Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 6, 846-863
2025
Publisher: Learning Gate
DOI: 10.55214/25768484.v9i6.7970

Optimizing data quality in big data through unsupervised record linkage techniques

Aissam Bendida[1*], Amar Bensaber Djamel[2], Réda Adjoudj[3], Yahia Atig[4]
[1]University of Sidi Bel Abbes (EEDIS Laboratory), Algeria, bendida@cuniv-naama.dz / aissam.bendida@gmail.com (A.B.)
[2]Ecole Superieure en Informatique (LabRI-SBA Laboratory), Algeria, d.amarbensaber@esi-sba.dz (A.B.D.)
[3]University of Sidi Bel Abbes (EEDIS Laboratory), Algeria, adjreda@yahoo.fr (R.A.)
[4]University University Centre of Naama (Department of Computer Science), Algeria, atig@cuniv-naama.dz (Y.A.).

Abstract: In today's era of Big Data, maintaining high-quality data is crucial for effective data management. One key aspect of this is record linkage, which involves identifying, comparing, and merging records from different sources that refer to the same real-world entity. However, traditional record linkage methods struggle to keep up with the rapidly increasing volume and diversity of data. These methods often rely on labeled data, which can be expensive and difficult to obtain. To overcome these challenges, unsupervised blocking techniques have emerged as a promising alternative, allowing large-scale datasets to be managed efficiently without the need for pre-labeled data. In this article, we introduce a novel approach that integrates the Firefly Algorithm for optimized feature selection, Locality-Sensitive Hashing (LSH) for dimensionality reduction, and Length-based Feature Weighting (LFW) for improved data representation. Our methodology aims to enhance both the accuracy and scalability of record linkage in Big Data environments. Experimental results show that our approach is highly effective, demonstrating its potential to significantly improve data quality in large-scale datasets.

Keywords: *Blocking, Locality-sensitive-hashing, Firefly algorithm, Length-based feature weighting, Record Linkage.*

1. Introduction

Effective data management largely depends on the quality of the Record Linkage (RL) process, which involves identifying, comparing, and merging records originating from multiple databases but representing the same real-world entity (see Figure 1). However, as data volume increases, the number of record pairs requiring comparison grows exponentially, making the linkage process increasingly difficult and computationally expensive. To address this issue, an initial step known as "blocking" is typically implemented prior to the linkage stage itself. This preliminary phase significantly reduces the number of comparisons by grouping similar records into separate blocks, thus simplifying the linkage task and making it practically feasible even for large-scale datasets [1].

History: Received: 31 March 2025; Revised: 20 May 2025; Accepted: 23 May 2025; Published: 11 June 2025
* Correspondence: bendida@cuniv-naama.dz

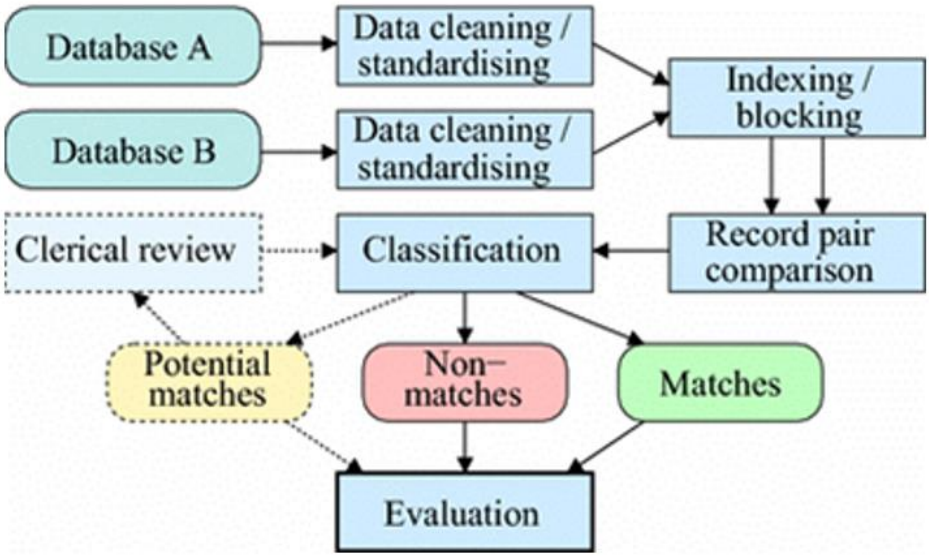

Figure 1.
A summary of the record linkage process [2].

He The evaluation of record linkage methods, including those based on unsupervised blocking, relies on several essential metrics. Among these, the Reduction Ratio (RR) measures the effectiveness in reducing the number of comparisons made, while Pairs Completeness (PC) assesses how thoroughly all relevant matches have been identified. Pairs Quality (PQ) evaluates the accuracy of the process by analyzing the proportion of correctly matched pairs. Finally, a common approach to achieve a balanced evaluation is to calculate the harmonic mean between RR and PC, thus providing an overall measure of the record linkage performance [3].

The performance of record linkage techniques, including those that utilize unsupervised blocking strategies, is commonly evaluated using several key metrics:

- Reduction Ratio (RR): This metric indicates the efficiency of the blocking process by measuring the reduction in the total number of comparisons. It is calculated using Equation (1):

$$RR = 1 - \frac{N}{|A|*|B|} \quad (1)$$

Where N is the number of candidate record pairs after blocking, and |A|×|B| represents the total possible pairs from datasets A and B.

- Pairs Completeness (PC): PC reflects the ability of the method to retrieve all true matches. It is expressed (2) as:

$$\boldsymbol{PC = \frac{N_m}{|M|}} \quad (2)$$

With N_m denoting the number of correctly identified true matches, and |M| the total number of actual matches in the data.

- Pairs Quality (PQ): This metric evaluates the precision of the blocking technique by determining the proportion of correct matches among all proposed pairs. As shown in Equation (3):

$$PQ = \frac{N_m}{N} \quad (3)$$

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 6: 846-863, 2025
DOI: 10.55214/25768484.v9i6.7970

Where N_m is the number of accurate matches and N_b is the total number of pairs generated post-blocking.

To balance completeness and efficiency, the harmonic mean of RR and PC is often used as a composite score. This is formally expressed in Equation (4):

$$F_{RR,PC} = \frac{2 .RR .PC}{RR+PC} \quad (4)$$

In this article, we propose an unsupervised blocking method that combines the Firefly Algorithm [4] for feature selection, Locality-Sensitive Hashing (LSH) [5] for dimensionality reduction, and Length-based Feature Weighting (LFW) [6] for improved data representation. Our approach is designed to optimize both the quality and efficiency of the record linkage process, particularly in Big Data environments where traditional methods face limitations.

The remainder of this article is structured as follows: Section 2 provides a review of related work, highlighting the limitations of existing methods and the need for a more effective approach. Section 3 details our proposed methodology, explaining the integration of the Firefly Algorithm, LSH, and LFW. Section 4 presents the experimental setup and results, demonstrating the effectiveness of our approach. Finally, Section 5 concludes the article by summarizing our contributions and proposing future research directions.

