An End-to-End Big Data Deduplication Framework based on Online Continuous Learning

Widad Elouataoui[1]
Laboratory of Engineering Sciences
National School of Applied Sciences, Ibn Tofail University
Kenitra, Morocco

Imane El Alaoui[2]
Telecommunications Systems and Decision Engineering Laboratory, Ibn Tofail University
Kenitra, Morocco

Saida El Mendili[3]
Laboratory of Engineering Sciences
National School of Applied Sciences, Ibn Tofail University
Kenitra, Morocco

Youssef Gahi[4]
Laboratory of Engineering Sciences
National School of Applied Sciences, Ibn Tofail University
Kenitra, Morocco

***Abstract*—While big data benefits are numerous, most of the collected data is of poor quality and, therefore, cannot be effectively used as it is. One pre-processing the leading big data quality challenges is data duplication. Indeed, the gathered big data are usually messy and may contain duplicated records. The process of detecting and eliminating duplicated records is known as Deduplication, or Entity Resolution or also Record Linkage. Data deduplication has been widely discussed in the literature, and multiple deduplication approaches were suggested. However, few efforts have been made to address deduplication issues in Big Data Context. Also, the existing big data deduplication approaches are not handling the case of the decreasing performance of the deduplication model during the serving. In addition, most current methods are limited to duplicate detection, which is part of the deduplication process. Therefore, we aim through this paper to propose an End-to-End Big Data Deduplication Framework based on a semi-supervised learning approach that outperforms the existing big data deduplication approaches with an F-score of 98,21%, a Precision of 98,24% and a Recall of 96,48%. Moreover, the suggested framework encompasses all data deduplication phases, including data pre-processing and preparation, automated data labeling, duplicate detection, data cleaning, and an auditing and monitoring phase. This last phase is based on an online continual learning strategy for big data deduplication that allows addressing the decreasing performance of the deduplication model during the serving. The obtained results have shown that the suggested continual learning strategy has increased the model accuracy by 1,16%. Furthermore, we apply the proposed framework to three different datasets and compare its performance against the existing deduplication models. Finally, the results are discussed, conclusions are made, and future work directions are highlighted.**

Keywords*—Big data deduplication; online continual learning; big data; entity resolution; record linkage; duplicates detection

I. Introduction

Nowadays, data quality is gaining wide attention from both academics and professionals. Indeed, data quality dramatically impacts the business as executives rely mainly on data to manage their business and make informed decisions [1]. Indeed, better data quality translates directly into better business value. Data quality could be defined in terms of different dimensions such as completeness, accuracy, timeliness and consistency [2] [3].

This article addresses one of data quality's main aspects: uniqueness. Uniqueness ensures that there is only one instance of each information in the dataset and thus points out that there should be no data duplicates [4]. Indeed, data duplication issues are not only related to storage. Duplicate data also lead to inaccurate analysis, which may cause significant problems and costly mistakes. There are many sources of data redundancy, including users providing erroneous information, typing errors, data integration, etc. With the emergence of Big Data, data duplication has become more common and challenging. This is related to big data Volume, Variety, Velocity, and other characteristics of big data known as Big Data V's [5] [6]. Thus, because of the particular characteristics of big data, new data deduplication issues were raised related to the huge data volume, variety of data sources, inconsistency of data types and schemas, and so on.

Therefore, duplicate detection approaches have been widely discussed in the literature under different names, such as entity resolution, deduplication, or record linkage. All these terms refer to the same meaning: identifying records referring to the same real-world entity. The deduplication process is usually followed by an entity consolidation or fusion process defining the unified representation of duplicated values that best represents the real-world entity. Even if data deduplication was widely discussed in the literature, more efforts are needed to address the challenges related to Big Data Deduplication. Indeed, most existing big data deduplication approaches focus only on data volume. Also, most existing methods are limited to the duplicate detection phase, which is only a part of the deduplication process. Moreover, the current deduplication approaches are not ensuring a maintained accuracy score during the serving, so the model's performance usually decreases over time [7].

Believing that Big Data Deduplication should be addressed more comprehensively, we aim through this paper to enhance big data quality measurement with three main contributions:

- We suggest an End-to-End Big Data Deduplication Framework encompassing five phases: data pre-processing, data labeling, duplicate detection, data cleaning, and finally, model monitoring using continual retraining.
- We address the issue of the decreasing performance of the deduplication model by setting an online learning strategy for big data deduplication to maintain a high accuracy level during the serving.
- We design a framework that outperforms the existing big data deduplication methods and provides the best results based on a Semi-Supervised learning approach.

The rest of this paper is organized as follows: Section 2 describes the research methodology followed for the literature review. Section 3 reviews the most recent and relevant studies that have tackled data deduplication. Section 4 highlights the importance of deduplication for big data. Section 5 presents our suggested end-to-end big data deduplication framework. Section 6 offers the implementation of the suggested framework and discusses the obtained results. Finally, we highlight the primary outcomes as well as some research outlooks.

II. Research Methodology

A systematic literature review was conducted to capture and synthesize the relevant and available studies addressing data quality measurement. This literature review was performed following the guidelines stated in [8], where the authors have proposed a review methodology that consists of planning the review by preparing a review proposal. A second step consists of searching and selecting studies. Finally, the main findings of the review are reported. The goal of this study was to choose two main kinds of contributions:

- Studies suggesting deduplication frameworks in a big and non-big data context.
- Studies addressing Uniqueness as a quality metric

For this, primary research was conducted first using generic keywords such as “Data Deduplication”, “Entity Resolution” and “Data Uniqueness”. Then, to capture studies about big data, specific keywords such as “Big data Deduplication”, “Big Data Entity Resolution”, and “Big Data Redundancy” were used. Then, abstracts were reviewed, and irrelevant papers were excluded. This primary search yielded 60 articles. The research was limited to recent articles published in journals and conference proceedings and was performed on: IEEE Xplore, Springer, Google Scholar, Science Direct, Research Gate, and ACM digital libraries. After a literature search, the next step consists of narrowing down the papers based on their relevancy, freshness, and availability. For this, a diagonal reading was performed on the selected papers filtered out based on multiple criteria: we included studies that were addressing data deduplication, recent, available, in English, and published in digital libraries. A total of 23 articles were selected, followed by a more in-depth analysis.

Further, we reviewed the references of the selected studies and added two more articles to the selected papers. Then, the chosen studies were thoroughly read and carefully examined, and 17 studies were deemed relevant to the scope of our research. Finally, the articles' descriptive details were checked and filed in a Zotero database. This literature review has shown a significant lack of deduplication frameworks that fit big data requirements, which motivates us to perform an in-depth analysis of the current state of the art to frame the need and make a significant contribution. The following section reviews the papers selected for our study and highlights the main findings of this literature review.

