual, M. Jha, A. Nikiforova, K. Sha, M. Alsmirat, and S. Jha, "A Record Linkage-Based Data Deduplication Framework with DataCleaner Extension," Multimodal Technol. Interact., vol. 6, 2022.

[35] F. Halawa, M. Abdul, and R. Mohammed, "Applying Machine Learning for Duplicate Detection, Throttling and Prioritization of Equipment Commissioning Audits at Fulfillment Network," in IISE Annual Conference and Expo 2022, 2022, p. 2022.

[36] F. Panse, M. Van Keulen, A. De Keijzer, and N. Ritter, "Duplicate detection in probabilistic data," Proc. - Int. Conf. Data Eng., no. Section II, pp. 179–182, 2010.

[37] Y. Huang and F. Chiang, "Refining Duplicate Detection for Improved Data Quality," TDDL/MDQual/Futurity@ TPDL, p. { 1–10}, 2017.

[38] A. Biloshchytskyi, A. Kuchansky, S. Biloshchytska, and A. Dubnytska, "Conceptual model of automatic system of near duplicates detection in electronic documents," 2017 14th Int. Conf. Exp. Des. Appl. CAD Syst. Microelectron. CADSM 2017 - Proc., vol. 4, no. 1, pp. 381–384, 2017.

[39] D. V Luciv, D. V Koznov, G. A. Chernishev, and A. N. Terekhov, "Detecting Near Duplicates in Software Documentation 1," Program. Comput. Softw., vol. 44, no. 5, pp. 335–343, 2018.

[40] Y. He and J. Gao, "Detecting Short Near-Duplicates with Semantic Relations," Proc. IEEE Int. Conf. Softw. Eng. Serv. Sci. ICSESS, vol. 2018-Novem, pp. 122–125, 2019.

[41] M. Fröbe, J. Bevendorff, J. H. Reimer, M. Potthast, and M. Hagen, "Sampling Bias Due to Near-Duplicates in Learning to Rank," in Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020., 2020, pp. 1997–2000.

[42] H. Matatov, M. Naaman, and O. Amir, "Dataset and Case Studies for Visual Near-Duplicates Detection in the Context of Social Media," arXiv Prepr. arXiv2203.07167 (2022)., 2022.

[43] K. K. Thyagharajan and G. Kalaiarasi, "A Review on Near-Duplicate Detection of Images using Computer Vision Techniques," Arch. Comput. Methods Eng., vol. 28, no. 0123456789, pp. 897–916, 2021.

[44] E. Provencal and L. Laperrière, "Detection of exact and near duplicates in phased-array ultrasound weld scan," Procedia Manuf., vol. 54, no. 2019, pp. 263–268, 2020.

[45] H. B. Tambunan, D. H. Barus, J. Hartono, A. S. Alam, D. A. Nugraha, and H. H. H. Usman, "Electrical peak load clustering analysis using K-means algorithm and silhouette coefficient," Proceeding - 2nd Int. Conf. Technol. Policy Electr. Power Energy, ICT-PEP 2020, pp. 258–262, 2020.

[46] P. Rathore, Z. Ghafoori, J. C. Bezdek, M. Palaniswami, and C. Leckie, "Approximating Dunn's cluster validity indices for partitions of big data," IEEE Trans. Cybern., vol. 49, pp. 1629–1641, 2019.

[47] Dua and D. and Graff, "UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.," 2019.

[48] "https://dbs.uni-leipzig.de/research/projects/object_matching/benchmark_datasets_for_entity_resolution," Last Visited 29/10/2022.

[49] M. Alkhayrat, M. Aljnidi, and K. Aljoumaa, "A comparative dimensionality reduction study in telecom customer segmentation using deep learning and PCA," J. Big Data, vol. 7, 2020.

[50] S. Ayesha, M. K. Hanif, and R. Talib, "Overview and comparative study of dimensionality reduction techniques for high dimensional data," Inf. Fusion, vol. 59, no. May 2019, pp. 44–58, 2020.

[51] S. Sun, J. Zhu, Y. Ma, and X. Zhou, "Accuracy, robustness and scalability of dimensionality reduction methods for single-cell RNA-seq analysis," Genome Biol., vol. 20, pp. 1–21, 2019.