2. Related Work

T Record linkage has been extensively studied due to its importance in data integration and consolidation. Traditional methods can be broadly categorized into supervised and unsupervised approaches. Supervised methods rely on labeled data to train a model capable of predicting whether two records refer to the same entity [7-9]. These methods require a careful selection of blocking predicates, whose combination and evaluation directly impact the performance of the process. While effective, these methods are limited by their dependence on high-quality labeled data, which is often scarce and expensive to obtain. Additionally, supervised methods struggle to scale to large datasets with multiple attributes, and they require manual tuning of parameters, which can be time-consuming and error-prone. In contrast, unsupervised blocking techniques have gained popularity due to their ability to handle large-scale datasets without the need for labeled data. Progress in this area has been advanced by Kejriwal and Miranker [10] who developed a method for learning unsupervised blocking schemes. This method automatically generates labeled data from a target dataset, leveraging the potential of machine learning to enhance the accuracy of data linkage.

Their process begins by grouping records that share similar tokens, thereby forming initial clusters. Subsequently, a predefined window size is iteratively moved across these clusters, and pairs of records within this window are assessed. This assessment is conducted using the Term Frequency–Inverse Document Frequency (TF-IDF) measure [11] widely recognized for its ability to assess the relevance of words within a text. This technique quantifies the significance of terms within a specific context, thus facilitating the accurate identification of matches between records. This approach proves particularly valuable for efficiently processing large volumes of data while minimizing the need for manual intervention in the data linkage process. Unsupervised blocking methods classify data pairs as positive or negative based on their similarity score, established against specific thresholds. These classifications are used to evaluate blocking predicates through the Fisher Score [12] allowing for effective differentiation without the need for pre-labeled data, thereby reducing costs and labor. However, adjusting parameters such as window size, the number of pairs to label, and similarity thresholds is crucial for optimizing system performance. The computational load of the automatic labeling process increases with the dataset size, which may limit the applicability of this method in contexts involving large and complex data sets.

Canopy Clustering [8, 13-16] involves creating overlapping clusters around randomly selected centroids, with records grouped according to predefined distance thresholds. The accuracy of selecting

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 6: 846-863, 2025
DOI: 10.55214/25768484.v9i6.7970

distance thresholds and the number of centroids is essential for ensuring the effectiveness of this method. Despite its promise, this approach may present practical limitations depending on the type of data used.

Canopy Clustering, when properly set up, can be more effective than traditional blocking methods. However, choosing the right distance thresholds is essential because too broad or too restrictive thresholds can adversely affect recall and precision rates. The authors of Fisher, et al. [17] introduced a clustering approach that uses string similarity to regulate block size, aiming to find the right balance between block quality and size to enhance performance.

Moreover, the article O'Hare, et al. [1] discusses an unsupervised blocking method that simplifies data linkage by automating the selection and weighting of predicates. Yet, this method might struggle with generalizability across complex data sets and lacks transparency in its processes, which could make understanding and verifying the outcomes challenging. Carefully selecting parameters is vital for the success of Canopy Clustering, necessitating focused attention to ensure accuracy while reducing potential biases and computational demands.

Locality-Sensitive Hashing [18, 19] is a technique that groups similar records together using hash functions aligned with a particular distance metric. It's known for its scalability and has a computational complexity of O (n), but it does require a lot of memory and may not perform well with datasets that have many duplicates. To address these challenges, several variations like FastMap [20] SparseMap [21] MetricMap [22] and LSB [23] have been introduced. Notably, MinHash LSH [24, 25] improves the estimation of Jaccard similarity [2] while also reducing the amount of storage needed. This approach breaks down records into k-shingles that are then represented in a sparse Boolean matrix. Through random permutations of the matrix rows, MinHash LSH calculates the Jaccard similarity, creating MinHash signatures that help efficiently cluster records by their similarity. The success of this technique largely depends on carefully adjusting parameters to balance speed and accuracy effectively.

The study in Cui [26] explores a two-level blocking method for entity resolution that combines textual similarity with semantic constraints. By leveraging Locality-Sensitive Hashing (LSH) and a domain tree, this approach addresses the limitations of traditional methods, offering both scalability and accuracy.

In Gionis, et al. [18] the authors tackle the challenge of performing nearest-neighbor queries in high-dimensional databases, which are often affected by the "curse of dimensionality." To improve search efficiency and scalability, they propose a specialized hashing strategy.

The work in Karapiperis and Verykios [27] introduces the Λ-fold Redundant Blocking Framework, which applies LSH to identify pairs of anonymized records. It evaluates how different hashing function families perform when dealing with anonymized data and their ability to preserve distances. To further enhance security, the study presents a Secure Multi-Party Computation (SMC) protocol that allows safe comparison of record pairs without compromising privacy.

In Karapiperis and Verykios [27] the focus is on iterative record linkage, where match and merge operations are repeated until convergence. The proposed [28] Iterative Locality-Sensitive Hashing (ILSH) technique dynamically merges LSH-based hash tables to improve blocking efficiency and accuracy. Additionally, a suite of record linkage algorithms called HARRA (Hashed Record Linkage) is introduced, optimizing performance by leveraging dataset characteristics.

The study in Simonini, et al. [29] presents Blast, an entity resolution method that integrates "loose" schema information and uses LSH for scalable processing. The approach consists of three key phases: extracting schema information, schema-aware blocking, and schema-aware meta-blocking. By leveraging attribute partitioning and aggregate entropy, Blast enhances traditional blocking methods and improves the quality of generated blocks.

The research in Wang, et al. [25] introduces a semantic-aware blocking framework for entity resolution, which incorporates both textual and semantic features through LSH. The study examines how similarity metrics perform across LSH families and explores the integration of semantic similarity across different similarity spaces, ultimately improving blocking precision.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 6: 846-863, 2025
DOI: 10.55214/25768484.v9i6.7970

Feature selection is a fundamental challenge in record linkage, as it aims to identify the optimal subset of variables to enhance the accuracy and efficiency of record matching methods. However, since this problem is NP-hard, exact methods quickly become impractical due to their high computational complexity. To address this limitation, research increasingly favors stochastic approaches, particularly heuristics and metaheuristics, which provide effective solutions for feature selection [30]. Three main strategies are commonly used in this context: Filter, Hybrid, and Embedded approaches [31]. Various metaheuristic techniques have been explored to tackle this challenge. Some studies focus on using a single metaheuristic [8, 32-36] while others combine multiple metaheuristic techniques [37-43] to optimize both exploration and exploitation in the feature selection process.

In this perspective, study Abualigah [6] presents a two-stage text document clustering method. The first stage applies the Particle Swarm Optimization algorithm to optimize feature selection, while the second stage incorporates several advanced versions of the Krill Swarm Algorithm to refine document clustering. Inspired by this approach, I integrated the first stage into my own project to enhance feature selection, leveraging its results to develop a more targeted and effective methodology. To overcome the limitations of existing record linkage methodologies, we propose an integrated approach combining unsupervised blocking techniques, the Firefly Algorithm, and LSH. This methodological synergy optimizes feature selection while improving clustering efficiency, offering a more robust and scalable solution. By enhancing both accuracy and scalability, our approach makes a significant contribution to advancements in the field of record linkage.

3. Research Method

Our proposed methodology consists of three main components: feature selection using the Firefly Algorithm, dimensionality reduction using Locality-Sensitive Hashing (LSH), and data preprocessing using Length-based Feature Weighting (LFW). The goal is to optimize the record linkage process by improving both the accuracy and efficiency of blocking. The Firefly Algorithm [44] is used to identify relevant features, complemented by the Learning Feature Weighting (LFW) strategy [6] to better distinguish informative attributes from non-relevant elements. To optimize the blocking phase, we integrate Locality-Sensitive Hashing (LSH) [45] reducing the number of required comparisons while improving the overall accuracy of record linkage. Experiments conducted on medium-sized datasets demonstrate significant improvements in both speed and accuracy. Designed for scalability, our approach represents a significant advancement in record linkage optimization and its large-scale application(Figure 2).