III. Related Work

Data deduplication is a trending topic that many researchers have long addressed in the literature. Thus, many approaches for data deduplication have been suggested in the literature, such as [9], where the authors have proposed a six-step deduplication framework that detects duplicated records using record linkage. The framework includes preparing data, matching attributes using sorted neighborhood, building a decision mode, and clustering. In this paper, the authors have raised the current issue of the lack of labeled data for big data deduplication, which hinders the evaluation of the model performance. This issue was also mentioned in [10], where the authors have raised the lack of labeled data for deduplication and suggest using active learning as an alternative. The authors have achieved the highest results with an F-score of 98,4% for structured datasets. However, for dirty datasets, a lower score of 52% was achieved. Deep learning was also used to address data duplication, such as in [11], where the authors define a binary classification approach for safety engineering based on fuzzy string-matching algorithms. The proposed approach is based on Convolutional Neural Networks (CNN) binary classifier and string similarity-based classifier. All the research mentioned above has significantly contributed to duplicate detection and entity resolution. However, these approaches are not appropriate for large-scale datasets in terms of accuracy and execution time. Also, in addition to the dataset volume concern, the above approaches did not address the particular issues raised in a big data context. Indeed, with the emergence of big data, new challenges have been raised, such as the diversity of data sources, the variety of data types, and the high velocity and veracity of data [12] [13]. These particular issues were discussed in recent studies, such as in [14], where the authors have performed a survey of the indexing techniques for big data deduplication. The experiments have shown that sorted neighborhood is the best indexing technique for large datasets in terms of complexity. Also, in [7], Christophides et al. have performed a comprehensive survey of all the existing methods for entity resolution. They provided an overview of the different steps of entity resolution for big data, including blocking, block processing, matching, and clustering. The authors have also raised the challenge of the decreasing performance of the deduplication methods over time. Likewise, in [15], the authors have discussed big data's challenges to entity resolution and proposed a hybrid similarity measurement approach based on traditional syntactic and word-embedding approaches. In [16], Abd El-Ghafar et al. have suggested an entity resolution approach for big data based on hashing TF and Jaccard similarity. The approach was applied to seven scenarios where different Natural Language Processing (NLP) techniques were used to show the impact of these techniques on entity resolution. This approach reaches an accuracy of 91% for a dataset of 1M records. To address data duplication in the web of data, Efthymiou et al. have proposed in [17] a deduplication that allows reducing the required number of pairwise comparisons. The suggested process blocks data when comparing entity descriptions within the same blocks. The results show a high performance of the suggested method; however, it is only appropriate for the web of data as it is based on entity descriptions. Using deep learning, the authors in [18] have introduced a new Stacked Dedupe Learning entity resolution approach based on Bidirectional Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM). However, the study does not show the impact of big data on the performance and accuracy of the model. Moreover, recent studies have addressed the data blocking for big data, such as in [19], where the authors have defined a progressive blocking (PB), detecting 93%. of duplicates during the first third of the execution time. Likewise, a multi-phase blocking strategy detecting big data duplicates has been suggested in [20].

Even if data deduplication has been widely discussed in the literature, there have been few efforts to address deduplication issues in the big data context. In addition, most of the existing approaches are only limited to the duplicate detection phase and are not comprehensively managing data deduplication. Furthermore, the model's predictive performance usually degrades over time as the data keep changing. To the best of our knowledge, no deduplication approaches have been set to address the decreasing accuracy during the model serving. Therefore, we aim through this paper to propose an end-to-end Big Data Deduplication framework with three main contributions:

- Ensuring increased performance and accuracy of the deduplication by setting an online learning strategy for deduplication.
- Setting a more comprehensive Big Data deduplication Framework that addresses all the big data deduplication processes and consists of five steps: data preprocessing, data labeling, duplicate detection, data cleaning, and model monitoring using continual retraining.
- Suggesting a novel framework that outperforms the existing big data deduplication methods based on a Semi-Supervised learning approach.

In the next section, we highlight the importance of deduplication for big data.

IV. Big Data Deduplication

In a Big Data context, ensuring data quality has always been a critical concern for data managers. Data quality could be defined in terms of multiple dimensions, also called "Data Quality Dimensions" such as completeness, accuracy, readability, consistency, etc. In this paper, we are addressing one of the primary data quality dimensions: uniqueness. It refers to the unicity of the information provided by the dataset and ensures that there are no duplicated records. To improve data uniqueness, data should be cleaned from duplicated data. This process is known as Data Deduplication. Data duplication can occur for different reasons, such as data integration, where data are gathered from multiple data sources so the same information can be recorded more than once in another format [21] [22]. Also, data duplication could be related to human errors, so the same person, for example, could provide data with slightly different information intentionally or by mistake multiple times. Indeed, Experian [23] found that human input error is the leading cause of data inaccuracy and duplication. Data duplication heavily impacts data analysis and can negatively affect the business. Data duplication can bias data analytics. For example, companies lack a single customer view with duplicated customer dataset. They could not have a clear idea about the real number of their customers and their behavior which may hinder activities like targeted marketing. Also, data duplication incurs a high cost as it leads to wasteful marketing activities, such as targeting the same customer multiple times. Data duplication could also be costly in terms of storage, as redundant records can take up a lot of space, which increases storage costs. A recent study [24], about the impact of data duplication has shown that companies that store big data and apply a backup policy can see that 80% of their corporate data are duplicated. Also, according to another study [25], reducing the transmitted data can save money in terms of storage costs and backup speed up to 50%. Thus, data deduplication helps optimize marketing spending in terms of time and cost. In short, data duplication can result in significant damage and cost for businesses and, therefore, should be addressed effectively for accurate and successful data management. In the next section, we present the suggested end-to-end big data deduplication framework and describe each step of the framework straightforwardly.

V. A Smart End-to-End Big Data Deduplication Framework

In this section, we present an end-to-end Big Data deduplication Framework, shown in Fig. 1 to 6 that consists of five steps: The first step is a preprocessing phase where data is cleaned and prepared for deduplication due to the low quality of the extracted data in big data environments. The next step consists of building a training dataset using an automated data labeling process. Then, fuzzy matching is performed on the dataset to detect duplicates. The detected duplicates are then cleaned using the appropriate strategies. Finally, the model is deployed using a real-time continual learning strategy for continuous accuracy improvement during the serving. The framework is designed to address the different issues linked to big data environments. In the following, we provide a detailed description of each stage of the framework.

A. *Pre-processing*

Because of the Big Data V's, the extracted data in big data environments are usually unstructured, noisy, and poorly formatted. Therefore, going through a pre-processing phase is highly required before using data [26]. In this first phase, raw data is prepared and converted into a more appropriate format making it understandable and suitable for use by Machine Learning (ML) algorithms. This process significantly impacts the efficiency and accuracy of the model and can ruin the subsequent phases if it is not done correctly. In the following, we present the transformations required to prepare big data for deduplication, as shown in Fig. 2.