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 6: 846-863, 2025
DOI: 10.55214/25768484.v9i6.7970

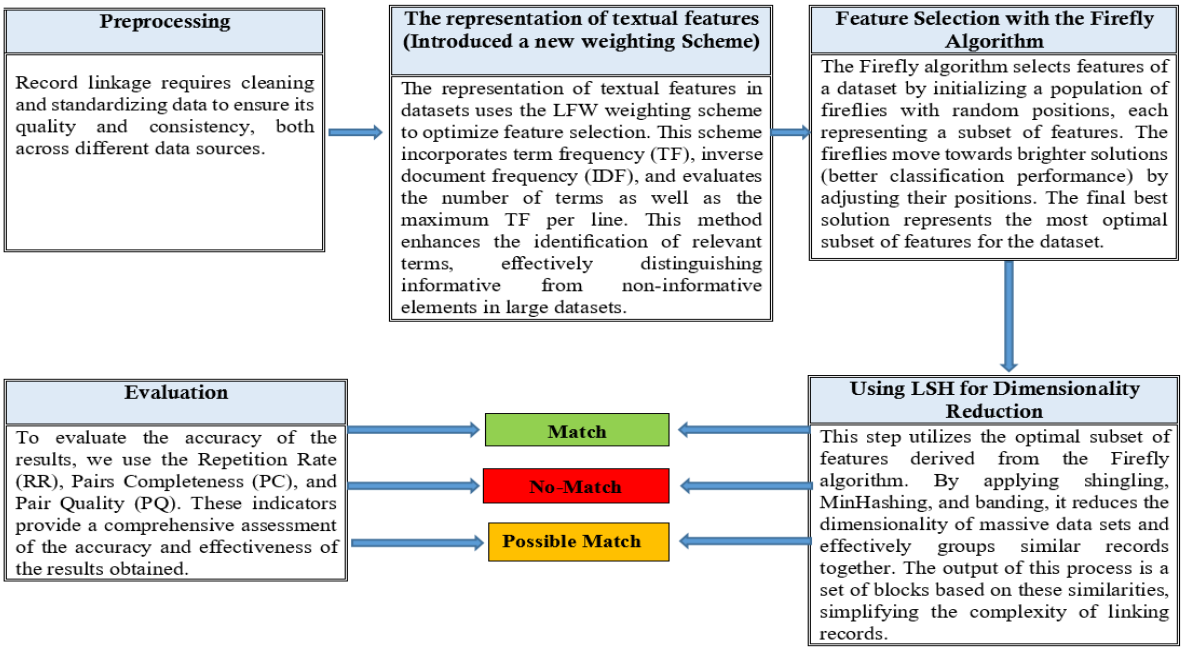

Figure 2.
Our General Approach for Optimizing Record Linkage with the Firefly Algorithm and LSH.

3.1. Feature Selection Using the Firefly Algorithm (FSUFA)

3.1.1. Step 1-Representing Dataset Rows with LFW

- Each dataset row is represented as a "firefly"
- The attributes of each row are concatenated to form a text.
- The size of each dataset row is evaluated (Size I).
- LFW assigns a weight to each term based on its frequency and significance in the resulting text. This weighting is computed using Equation (5):

$$LFW_{\{i,j\}} = \frac{TF(i,j)\times DF(j)}{\max TF(i)\times \log\left(\frac{n}{DF(j)}\right)} \quad (5)$$

Example:

Table 1.
Two sample tuples from the restaurant dataset [2].

Name	Address	City	Cuisine
Arnie morton's of chicago	435 s. la cienega blv	los angeles	American
Campanile	624 s. la brea ave.	los angeles	American

The two records presented in Table 1 are sample entries from a restaurant dataset. In the application of the FAFS to the information from the two restaurants, each row is treated as a document, and each word is considered a potential feature. Feature initialization is performed using the LFW method, which adjusts the initial representation of terms based on their relative importance. For example, the terms "arnie" (0.5), "morton's" (0.8), "chicago" (0.7), and "american" (0.8) are assigned specific weightings, reflecting their relevance in the context of record linkage. These initial values refine the document representation by highlighting the most discriminative terms, thereby enhancing the efficiency of the feature selection process.

3.1.2. Step 2 - Discretization of Positions with The Sigmoid Function

The LFW output vector, which represents a firefly, is subjected to position discretization using the sigmoid function. The process is mathematically defined by Equation (6):

$$S(x_{\{i,j\}}) = \frac{1}{1+\exp^{(-x_{\{i,j\}})}} \quad (6)$$

Where $x_{i,j}$ denotes the position of a firefly i in dimension j, indicative of the intensity of the feature considered for selection. This function converts the continuous intensity of features into probabilities between 0 and 1.

If (rand < S(x_{ij})) then x_{ij}= 1; else x_{ij}=0;

The selection of features by each firefly is determined by a stochastic condition:

- If rand<S(x_{ij}), then x_{ij}=1, meaning that firefly ***i*** selects feature ***j***, Otherwise x_{ij}=0, indicating that the feature is omitted.

Here, randrand is a random number between 0 and 1, introducing an element of chance into the feature selection process. This allows the fireflies to adaptively navigate through the feature space in search of the optimal combination.

3.1.3. Step 3 - Calculation of Intensity with the Objective Function

This approach leverages the LFW method to assess the relevance and uniqueness of dataset rows by identifying the most significant terms while considering their rarity within the dataset. By assigning an LFW-based score, it enables the identification of distinctive records, facilitating their classification and grouping based on their specificity. At a key stage of the algorithm, the objective function evaluates the intensity of each firefly (representing a dataset row) by summing the LFW weights assigned to the document's terms, thereby estimating its contribution to record differentiation. The computation is formally given by Equation (7):

$$\text{score} = \sum_{j=1}^{d} LFW(x_{ij}) \quad (7)$$

Here, ***d*** represents the length of the row (size), x_{ij} is the j-th component (term) of the i-th "firefly", and LFW (x_{ij}) is the LFW weighting associated with the term x_{ij}. This objective function measures the quality of each term, guiding the iterative optimization process towards more relevant solutions.

3.1.4. Step 4 - Optimization by Movement of "Fireflies"

In the algorithm, the movement of fireflies is designed to optimize feature selection by prioritizing the most relevant ones. This process enables the exploration of the solution space to identify new term combinations, avoids local minima through an integrated random component, and gradually converges toward an optimal solution. The goal is to ensure a coherent and informative representation of each dataset row.

The fundamental principle is to guide the fireflies toward positions with higher light intensity, representing improved feature selection quality. This step is carried out according to the following process:

Intensity Comparison: For each pair of "fireflies" (dataset row), the algorithm compares their respective intensities. If the intensity of "firefly" k is greater than that of "firefly" I (Ik> Ii), it indicates that the dataset row representation associated with k is better than that associated with i.

Movement of "Fireflies": When a "firefly" k is brighter than i, "firefly" i is moved towards k to improve its own intensity. This movement is a function of the distance between the "fireflies," their relative attractiveness, and a random component.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 6: 846-863, 2025
DOI: 10.55214/25768484.v9i6.7970

Calculation of the Distance r: The distance between two "fireflies" is calculated using the Euclidean distance. This allows for determining the proximity in the solution space. This is mathematically expressed in Equation (8):

$$r_{ik} = \sqrt{\sum_{j=1}^{d} \left(x_{ij} - x_{kj}\right)^2} \quad (8)$$

Calculation of Attractiveness β: The attractiveness is a function of the distance between the fireflies. A shorter distance and a higher intensity of the target firefly result in greater attractiveness. This is defined in Equation (9):

$$\beta(r) = \beta_0 * e^{-\gamma r} \quad (9)$$

An alternative expression of the distance is also given in Equation (10):

$$r(i,k) = \left|x(\vec{i}) - x(\vec{k})\right| = \sqrt{\sum_{j=1}^{d} \left(x_{ij} - x_{kj}\right)} \quad (10)$$

Updating the Intensity: Where β0 is the attractiveness at r=0, and γ is a light absorption coefficient. The position of a "firefly" in the algorithm is updated by moving closer to brighter fireflies and incorporating a random movement, This is formally defined in Equation (11)::

$$x_i = x_i + \beta(r) * (x_k - x_i) + \alpha(rand - \frac{1}{2}) \quad (11)$$

This approach maintains a balance between attraction to optimal solutions and exploration of the search space to enhance the diversity of considered solutions. At the end of the process, the best-performing positions are converted into a binary representation, facilitating the identification of the most relevant features for the application of the LSH method.

To ensure a reliable and efficient exploration of the feature space, we adopted the parameter values recommended by Arora and Anand [38]. Specifically, the initial attractiveness coefficient was set to $\boldsymbol{\beta_o}$=1.0, the light absorption coefficient to $\boldsymbol{\gamma}$=1.0, and the randomization parameter to $\boldsymbol{\alpha}$=0.2. These values are known to provide a good balance between local exploitation and global exploration, which supports robust and stable optimization performance in bio-inspired algorithms. The pseudo-code depicts the proposed Feature Selection Using the Firefly Algorithm..

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 6: 846-863, 2025
DOI: 10.55214/25768484.v9i6.7970

Table 2.
Feature Selection: Firefly-LFW Algorithm.

Algorithm 1: Firefly Algorithm for Feature Selection

1: Input: Dataset
2: A firefly is a line in the dataset = (z_1, z_2, ..., z_d).
3: Concatenation of the line attributes.
4: LFW representation for each segment (Eq. 1).
5: Discretization according to (Eq. 2 & Eq. 3).
6: Randomly generate a swarm of fireflies.
7: Calculate the dimension of the fireflies' position: d = line size.
8: Define the light intensity as score I = score (Eq. 4).
9: Set a maximum number of iterations: *maxIterations*.
10: Initialize t = 0.
11: while t < maxIterations do
12: for i = 1 to n do
13: for k = 1 to l do
14: if $I_k > I_i$ then
15: /* Move firefly i towards firefly k following these steps */
16: Calculate the distance r (Eq. 5).
17: Calculate the attractiveness β (Eq. 6 & Eq. 7).
18: Calculate the new position z_i of the firefly (Eq. 8).
19: Evaluate the firefly by updating the light intensity (score: Eq. 4).
20: end if
21: Rank the fireflies and find the current best.
22: Record the position of the current best firefly (in real values).
23: Discretize the real position of the best firefly (Eq. 2 & Eq. 3).
24: end for
25: end for
26: Increment $t = t + 1$.
27: end while
28: Return: Selected relevant features.

3.2. Dimensional Reduction With LSH

In the second phase of our methodology, we extend the process initiated with feature selection using the Firefly-LFW algorithm by applying dimensionality reduction based on LSH. This step relies on three complementary techniques: shingling, MinHashing, and banding, which optimize data structuring and organization. The objective is to efficiently group similar rows while reducing the computational complexity of record linkage. Integrating LSH into the clustering process enhances both the accuracy and efficiency of record matching while minimizing the computation time required to identify similar row pairs. The key steps of this approach are presented in the following pseudocode.

The pseudo-code (See Table 3) illustrates the proposed Dimensionality Reduction with LSH Algorithm.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 6: 846-863, 2025
DOI: 10.55214/25768484.v9i6.7970

Table 3.
Algorithm for Record Linkage Using LSH and Firefly with LFW.

Algorithm 2: Record Linkage Using LSH and Firefly with LFW

```
Input: Dataset, Threshold
Output: Set of similar pairs
Apply Function Firefly LFW Algorithm to dataset D to obtain the selected features;
Initialize hash functions;
Initialize hash tables;
procedure LSH(D, t)
    Initialize an empty set S;
    for each item x in D do
        for each hash function h do
            Compute the hash code h(x);
            Insert x into the corresponding bucket in the hash table using the hash code;
        end for
    end for
    for each item x in D do
        for each hash function h do
            Compute the hash code h(x);
            Retrieve the bucket from the hash table using the hash code;
            for each item y in the bucket do
                if similarity(x, y) ≥ t then
                    Add (x, y) to S;
                end if
            end for
        end for
    end for
    return S;
end procedure
```

The Length-based Feature Weighting technique is used to assign weights to features based on their frequency and importance in the dataset. This step enhances the representation of the data and facilitates more accurate feature selection, ultimately improving the quality of the record linkage process.

4. Results and Discussion

In our experimental evaluation, we compared our blocking approach to the one described in O'Hare, et al. [1] which also assesses its performance against several existing methods. To ensure a thorough analysis and an objective comparison, we included the same reference techniques in our study. This approach allowed us to evaluate performance using key metrics such as RR, PC, and PQ for the blocking phase, as well as Precision (Prec), Recall (Rec), and their harmonic mean for the record matching phase. To maintain analytical consistency, we adopted a two-step approach similar to the one described in O'Hare, et al. [1] combining a blocking phase followed by a record matching process. However, instead of using the TF-IDF method, we used the LFW technique for the matching phase. This strategic choice enabled us to make direct comparisons with the results reported in O'Hare, et al. [1]. The objective of this study is to precisely evaluate the effectiveness of our method compared to existing approaches and to highlight the advantages it offers in terms of performance and accuracy in record linkage.

4.1. Benchmarking Datasets

As part of our evaluation, we selected five datasets to assess the performance of our approach across various contexts. These datasets are essential for evaluating the effectiveness and relevance of our method in real-world data scenarios. Among them, two are specifically dedicated to deduplication, while the remaining three focus on record linkage. Each dataset presents distinct challenges, such as variations in data structuring, the presence of redundant values, and the need to establish correspondences between heterogeneous databases. By leveraging this diversity, our study aims to

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 6: 846-863, 2025
DOI: 10.55214/25768484.v9i6.7970

demonstrate the capability of our approach to accurately identify and process duplicates, ensure precise record matching, and adapt to various domains and data structures. The characteristics of the selected datasets are detailed in Table 4.

Table 4.
Table of Characteristics of the Evaluated Datasets.