1) Feature Selection and Extraction: Feature Selection and Extraction are crucial in dealing with a high-dimensional dataset as not all the extracted data in Big Data environments are relevant for the intended use. The goal is to keep relevant information by selecting only the most informative variables (Feature Selection) or creating new useful ones (Feature Extraction). This process is required for data deduplication as it allows determining the most significant features on which the model will be based to detect duplicates.

2) Imputing: Big data is usually messy, skewing data analysis and leading to biased results. Imputing data is required for deduplication, especially when there is a large number of missing values, as, with low information, duplicates cannot be detected effectively. There are various ways to address the missing values depending on the ratio of the missing values. The missing values can either be ignored, deleted, or replaced by an estimate based on the existing part of the data. The estimated value could be the mean value, the most frequent value, the min or max value, etc. Data could also be attributed using ML algorithms such as K-Nearest Neighbour and Multivariate Imputation or deep learning such as DataWig.

3) Encoding: Encoding is the process of converting categorical variables into numeric types. Most ML algorithms cannot handle absolute values and work better with numerical inputs. There are multiple techniques for encoding, such as Label Encoding, One Hot Encoder, Vector Indexer, etc. Moreover, encoding ensures data consistency, a crucial factor for data deduplication. Indeed, as big data are gathered from multiple sources, categorical values may be represented differently, such inconsistency issues impede duplicate detection.

4) Uppercasing/Lowercasing: This transformation consists of standardizing text data to all Lowercase or Uppercase. For the sake of simplicity, it is more common to convert all data to lowercase, especially for NLP applications. This process is also essential for deduplication as the same word (Good/ good) may be taken as different words (in the vector space model) if we ignore this transformation.

5) Stop Words and Symbols Removal: This process consists of removing irrelevant words- the most common words- from the text data. The idea behind eliminating stop words is to provide more importance to the information contained within data, as ignoring them doesn't drastically impact the meaning. Also, the dataset should be cleaned from special symbols and punctuations as they will not help identify similarities. According to the context, other text elements could be removed, such as URLs, HTML tags, etc.

6) Normalization: Due to the variety of data sources, some variables in the data may have different scales. This inconsistency at the scale level will bias duplicate detection as records should be compared based on a unified scale. To overcome this, data should be normalized so that the range of all the variables is similar (usually between 0 and 1).

We consider these transformations the most important ones to prepare data for deduplication. However, according to the dataset context, more text cleaning may be needed, such as Spell Corrections and Stemming.

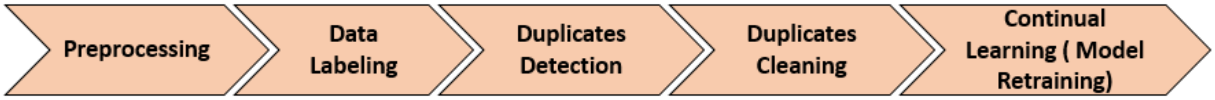

Fig. 1. End-to-End Data Deduplication Approach.

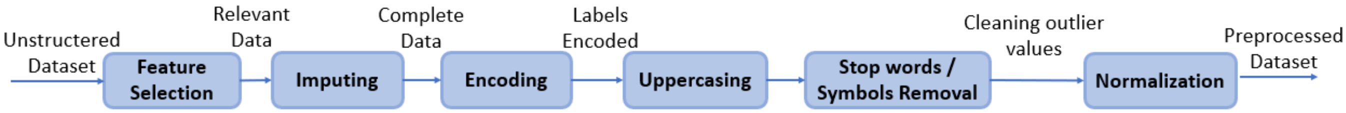

Fig. 2. Preprocessing Steps.

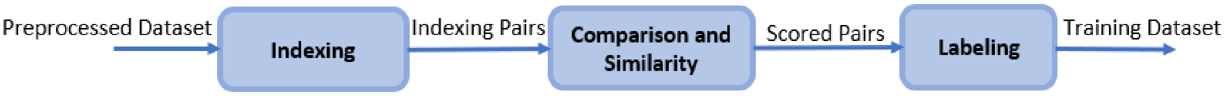

Fig. 3. Labeling Steps.

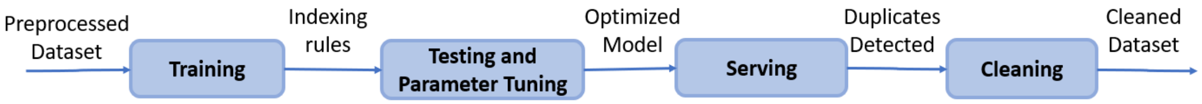

Fig. 4. Deduplication and Cleaning Steps.

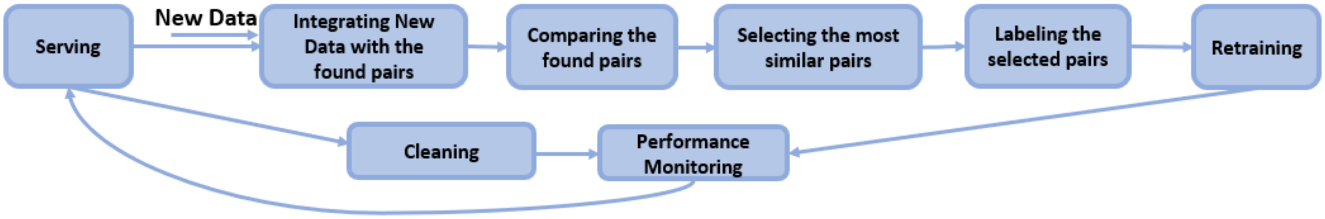
Serving
New Data
Integrating New Data with the found pairs
Comparing the found pairs
Selecting the most similar pairs
Labeling the selected pairs
Retraining
Cleaning
Performance Monitoring

Fig. 5. Online Continual Learning Process.