Dataset Name	Type	Records	Number of Attributes	True matches
Restaurant	Deduplication	864	5	112
DBLP/ACM	Linkage	4910	4	2224
Amazon/GoogleProduct	Linkage	4598	4	1300
Clean-Synth	Deduplication	10 000	10	2000
Dirty-Synth	Deduplication	10 000	9	26 692

4.2. Experimental Results and Discussion

The performance of our approach was rigorously evaluated across multiple datasets, each posing specific challenges in record linkage. On the Restaurant dataset, our method achieved outstanding results, with a Recall Ratio (RR) of 0.999, Pair Completeness (PC) of 0.998, and an F-measure of 0.998, indicating a near-perfect balance between recall and completeness. Most notably, it attained a Pair Quality (PQ) of 0.612, significantly higher than competing methods, which ranged between 0.001 and 0.466, despite high PC values. These results confirm the ability of our approach to minimize false positives while preserving high recall, demonstrating its robustness and effectiveness for deduplication tasks in challenging scenarios. A comparative summary of the numerical results is presented in Table 5, and a visual representation of the performance is shown in Figure 3.

Table 5.
Elaborate on the numerical results related to the restaurant.

Method	RR	PC	*F*RR.PC	PQ
Our method	0.999	0.998	0.998	0.612
Non-standard key-based blocking	0.999	0.991	0.995	0.466
SortedNeighbourhood	0.772	1.000	0.871	0.001
CanopyClustering	0.998	1.000	0.999	0.176
MinHash-LSH	0.981	1.000	0.990	0.016
Méta-bloquants	0.936	0.998	0.965	0.015

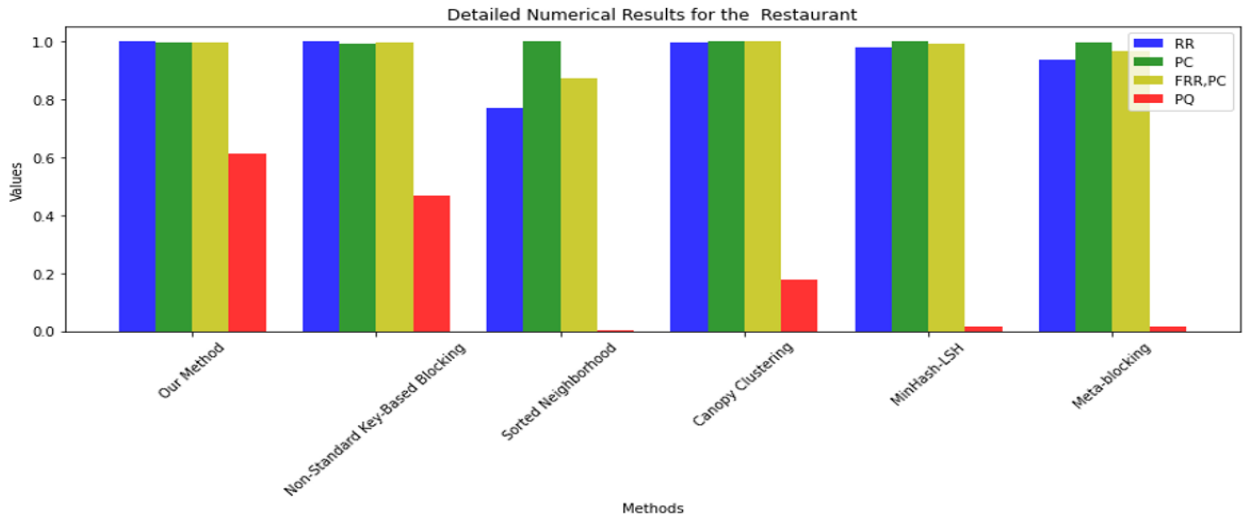

Figure 3.
Detail numerical for the restaurant.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 6: 846-863, 2025
DOI: 10.55214/25768484.v9i6.7970

Table 6 presents the numerical results obtained on the DBLP-ACM dataset, highlighting the performance of our method in comparison to several established techniques. The evaluation of our method on the DBLP-ACM dataset shows a Reduction Rate (RR) of 0.971, which is slightly lower than that of Non-Standard Key-Based Blocking (0.990). This difference reflects a deliberate trade-off, where our approach performs slightly more comparisons in order to avoid missing relevant matches. This strategic choice is justified by the results on other metrics: our method achieves a Pair Completeness (PC) of 0.998, compared to 0.987 for Non-Standard Blocking, indicating a superior ability to cover true matches. The most decisive factor remains the Pair Quality (PQ): our approach reaches 0.263, while all other methods, including Non-Standard Blocking, fail to exceed 0.036. This clearly shows that our system produces significantly fewer false positives, while maintaining nearly complete coverage. Therefore, although our method achieves a slightly lower reduction, it provides much higher overall precision. It strikes an optimal balance between RR, PC, and PQ, demonstrating its robustness, noise control, and relevance in contexts where the quality of matches is critical.

Table 6.
Elaborate on the numerical results related to the DBLP-ACM.

Method	RR	PC	*F*RR, PC	PQ
Our method	0.971	0.998	0.984	0.263
Non-standard key-based blocking	0.990	0.987	0.988	0.036
SortedNeighbourhood	0.939	0.990	0.964	0.005
CanopyClustering	0.897	0.979	0.937	0.004
MinHash-LSH	0.958	0.929	0.943	0.008
Méta-bloquants	0.960	0.999	0.979	0.034

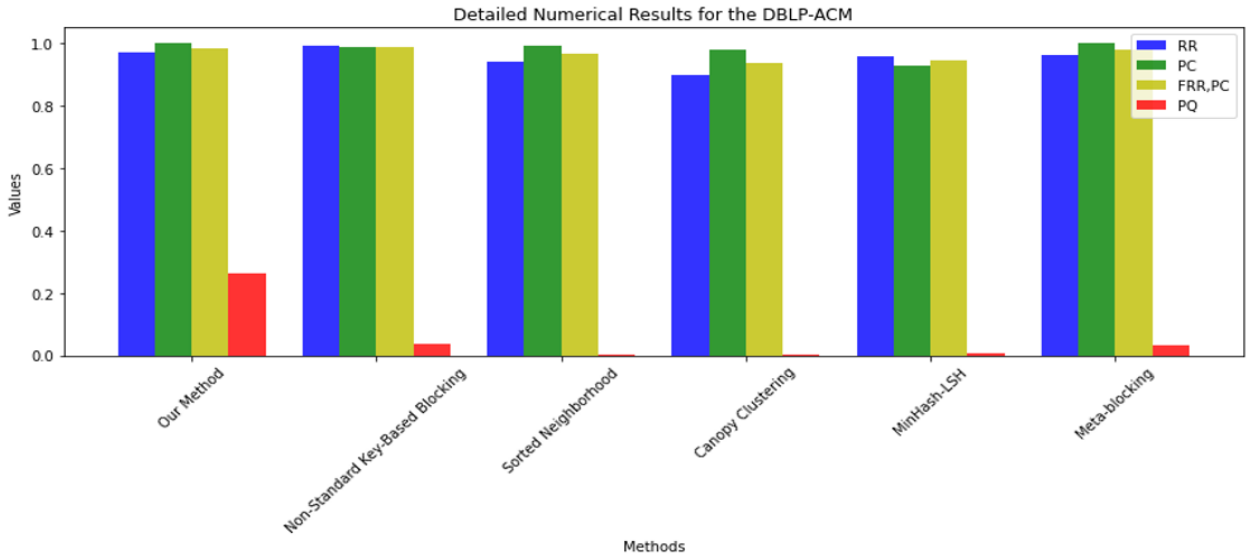

Figure 4.
Detail numerical for the DBLP-ACM.