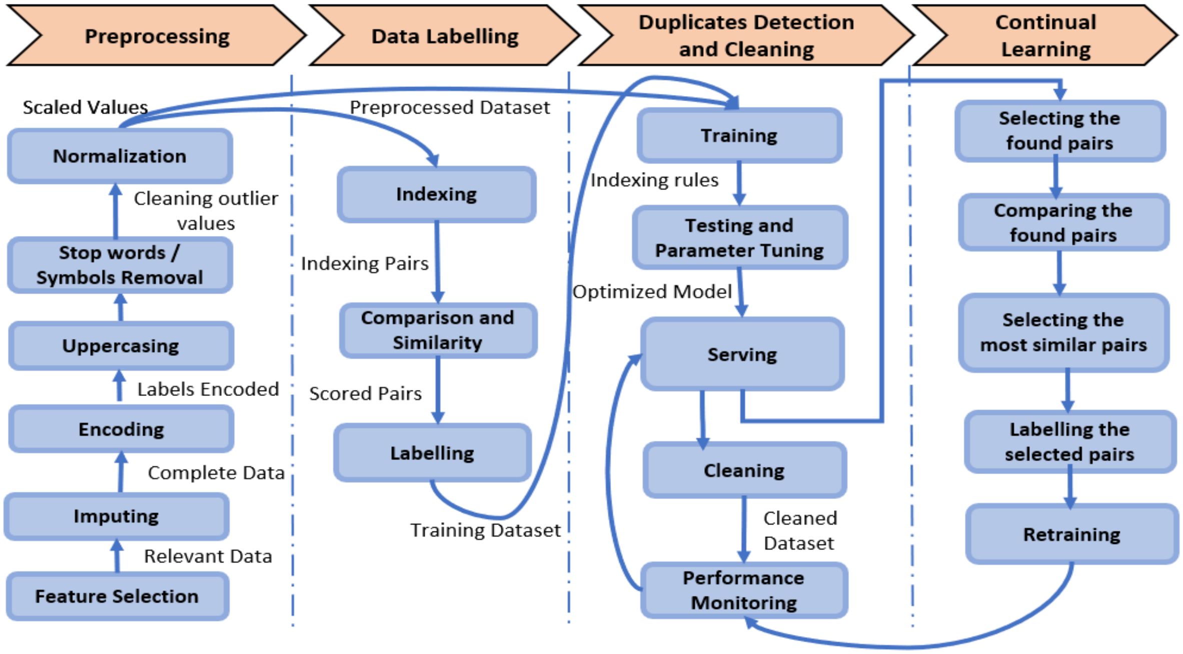

Fig. 6. Data Deduplication Pipeline.

B. Data Labeling

Once the dataset is pre-processed, and in a ready-to-use state, the next step consists of building a labeled dataset that will be used to train the deduplication model. Labeling data is one of the most challenging tasks that could be faced in AI projects. According to a study [27], labeling data takes up to 80% of AI project time. If most labeling approaches use human labelers, this solution becomes unsuitable when dealing with big data, not only for quantity reasons but also for quality reasons. To overcome this, we are using an automated approach to produce labeled data for deduplication based on Record Linkage techniques. This approach, shown in Fig. 3, consists first of indexing records into pairs. Then, a weighted similarity score is computed to determine if the couples are duplicates, and finally, pairs are labeled based on their similarity score.

1) Indexing: Indexing consists of generating pairs of candidate records. The idea behind this step is not to create all possible combinations of record pairs in the data set, as it will lead to quadratic time complexity, but to select only the likely duplicated pairs. Several indexing techniques are available for record linkage, such as Blocking, Sorted Neighbourhood, TF-IDF, etc. In this paper, we use Sorted Neighbourhood for pairs indexing as it is the most suitable indexing technique for big data [14]. More details are provided in the implementation section.

2) Comparison and Similarity: After generating the record pairs, a comparison of the candidate records is performed, and a similarity score is then attributed to each pair. Depending on the field type (string, numerical value, date…), multiple comparison measures, such as Jarowinkler, Levenshtein, Cosine, Jaccard, etc., could be used. For more accurate measurements, weights could also be assigned to data fields, as some areas may be more significant than others to determine duplicate records.

3) Labeling: Once similarity scores are measured, pairs are classified using supervised or unsupervised methods such as Optimal Threshold, SVM, K-Means, Farthest First, etc. Pairs are then classified into two classes (matches/non-matches). As a record can have more than one duplicate, we are suggesting in this approach to gather duplicates into clusters instead of pairs so that each cluster can contain more than two records. Non duplicates records are removed, and only matching records are kept as a training dataset.

C. Duplicates Detection

With a set of labeled data, we can thus train the deduplication model. Then, use the trained model to identify matches and find the correct parameters to get optimal results. It is worth noting that in this approach duplicate detection is not based on a text comparison but on deduplication predicates and indexing rules generated by the model after the training. More details about the generated indexing rules are provided in the implementation section.

1) Training: This first step consists of training the model to classify records as duplicates and non-duplicates based on the training dataset. At the end of the training, the model comes up with indexing rules that will be used to identify potential matches. Thus, records will be blocked by matching the deduced indexing rules (also called Predicates) during the learning.

2) Testing and Parameter Tuning: The next step consists of assessing the model's accuracy and maximizing its performance by finding the best-suited clustering threshold for the model to give optimal results. The parameter tuning can be done either manually or automatically using methods such as Bayesian Optimization, Random Search, and Tree-structured Parzen Estimator (TPE). Also, the parameter tuning remains relative to how precise we want to be on finding or dropping matches while clustering, as there is always this trade-off between precision and recall.

3) Serving: This last step uses the trained and optimized model to identify matches and classify the records as “duplicate” or “not duplicate”. Finally, the model returns clusters of partners. As duplication is transitive, clustering is performed on the matching pairs, so the same cluster's records are considered duplicates.

D. Duplicates Cleaning

Once matches are gathered into clusters, data should be consolidated from many records into one. Many data fusion strategies could be used at this stage according to the strategic priorities of the data team (see Fig. 4). For example, if the process is more oriented towards data accuracy, the record of the most reliable source will likely be kept. Otherwise, the complete record will be held if the goal is to gather as much data as possible. Another data fusion strategy is to create a new record by merging the existing ones. In this case, a conflict resolution approach should be implemented to integrate duplicated columns. Multiple data fusion strategies were discussed in [28] [29].

E. Continual Learning (Model Retraining)

Because of big data variability, data keeps changing constantly. Data could be changed regarding schema, statistical distribution, data quality, etc. This kind of change is known as data drift. In addition, data could also be exposed to a concept drift when the statistical properties of the target variable change over time [30]. Thus, the model's predictive performance may degrade over time because of data drift and concept drift. Therefore, it is crucial to adapt the model to data changes to ensure that the model accuracy is always maintained. For this, the model should be retrained after deployment according to an ML strategy called Continual Learning. Continual Learning is a process that automatically and continuously retrains a ML model with new data, which makes the model auto-adaptative and improves its performance. A critical use case of continual learning is recommendation systems that should always be updated with new data as user behavior changes over time. There are two approaches to performing continual learning:

- Offline Mode (Batch learning): In this approach, the model is retrained from time to time with the new accumulated data.
- Online Mode (Incremental Learning): the model is retrained sequentially with a live data stream.