The numerical evaluation results for the Amazon-Google dataset are presented in Table 7, offering a comparative perspective on the performance of our method against several baseline techniques.On the Amazon-Google dataset, our method achieved strong results in terms of reduction rate (RR = 0.971), pair completeness (PC = 0.962), and F-measure (0.966), demonstrating its ability to maintain an effective balance between efficiency and coverage in a highly heterogeneous data context. It significantly outperforms the baseline methods, particularly in RR and F-measure, where approaches like Non-Standard Blocking and MinHash-LSH show notably weaker performance. However, the pair quality

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 6: 846-863, 2025
DOI: 10.55214/25768484.v9i6.7970

(PQ = 0.007) remains modest, a limitation observed across most methods on this dataset. This can be attributed to the high lexical and structural variability in the data, which makes precise match detection more difficult. Despite this, our method stands out for its overall stability and confirms its adaptability to complex datasets, while highlighting the potential for future improvements through the integration of more advanced semantic matching techniques.

Table 7.
Elaborate on the numerical results related to the Amazon-Google.

Method	RR	PC	*F*RR,PC	PQ
Our method	0.971	0.962	0.966	0.007
Non-standard key-based blocking	0.779	0.954	0.858	0.001
SortedNeighbourhood	0.933	0.627	0.750	0.002
CanopyClustering	0.956	0.820	0.883	0.006
MinHash-LSH	0.418	0.802	0.549	0.000
Méta-bloquants	0.907	0.913	0.904	0.009

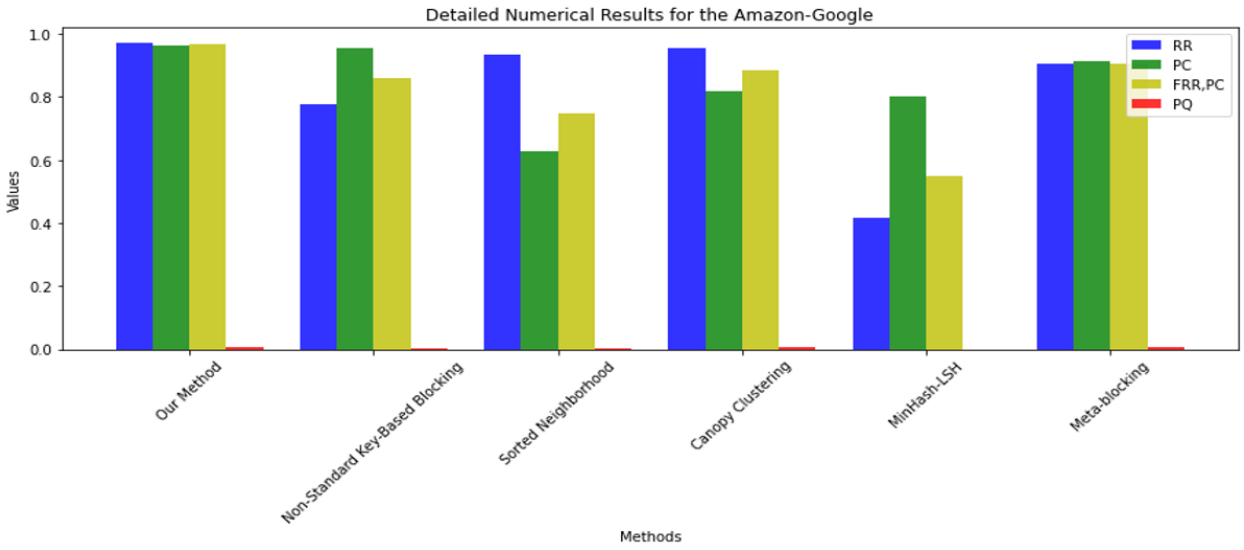

Figure 5.
Detail numerical for the Amazon-Google.

The Clean-Synth dataset is a well-structured synthetic dataset, free from noise or significant variations, making it ideal for evaluating performance in a controlled deduplication setting. On this dataset, our method achieved near-perfect results, with a Reduction Rate (RR) of 0.999, a Pair Completeness (PC) of 0.998, and an F-measure of 0.998, demonstrating its ability to cover almost all true matches while significantly limiting the number of comparisons. Most notably, it achieved a Pair Quality (PQ) of 0.175, which is substantially higher than all competing methods, whose scores remain very low (below 0.016). This shows that our approach is able to maintain high precision, even when other methods achieve high completeness at the cost of many false positives. These results confirm the robustness and accuracy of our method in clean, well-structured synthetic environments.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 6: 846-863, 2025
DOI: 10.55214/25768484.v9i6.7970

Table 8.
Elaborate on the numerical results related to the Clean-Synth.

Method	RR	PC	*F*RR,PC	PQ
Our method	0,999	0.998	0,998	0.175
Non-standard key-based blocking	0.997	1.000	0.998	0.012
SortedNeighbourhood	0.960	1.000	0.980	0.001
CanopyClustering	0.985	1.000	0.992	0.003
MinHash-LSH	0.991	1.000	0.996	0.005
Méta-bloquants	0.987	1.000	0.993	0.016

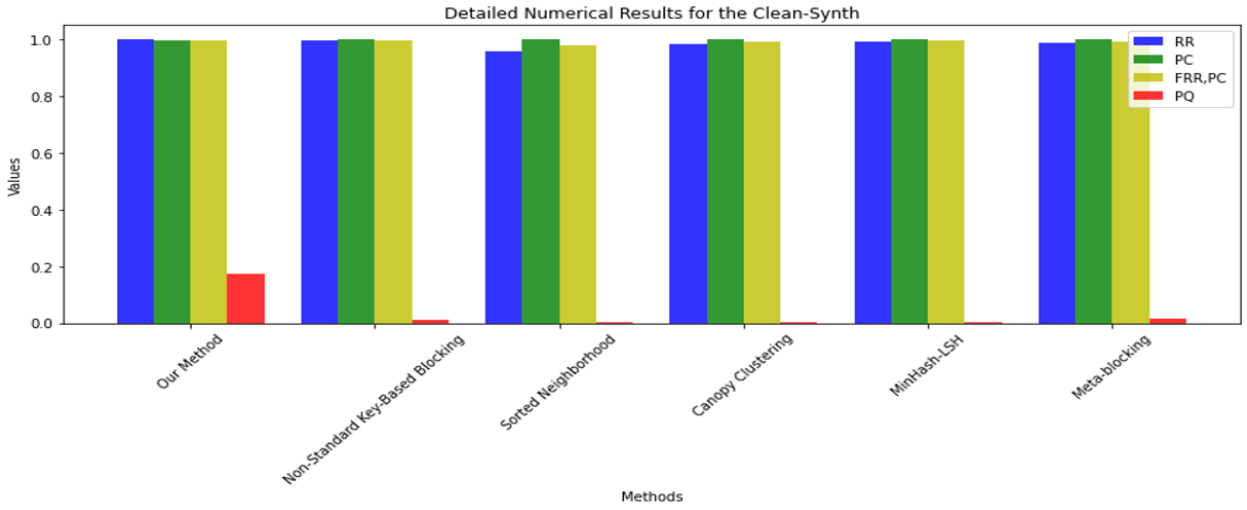

Figure 6.
Detail numerical for the Clean-Synth.

Table 9.
Elaborate on the numerical results related to the Dirty-Synth.