With Online Continual Learning, the model does not decay following a data or concept drift as it is dynamically updated with new data patterns. The online mode is also a time effective solution as there is no need to store and manage large batches of accumulated data. On the other hand, the input data should constantly be monitored if the model is fed with insufficient data, the performance will be impacted instantly. The online mode remains suitable, especially in big data environments and real-time applications. Research has recently been conducted on Online Continual Learning, especially in a deep learning community. In [31], the authors have shown that algorithms and the architecture of neural networks impact continual learning performance. In [32], the authors have suggested a supervised training method for continual learning. The method's effectiveness was proven in three systems for continual online learning. In [33], the authors have introduced a new memory population approach (CBRS) for continual online learning that deals with imbalanced and temporally correlated data. Other pertinent methods for enhancing Online Continual Learning were suggested in [34] [35] [36]. For data deduplication, even if the deduplication model is trained with high-quality pairs, features defining duplications may change over time, especially when data is human input. Thus, new duplication features may come into play. Also, the used parts may become misleading, so they must be excluded or reweighted. Deduplication models are susceptible to duplication features, so a small features drift may drastically impact the model performance. In this regard, we suggest an Online Continual Learning approach for deduplication that consists of the following steps:

1) Building a dataset composed of new data and the found pairs during the serving.

2) Comparing and computing a similarity score of the built dataset.

3) Selecting the most similar pairs using a Threshold

4) Labeling the selected pairs as duplicates

5) Retraining the model with the new labeled pairs

6) Evaluating the model performance

This approach (Fig. 5) is executed in online mode, which makes it memory and time efficient, and hence, suitable for large datasets. This approach has also shown remarkable results in improving the model's accuracy. The model is continuously trained with new pairs, which allows updating the indexing rules with more pertinent ones. More details about the obtained results are provided in the next section.

In this section, we have presented the different steps of an end-to-end deduplication framework, including the data pre-processing, the labeling, the training, the serving phase, and finally, the retraining phase according to an online continual learning approach. For each phase, we have presented the different implementing techniques that could be used. Thus, the suggested framework is comprehensive and may be implemented differently depending on the intended use. Fig. 6 shows the machine learning pipeline of the whole framework. In the next section, we present how each step of the framework is implemented and the dataset and tools used for the implementation. Also, the suggested framework is compared against the existing approaches in terms of accuracy and scalability as the framework is designed to work in big data environments. Finally, a discussion is conducted about the possible evolutions.

VI. Implementation

A. Datasets Description

This section presents the implementation of the deduplication framework described in the previous section. The suggested framework was applied to 3 datasets:

Dataset 1: This first dataset is a built dataset with synthetic duplicated records. Indeed, to assess and evaluate the performance of the proposed strategy, the framework should be used for an extensive dataset with labeled duplicated records. Thus, we conducted research for datasets with two main criteria:

- A labeled dataset with a pre-defined state of true and false duplicates.
- Large Scale dataset with over 1M records.

Unfortunately, among the found datasets, no dataset matches the above criteria and, thus, was not appropriate for our use case. Indeed, previous research has also faced the same challenge as labeling big data sets manually are a very tedious and effortful task. To overcome this challenge, we built a labeled dataset with synthetic duplicated records using the Duplicate Generator tool DupGen [20] which allows for generating a synthetic dataset according to multiple criteria, such as the percentage of generated records and the changes made to data values. The built dataset contains over 1M records. It matches the Big Data characteristics not only in terms of Volume but also in terms of Variety, as the dataset was gathered from multiple restaurant data sources with different formats and schemas. To ensure consistency, we have only kept standard fields: name, address, city, and type that refer to the restaurant's specialty. To stress our deduplication Framework, distinguishing features such as phone number and email were not considered even if they were available in all the datasets. The data sources used were clean of duplicates and were chosen from different countries so to avoid having common records between the datasets. After integrating and pre-processing source datasets, we have gathered a dataset with over 500 000 unique restaurants. The next step consists of creating duplicated records. For an accurate assessment, this process should not be done randomly. For this, we have reviewed restaurant datasets with real duplicates (these datasets were not suitable for our use due to their small volume) and tried to simulate duplicated data using the DupGen tool. Thus, we have noticed that most duplicated restaurants have either:

- Identical name, similar address and similar city and type
- Identical address, similar name, and similar city and type
- Similar name, similar address, and similar city and type

Also, we measured the average number of different characters between two duplicates for each column and applied the same distribution to our built dataset. Finally, we have created a dataset with over 1 M records with the following duplicates distribution: 80%: no duplication, 10%:1 duplication, 4%: 2 duplications, 2%: 3 duplications, 2%:4 duplications, 1%:5 duplications, and 1%: 6 to 10 duplications.

The number of duplicates was around 122 000, so the goal was to reduce over 1M records to about 878 000.

Dataset 2: The second dataset is a real companies name dataset containing 663000 records with 58700 duplicated records [37]. The dataset is prelabelled and intended for deduplication frameworks. This dataset was chosen to test our framework performance with a dataset of real-world values.

Dataset 3: The third dataset is a small dataset of 864 records with 112 duplicated records [38]. This dataset is prelabelled and was used by previous research to evaluate the deduplication methods. This dataset was chosen to compare our framework performance against the existing models.

Table I presents the characteristics of the three datasets used for our experiments. Before submitting the simulation results, we will first review the implementing tools and techniques in the next section.

TABLE I. DATASETS CHARACTERISTICS

Dataset	Records	Matchings	Threshold
Restaurant	864	112	0.76
Company	663000	58700	0.83
Built Dataset	1001300	122 000	0.75

B. Adopted Tools and Techniques

The deduplication framework was developed on Apache Spark, suitable for Big Data. It was implemented in Python using Pyspark libraries such as Scikit-Learn for NLP and Fuzzy matching, Pandas, Scipy, and Numpy. For data pre-processing, string functions were used as well as some Python's preprocessing packages such as NLTk, Stopwords, Unicode, Geocoder, LabelEncoder, RE (regular expressions) and Text blob. Then, data were first indexed using **Sorted neighborhood to build the training dataset.** The sorted neighborhood is an indexing technique that consists of sorting data values using the blocking key value and then moving a window of a fixed number of records over the sorted values. The sorted neighborhood index method is great when there is a relatively large amount of unstructured data. A recent study [14] has compared the indexing techniques for scalable linkage. It has shown that a sorted neighborhood is the most appropriate indexing method for big data in terms of execution time and accuracy. For more precision, we have applied weighted indexing to the restaurant dataset using the weights presented in Table II. With this parametrization, we suppose that the name and address are the most important columns to consider for restaurant's deduplication. These weights have allowed detecting accurately most of the pairs. No weights were applied to the company datasets with only one column (Company Name). Then, a similarity score is measured using **Cosine Similarity**. As mentioned before, various methods, such as Euclidean distance, and the Jaccard coefficient, can be used. However, Cosine Similarity is the most suited to measure text similarity, according to several studies [39] [40]. For more accuracy, the pairs are filtered out based on a min and max threshold range to keep only the most similar records. The selected records are then gathered into clusters to be used as a training dataset for the deduplication process. As mentioned before, the next step consists of dedupling the dataset using a ML algorithm. As mentioned before, in this approach duplicate detection is not based on a text comparison but on deduplication predicates and indexing rules generated by the model after the training. For this, we have used **Dedupe**. Dedupe is a Python library for accurate and scalable data deduplication and fuzzy matching based on ML [41] [42]. The first step consists of creating a dedupe instance for the dataset. Then, the dedupe instance is trained using the dataset built in the previous step. After the training phases, the model generates the indexing rules that will be used to detect similar records. In our case, one of the generated predicates was: (CommonTwoTokens, name), (SameSevenCharStart, name), (CommonThreeTokens, address). This means that the records with Names with the same two tokens AND Addresses with the same three tokens are considered duplicates. Once trained, the model can be used for deduplication using a semi-supervised clustering method, so similar records are clustered based on the provided labeled dataset. Then, the clustering threshold is tuned to get an optimized accuracy. Finally, duplicates are cleaned.