Method	RR	PC	*F*RR,PC	PQ
Our method	0,999	0,997	0,997	0,881
Non-standard key-based blocking	0.997	1.000	0.998	0.173
SortedNeighbourhood	0.964	1.000	0.982	0.015
CanopyClustering	0.999	1.000	1.000	0.969
MinHash-LSH	0.967	0.999	0.983	0.016
Méta-bloquants	0.993	0.926	0.953	0.239

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 6: 846-863, 2025
DOI: 10.55214/25768484.v9i6.7970

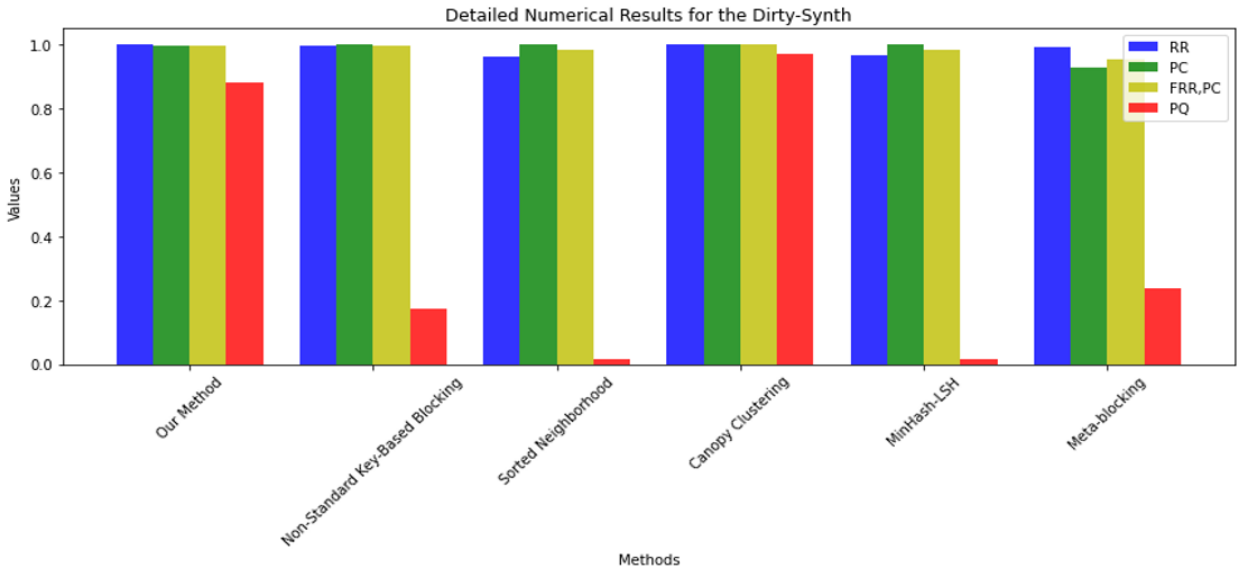

Figure 7.
Detail numerical for the Dirty-Synth.

From a methodological perspective, the experimental results confirm the relevance and effectiveness of our approach, particularly its ability to maintain an optimal balance between Pair Completeness and Pair Quality, which are key indicators of performance in record linkage systems. However, it is important to note that this study focuses exclusively on quality metrics, without including an analysis of execution time or an assessment of computational complexity. This omission represents a significant limitation, especially in the context of Big Data, where computational efficiency is a critical factor for the practical deployment of any solution, particularly in real-time applications or when processing large-scale datasets.

This methodological choice is deliberate, as the primary objective of this initial study was to demonstrate the capacity of the proposed approach to deliver high-quality results. Nevertheless, a more in-depth analysis of scalability, time complexity, and resource consumption will be conducted in future work. This analysis will be carried out within a distributed computing environment, such as Hadoop, in order to leverage parallelism and evaluate the method's performance under more realistic and large-scale conditions. Through this perspective, we aim to confirm the operational feasibility of our approach while ensuring a robust trade-off between efficiency, accuracy, and scalability in complex application scenarios.

5. Conclusion

In this article, we have proposed a novel unsupervised record linkage approach that combines the Firefly Algorithm, Locality-Sensitive Hashing (LSH), and Length-based Feature Weighting (LFW) to optimize data quality in Big Data environments. Our experimental results demonstrate the effectiveness of this approach, particularly in terms of balancing Pairs Completeness (PC) and Pairs Quality (PQ), two critical metrics for evaluating record linkage performance. These results confirm the robustness, relevance, and flexibility of our method, which adapts well to different data contexts while ensuring high performance. However, despite the promising quality outcomes, a significant limitation of our work lies in the absence of execution time analysis and computational complexity assessment. In a Big Data context, where computational efficiency is a critical factor, it becomes essential to evaluate the scalability of the proposed method and its capacity to operate in distributed environments, particularly on platforms such as Hadoop. This evaluation will be the focus of dedicated future research.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 6: 846-863, 2025
DOI: 10.55214/25768484.v9i6.7970

In addition, several improvement paths will be explored, including the integration of machine learning techniques to enhance both the accuracy and efficiency of record linkage, as well as a deeper exploration of semantic analysis through advanced meaning-processing mechanisms to improve data interpretation and utilization. Furthermore, we plan to conduct an in-depth investigation of Arabic language processing, leveraging its morphological and syntactic specificities to refine linkage performance in this linguistically complex context. All these research directions are part of a broader effort to optimize large-scale data management, address Big Data challenges, and enhance both the semantic and linguistic dimensions of the record linkage process for more precise, contextualized, and meaningful data analysis.

Transparency:

The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

Copyright:

References

[1] K. O'Hare, A. Jurek-Loughrey, and C. de Campos, "An unsupervised blocking technique for more efficient record linkage," *Data & Knowledge Engineering*, vol. 122, pp. 181-195, 2019.

[2] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*. United States: Prentice-Hall, Inc, 1988.

[3] F. Guillet and H. J. Hamilton, *Quality measures in data mining*. Berlin, Germany: Springer, 2007.

[4] E. Emary, H. M. Zawbaa, K. K. A. Ghany, A. E. Hassanien, and B. Parv, "Firefly optimization algorithm for feature selection," in *Proceedings of the 7th Balkan Conference on Informatics Conference*, 2015, pp. 1-7.

[5] O. Jafari, P. Maurya, P. Nagarkar, K. M. Islam, and C. Crushev, "A survey on locality sensitive hashing algorithms and their applications," *arXiv preprint arXiv:2102.08942*, 2021. https://doi.org/10.48550/arXiv.2102.08942

[6] L. M. Q. Abualigah, *Feature selection and enhanced krill herd algorithm for text document clustering*. Berlin: Springer, 2019.

[7] K. O'Hare, A. Jurek-Loughrey, and C. d. Campos, "A review of unsupervised and semi-supervised blocking methods for record linkage," *Linking and Mining Heterogeneous and Multi-view Data*, pp. 79-105, 2019. https://doi.org/10.1007/978-3-030-01872-6_4

[8] M. Bilenko, B. Kamath, and R. J. Mooney, "Adaptive blocking: Learning to scale up record linkage," presented at the Sixth International Conference on Data Mining (ICDM'06), 2006.

[9] M. Michelson and C. A. Knoblock, "Learning blocking schemes for record linkage," in *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06)*, 2006, vol. 6, pp. 440-445.

[10] M. Kejriwal and D. P. Miranker, "An unsupervised algorithm for learning blocking schemes," presented at the 2013 IEEE 13th International Conference on Data Mining, 2013.

[11] A. Bilke and F. Naumann, "Schema matching using duplicates," in *21st International Conference on Data Engineering (ICDE'05)*, 2005: IEEE, pp. 69-80.

[12] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*, 2nd ed. New York: Wiley-Interscience, 2000.