The Continual learning is performed in an online mode as it is carried out sequentially after each deduplication and is executed in real-time according to an automated machine learning pipeline. For this, a new dataset is built based on the found pairs and additional 100 000 records. Then, the similarity of the detected pairs is evaluated using Cosine Similarity. A similarity threshold is set to select only the most matching pairs which are then labeled appropriately. The deduper is then retrained with the selected pairs to enhance the model's performance. An accuracy assessment is performed to evaluate the impact of continual learning on the model.

TABLE II. RESTAURANT DATASET INDEXING WEIGHTS

Name	Address	City	Type
0.55	0.35	0.05	0.05

C. Results

a) Accuracy:

The framework was first applied to an extensive dataset of over 1M records with 122 000 duplicated records. 70% of the dataset was set to build the training dataset. Thus, the dataset was first indexed into pairs. Only 420 000 records were indexed as pairs. Then a similarity score was computed for the indexed pairs and a threshold of 0.75 was set to filter out only the most similar pairs. This first process resulted in 165000 pairs considered duplicates. The selected pairs were then gathered into clusters to detect records with more than one duplicate and were exported to a .csv file as a training dataset. Then, for each dataset, a dedupe model was trained using the built training dataset, tested, and optimized using the appropriate threshold. The performance of the framework was evaluated using the confusion matrix defined by the following metrics:

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$F - score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (3)$$

TP, FP, and FN are True Positive, False Positive, and False Negative, respectively.

The metrics above were measured for three different datasets: Restaurants, Companies, and our Built Big Dataset coming out with the results presented in Table III.

TABLE III. DEDUPLICATION FRAMEWORK EVALUATION

Dataset	Precision (%)	Recall (%)	F-s (%)
Restaurant	98,25%	100,00%	99,12%
Company	94,17%	98,13%	96,11%
Built Dataset	98,24%	96,48%	98,21%

It is worth noting that the framework accuracy has evolved considerably after applying online continual learning. For our built dataset, the framework detected 117700 out of 122 000 duplicated records with an F-score of 98,21%. Indeed, the resulting F-score was initially 97,05% and has increased by 1,16% after applying the continual learning process to the model with an additional dataset of 100 000 records.

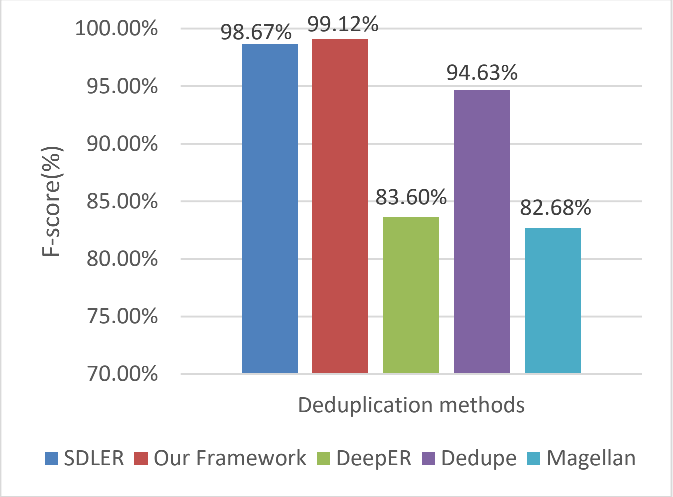

Fig. 7. F-Score Comparison.

As mentioned previously, we chose the third dataset to compare our framework's performance against the existing models that have used the same dataset (restaurant dataset), such as SDLER [18], DeepER [43], Magellan [44], and Dedupe [42]. Fig. 7 compares the F-score achieved by each model when applied to the Restaurant dataset. The obtained results show that the proposed framework provides the best results in terms of accuracy. When dealing with big data, the execution time is yet another factor that should be considered besides accuracy. We present in the next part the time complexity of the proposed framework.

b) Scalability

As the framework is intended to be used in a big data environment, the framework scalability also needs to be ensured. Table IV shows the processing time and the corresponding dataset size. Thus, the framework has shown acceptable results in terms of processing time with a linear complexity O(n). Indeed, the framework is based on scalable methods such as Sorted Neighborhood, Cosine Similarity, and Dedupe having a linear complexity and hence, are suitable for Big Data [45] [41].

TABLE IV. PROCESSING TIME

Dataset	Records	Processing Time
Restaurant	864	~3 m
Company	663000	~ 1h
Built Dataset	1001300	~ 3h30

c) Framework Limitations

A second phase of the implementation consists of scrambling the built dataset intentionally by feeding the datasets with more challenging duplicates. The goal is to uncover the framework limitations and discover how the accuracy is impacted by the inferior and very poor data quality and to what extent the framework remains suitable for use. For this, we have unfiltered in the dataset extreme cases of non-duplicates where for example the name and the address are similar, but the records are not duplicates. The framework was applied to a very poor big data quality to uncover the limitations of the framework. The dataset was then scrambled progressively with a very poor-quality dataset, and the accuracy was assessed in each round. The F-score has decreased in each round, as shown in Table V. Thus, it turns out that the framework resists and remains functional in a half-scrambled dataset with an F-score of 88,2%. Therefore, the accuracy is acceptably impacted by a very-poor biasing dataset.

TABLE V. F-SCORE EVOLUTION IN A VERY POOR-QUALITY DATASET

Percentage of scrambled data	F-s (%)
20 % of the very poor-quality dataset	97,9%
35 % of the very poor-quality dataset	94,8%
50 % of the very poor-quality dataset	88,2%
60 % of the very poor-quality dataset	83,4%

D. Discussion

Although significant efforts have been made in recent years for data deduplication, there are still challenges to be addressed, especially for big data. Indeed, data uniqueness as a quality metric depends highly on other quality metrics such as completeness, accuracy, validity, etc. For example, even if we have imputed data during the pre-processing phase, most imputation methods are not accurate, which can impact the deduplication accuracy as data is credited with inaccurate values. Meanwhile, ignoring missing values will negatively affect the model accuracy, especially in a big data environment where most of the data are incomplete. On the other hand, deduplication can also impact the other metrics, as the cleaning phase consists of keeping the most accurate, complete, or recent record. In some cases, records can even be merged. All these changes have a high impact on the other metrics. Thus, data deduplication could not be improved separately and, therefore, should be addressed in a more comprehensive approach that considers this strong relationship between the quality metrics. Continual Learning is yet another research area that needs more focus. Even if Continual Learning has been around for more than 20 years, there are challenges that still need to be addressed, such as catastrophic forgetting, auditing, mentoring, evaluating continual learning techniques, etc. In addition to these challenges, new issues have been raised with big data, such as handling memories, learning for streaming multimodal data, model saturation, etc. Thus, continual learning is not already in its explosion, and further research is needed. However, it is safe to say that Continual Learning will become increasingly crucial as ML models could not be effectively performed without accumulating the learned knowledge.

VII. CONCLUSION

While data deduplication has been the subject of several studies in the last decade, some challenges remain, especially in the Big Data Era. In this article, we have reviewed the most recent big data deduplication frameworks suggested in the literature. We also proposed a novel end-to-end big data deduplication framework based on a Semi-supervised clustering approach. The experiments have shown that the framework outperforms the existing big data deduplication approaches with an F-score of 98,21%. The suggested framework is also extended with an online continual learning phase that continuously improves the deduplication model performance and increases the model accuracy by 1,16%. In future work, we aim to enhance our framework by reducing the error rate when used on a very-poor quality dataset. Also, we aim to extend our framework to address more quality dimensions.

References

[1] Y. Gahi, M. Guennoun, and H. T. Mouftah, ‘Big Data Analytics: Security and privacy challenges’, in 2016 IEEE Symposium on Computers and Communication (ISCC), Jun. 2016, pp. 952–957. doi: 10.1109/ISCC.2016.7543859.

[2] I. El Alaoui, Y. Gahi, and R. Messoussi, ‘Big Data Quality Metrics for Sentiment Analysis Approaches’, in Proceedings of the 2019 International Conference on Big Data Engineering, New York, NY, USA, Jun. 2019, pp. 36–43. doi: 10.1145/3341620.3341629.

[3] I. E. Alaoui and Y. Gahi, ‘The Impact of Big Data Quality on Sentiment Analysis Approaches’, Procedia Comput. Sci., vol. 160, pp. 803–810, Jan. 2019, doi: 10.1016/j.procs.2019.11.007.

[4] W. Elouataoui, I. E. Alaoui, and Y. Gahi, ‘Data Quality in the Era of Big Data: A Global Review’, in Big Data Intelligence for Smart Applications, Y. Baddi, Y. Gahi, Y. Maleh, M. Alazab, and L. Tawalbeh, Eds. Cham: Springer International Publishing, 2022, pp. 1–25. doi: 10.1007/978-3-030-87954-9_1.

[5] I. E. Alaoui, Y. Gahi, and R. Messoussi, ‘Full Consideration of Big Data Characteristics in Sentiment Analysis Context’, in 2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), Apr. 2019, pp. 126–130. doi: 10.1109/ICCCBDA.2019.8725728.

[6] Elouataoui, W.; El Alaoui, I. and Gahi, Y. (2022). Metadata Quality Dimensions for Big Data Use Cases. In Proceedings of the 2nd International Conference on Big Data, Modelling and Machine Learning - BML, ISBN 978-989-758-559-3, pages 488-495. DOI: 10.5220/0010737400003101

[7] V. Christophides, V. Efthymiou, T. Palpanas, G. Papadakis, and K. Stefanidis, ‘An Overview of End-to-End Entity Resolution for Big Data’, ACM Comput. Surv., vol. 53, no. 6, p. 127:1-127:42, Dec. 2020, doi: 10.1145/3418896.

[8] D. Tranfield, D. Denyer, and P. Smart, ‘Towards a Methodology for Developing Evidence-Informed Management Knowledge by Means of Systematic Review’, Br. J. Manag., vol. 14, no. 3, pp. 207–222, 2003, doi: 10.1111/1467-8551.00375.

[9] 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, no. 4, Art. no. 4, Apr. 2022, doi: 10.3390/mti6040027.

[10] Simonini, Giovanni; Saccani, Henrique; Gagliardelli, Luca; Zecchini, Luca; Benevento, Domenico; Bergamaschi, Sonia. The Case for Multi-task Active Learning Entity Resolution / (2021).

[11] M. Pikies and J. Ali, ‘Analysis and safety engineering of fuzzy string matching algorithms’, ISA Trans., vol. 113, pp. 1–8, Jul. 2021, doi: 10.1016/j.isatra.2020.10.014.

[12] Y. Gahi and I. El Alaoui, ‘Machine Learning and Deep Learning Models for Big Data Issues’, in Machine Intelligence and Big Data Analytics for Cybersecurity Applications, Y. Maleh, M. Shojafar, M. Alazab, and Y. Baddi, Eds. Cham: Springer International Publishing, 2021, pp. 29–49. doi: 10.1007/978-3-030-57024-8_2.

[13] I. El Alaoui, Y. Gahi, R. Messoussi, A. Todoskoff, and A. Kobi, ‘Big Data Analytics: A Comparison of Tools and Applications’, in Innovations in Smart Cities and Applications, Cham, 2018, pp. 587–601. doi: 10.1007/978-3-319-74500-8_54.

[14] S. YEDDULA and K. LAKSHMAIAH, ‘INVESTIGATION OF TECHNIQUES FOR EFFICIENT & ACCURATE INDEXING FOR SCALABLE RECORD LINKAGE & DEDUPLICATION’, Int. J. Comput. Commun. Technol., vol. 6, no. 1, Sep. 2020, doi: 10.47893/IJCCT.2015.1275.

[15] X. Chen, ‘Towards Efficient and Effective Entity Resolution for High-Volume and Variable Data’, p. 167, 2020, doi: 10.25673/35204

[16] El-Ghafar, R.M., El-Bastawissy, A.H., Nasr, E.S., Gheith, M.H., & Independent Researcher, C.E. (2021). An Effective Entity Resolution Approach for Big Data. International Journal of Innovative Technology and Exploring Engineering.