[13] B. V. Babu and K. J. Santoshi, "Unsupervised detection of duplicates in user query results using blocking," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 3, pp. 3514-3520, 2014.

[14] M. Kejriwal and D. P. Miranker, "On linking heterogeneous dataset collections," in *Proceedings of the ISWC 2014 Posters & Demonstrations Track: A track within the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014 (CEUR Workshop Proceedings,*, 2014, pp. 217-220.

[15] M. Kejriwal and D. P. Miranker, "A two-step blocking scheme learner for scalable link discovery," *OM*, vol. 14, pp. 49-60, 2014.

[16] A. McCallum, K. Nigam, and L. H. Ungar, "Efficient clustering of high-dimensional data sets with application to reference matching," in *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000, pp. 169-178.

[17] J. Fisher, P. Christen, Q. Wang, and E. Rahm, "A clustering-based framework to control block sizes for entity resolution," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 279-288.

[18] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proceedings of the 25th International Conference on Very Large Data Bases (VLDB) VLDB Endowment*, 1999, vol. 99, no. 6, pp. 518-529.

[19] R. C. Steorts, S. L. Ventura, M. Sadinle, and S. E. Fienberg, "A comparison of blocking methods for record linkage," in *Privacy in Statistical Databases: UNESCO Chair in Data Privacy, International Conference, PSD 2014, Ibiza, Spain, September 17-19, 2014. Proceedings*, 2014: Springer, pp. 253-268.

[20] C. Faloutsos and K.-I. Lin, "FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets," in *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, 1995, pp. 163-174.

[21] G. Hristescu and M. Farach-Colton, "Cluster-preserving embedding of proteins," Technical Report 99-50, Computer Science Department, Rutgers University, 1999.

[22] J. T.-L. Wang, X. Wang, K.-I. Lin, D. Shasha, B. A. Shapiro, and K. Zhang, "Evaluating a class of distance-mapping algorithms for data mining and clustering," in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining ACM*, 1999, pp. 307-311.

[23] A. Sohail and W. u. Qounain, "Locality sensitive blocking (LSB): A robust blocking technique for data deduplication," *Journal of Information Science*, vol. 50, no. 6, pp. 1400-1413, 2024. https://doi.org/10.1177/01655515221121963

[24] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive datasets*, 2nd ed. Cambridge University Press. https://doi.org/10.1017/cbo9781139924801, 2014.

[25] Q. Wang, M. Cui, and H. Liang, "Semantic-aware blocking for entity resolution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 1, pp. 166-180, 2015. https://doi.org/10.1109/TKDE.2015.2468711

[26] M. Cui, "Towards a scalable and robust entity resolution-approximate blocking with semantic constraints," Technical Report, Australian National University, 2014.

[27] D. Karapiperis and V. S. Verykios, "An LSH-based blocking approach with a homomorphic matching technique for privacy-preserving record linkage," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 4, pp. 909-921, 2014. https://doi.org/10.1109/TKDE.2014.2349916

[28] H.-s. Kim and D. Lee, "HARRA: fast iterative hashed record linkage for large-scale data collections," in *Proceedings of the 13th International Conference on Extending Database Technology*, 2010, pp. 525-536.

[29] G. Simonini, S. Bergamaschi, and H. Jagadish, "BLAST: A loosely schema-aware meta-blocking approach for entity resolution," *Proceedings of the VLDB Endowment*, vol. 9, no. 12, pp. 1173-1184, 2016. https://doi.org/10.14778/2994509.2994533

[30] P. Agrawal, H. F. Abutarboush, T. Ganesh, and A. W. Mohamed, "Metaheuristic algorithms on feature selection: A survey of one decade of research (2009-2019)," *IEEE Access*, vol. 9, pp. 26766-26791, 2021. https://doi.org/10.1109/ACCESS.2021.3056407

[31] Z. Sadeghian, E. Akbari, H. Nematzadeh, and H. Motameni, "A review of feature selection methods based on meta-heuristic algorithms," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 37, no. 1, pp. 1-51, 2025. https://doi.org/10.1080/0952813X.2023.2183267

[32] J. Bala, J. Huang, H. Vafaie, K. DeJong, and H. Wechsler, "Hybrid learning using genetic algorithms and decision trees for pattern classification," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, 1995, pp. 719-724.

[33] N. Dif and Z. Elberrichi, "An enhanced recursive firefly algorithm for informative gene selection," *International Journal of Swarm Intelligence Research*, vol. 10, no. 2, pp. 21-33, 2019. https://doi.org/10.4018/IJSIR.2019040102

[34] A. Chaudhuri, "Search space division method for wrapper feature selection on high-dimensional data classification," *Knowledge-Based Systems*, vol. 291, p. 111578, 2024. https://doi.org/10.1016/j.knosys.2024.111578

[35] A. Zakeri and A. Hokmabadi, "Efficient feature selection method using real-valued grasshopper optimization algorithm," *Expert Systems with Applications*, vol. 119, pp. 61-72, 2019. https://doi.org/10.1016/j.eswa.2018.10.021

[36] B. Selvakumar and K. Muneeswaran, "Firefly algorithm based feature selection for network intrusion detection," *Computers & Security*, vol. 81, pp. 148-155, 2019. https://doi.org/10.1016/j.cose.2018.11.005

[37] M. Mafarja, I. Aljarah, H. Faris, A. I. Hammouri, A. M. Al-Zoubi, and S. Mirjalili, "Binary grasshopper optimisation algorithm approaches for feature selection problems," *Expert Systems with Applications*, vol. 117, pp. 267-286, 2019. https://doi.org/10.1016/j.eswa.2018.09.015

[38] S. Arora and P. Anand, "Binary butterfly optimization approaches for feature selection," *Expert Systems with Applications*, vol. 116, pp. 147-160, 2019. https://doi.org/10.1016/j.eswa.2018.08.051

[39] M. Ragab, "Hybrid firefly particle swarm optimisation algorithm for feature selection problems," *Expert Systems*, vol. 41, no. 7, p. e13363, 2024. https://doi.org/10.1111/exsy.13363

[40] N. Dif, M. w. Attaoui, and Z. Elberrichi, "Gene selection for microarray data classification using hybrid meta-heuristics," presented at the International Symposium on Modelling and Implementation of Complex Systems, 2018.

[41] H. M. Zawbaa, E. Emary, C. Grosan, and V. Snasel, "Large-dimensionality small-instance set feature selection: A hybrid bio-inspired heuristic approach," *Swarm and Evolutionary Computation*, vol. 42, pp. 29-42, 2018. https://doi.org/10.1016/j.swevo.2018.02.021

[42] M. Sarhani, A. E. Afia, and R. Faizi, "Facing the feature selection problem with a binary PSO-GSA approach," *Recent Developments in Metaheuristics*, pp. 447-462, 2018. https://doi.org/10.1007/978-3-319-58253-5_26

[43] L. Zhang, L. Liu, X.-S. Yang, and Y. Dai, "A novel hybrid firefly algorithm for global optimization," *PloS one*, vol. 11, no. 9, p. e0163230, 2016. https://doi.org/10.1371/journal.pone.0163230

[44] S. L. Marie-Sainte and N. Alalyani, "Firefly algorithm based feature selection for Arabic text classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 32, no. 3, pp. 320–328, 2020.
[45] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC '98)*, 1998.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 6: 846-863, 2025
DOI: 10.55214/25768484.v9i6.7970