[17] V. Efthymiou, K. Stefanidis, and V. Christophides, ‘Big data entity resolution: From highly to somehow similar entity descriptions in the Web’, in 2015 IEEE International Conference on Big Data (Big Data), Oct. 2015, pp. 401–410. doi: 10.1109/BigData.2015.7363781.

[18] A. Ngueilbaye, H. Wang, D. A. Mahamat, and I. A. Elgendy, ‘SDLER: stacked dedupe learning for entity resolution in big data era’, J. Supercomput., vol. 77, no. 10, pp. 10959–10983, Oct. 2021, doi: 10.1007/s11227-021-03710-x.

[19] T. Papenbrock, A. Heise, and F. Naumann, ‘Progressive Duplicate Detection’, IEEE Trans. Knowl. Data Eng., vol. 27, no. 5, pp. 1316–1329, May 2015, doi: 10.1109/TKDE.2014.2359666.

[20] El-Ghafar, R. M. A., El-Bastawissy, A. H., Nasr, E. S., & Gheith, M. H. (2020). An Efficient Multi-Phase Blocking Strategy for Entity Resolution in Big Data. In International Journal of Innovative Technology and Exploring Engineering (Vol. 9, Issue 9, pp. 254–263).

[21] W. Elouataoui, I. El Alaoui, and Y. Gahi, ‘Metadata Quality in the Era of Big Data and Unstructured Content’, in Advances in Information, Communication and Cybersecurity, Cham, 2022, pp. 110–121. doi: 10.1007/978-3-030-91738-8_11.

[22] A. Z. Faroukhi, I. El Alaoui, Y. Gahi, and A. Amine, ‘Big data monetization throughout Big Data Value Chain: a comprehensive review’, J. Big Data, vol. 7, no. 1, p. 3, Jan. 2020, doi: 10.1186/s40537-019-0281-5.

[23] Experian.com. (n.d.). Retrieved August 12, 2022, from http://experian.com/assets/decision-analytics/white-papers/the%20state%20of%20data%20quality.pdf

[24] Chaitra. (2021, June 22). Understanding data deduplication - and why it's critical for moving data to the cloud. Druva. Retrieved August 12, 2022, from https://www.druva.com/blog/a-simple-definition-what-is-data-deduplication

[25] Druvainc. (n.d.). Customers win with Druva and AWS. Druva. Retrieved August 12, 2022, from https://content.druva.com/c/eb-customers-win-with-druva-aws?x=4if2hg

[26] A. Z. Faroukhi, I. El Alaoui, Y. Gahi, and A. Amine, ‘An Adaptable Big Data Value Chain Framework for End-to-End Big Data Monetization’, Big Data Cogn. Comput., vol. 4, no. 4, Art. no. 4, Dec. 2020, doi: 10.3390/bdcc4040034.

[27] Buschi, N. (2021, June 23). Top 5 challenges making data labeling ineffective. Dataloop. Retrieved June 4, 2022, from https://dataloop.ai/blog/data-labeling-challenges/.

[28] Christen, P.: Further topics and research directions. In: Christen, P. (ed.) Data Matching, pp. 209–228. Springer, Heidelberg (2012)

[29] D. Elkington, X. Zeng, and R. Morris, ‘Resolving and merging duplicate records using machine learning’, US20160357790A1, Dec. 08, 2016.

[30] Komolafe, A. (2022, July 22). Retraining model during deployment: Continuous training and continuous testing. neptune.ai. Retrieved August 25, 2022, from https://neptune.ai/blog/retraining-model-during-deployment-continuous-training-continuous-testing

[31] S. I. Mirzadeh et al., ‘Architecture Matters in Continual Learning’, arXiv, arXiv:2202.00275, Feb. 2022. doi: 10.48550/arXiv.2202.00275.

[32] J. Gallardo, T. L. Hayes, and C. Kanan, ‘Self-Supervised Training Enhances Online Continual Learning’, arXiv, arXiv:2103.14010, Oct. 2021. doi: 10.48550/arXiv.2103.14010.

[33] A. Chrysakis and M.-F. Moens, ‘Online Continual Learning from Imbalanced Data’, in Proceedings of the 37th International Conference on Machine Learning, Nov. 2020, pp. 1952–1961.

[34] E. Fini, S. Lathuilière, E. Sangineto, M. Nabi, and E. Ricci, ‘Online Continual Learning under Extreme Memory Constraints’, arXiv, arXiv:2008.01510, Jan. 2022. doi: 10.48550/arXiv.2008.01510.

[35] H. Koh, D. Kim, J.-W. Ha, and J. Choi, ‘Online Continual Learning on Class Incremental Blurry Task Configuration with Anytime Inference’, presented at the International Conference on Learning Representations, Sep. 2021.

[36] C. Wiwatcharakoses and D. Berrar, ‘A self-organizing incremental neural network for continual supervised learning’, Expert Syst. Appl., vol. 185, p. 115662, Dec. 2021, doi: 10.1016/j.eswa.2021.115662.

[37] Caesarlupum. (2019, November 19). Deduping &amp; Record Linkage. Kaggle. Retrieved August 12, 2022, from https://www.kaggle.com/code/caesarlupum/deduping-record-linkage

[38] Duplicate detection, record linkage, and identity uncertainty: Datasets. (n.d.). Retrieved August 12, 2022, from https://www.cs.utexas.edu/users/ml/riddle/data.html

[39] Cosine similarity. Cosine Similarity - an overview | ScienceDirect Topics. (n.d.). Retrieved August 25, 2022, from https://www.sciencedirect.com/topics/computer-science/cosine-similarity

[40] M. Kirişci, cosine similarity FFS. 2022.

[41] F. Gregg, dedupe: A python library for accurate and scaleable data deduplication and entity-resolution. Accessed: Jun. 18, 2022. Available: https://github.com/dedupeio/dedupe.

[42] Vintasoftware. Deduplication-slides/slides.ipynb at vintasoftware/deduplication-slides. GitHub. Retrieved August 12, 2022, from https://github.com/vintasoftware/deduplication-slides/blob/631389413a558ea83a407a47870253325b7b068e/slides.ipynb

[43] M. Ebraheem, S. Thirumuruganathan, S. Joty, M. Ouzzani, and N. Tang, ‘Distributed representations of tuples for entity resolution’, Proc. VLDB Endow., vol. 11, no. 11, pp. 1454–1467, Jul. 2018, doi: 10.14778/3236187.3236198.

[44] A. Doan et al., ‘Human-in-the-Loop Challenges for Entity Matching: A Midterm Report’, in Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics, New York, NY, USA, May 2017, pp. 1–6. doi: 10.1145/3077257.3077268.

[45] Project C CSE 494/598 Hemal Khatri - Arizona State University. (n.d.). Retrieved August 12, 2022, from https://rakaposhi.eas.asu.edu/cse494/f05-projects/ProjC_Hemal.pdf