Journal of Quality Measurement and Analysis *JQMA* 22(SI) 2026, 277-296
e-ISSN: 2600-8602 *http://www.ukm.my/jqma*
https://doi.org/10.17576/jqma.22si.2026.15

AUTOMATIC RECORD LINKAGE FOR STRUCTURED DATASETS USING UNSUPERVISED RANDOM FORESTS MODEL AND NATURAL BREAK THRESHOLDING

(*Pautan Rekod Automatik bagi Set Data Berstruktur Menggunakan Model Hutan Rawak Tidak Diselia dan Penentuan Ambang Sempadan Semula Jadi*)

ZATURRAWIAH ALI OMAR*, ZAMIRA HASANAH ZAMZURI,
MOHD AFTAR ABU BAKAR & NORATIQAH MOHD ARIFF

ABSTRACT

Record linkage, which identifies the same entities across different datasets, is essential for ensuring data quality but is often manual and prone to errors. This study presents a two-step framework that automates threshold selection and generates training data using unsupervised machine learning and data-driven clustering. In the first step, an Unsupervised Random Forests (URF) model is used to compute similarity scores between record pairs without relying on labelled data. Four natural breaks algorithms, including Fisher (F), Jenks (J), Head/Tail (HT), and *K*-means (KM), are applied to these scores to determine upper and lower thresholds that separate matches from non-matches. These thresholds are then used to construct high-quality training data under "top" and "imbalanced" designs, supporting classifier training in the second step. The approach was evaluated on 13 datasets, including public benchmarks and synthetically corrupted data. Results showed that Random Forests (RF) with F or J breaks achieved the highest stability in numeric datasets, while Support Vector Machine (SVM) with J break performed best on string-only data. Mixed datasets benefited from both RF and SVM, with performance depending on the proportion of textual versus numeric attributes. Compared with the Fellegi-Sunter Expectation Maximisation and EpiLink baselines, the proposed approach achieved equal or superior F1 scores in most cases. Although performance declined in datasets with extreme corruption or poor imputation, the framework demonstrates that high-quality linkage can be achieved without heavy feature engineering or deep embeddings. This highlights its value as a scalable, interpretable, and label-independent solution for structured datasets.

Keywords: record linkage; unsupervised random forest; natural breaks

ABSTRAK

Pautan rekod ialah kaedah untuk mengenalpasti entiti yang sama merentasi set data yang berbeza. Kaedah ini penting dalam memastikan kualiti data, namun sering dilaksanakan secara manual dan terdedah kepada ralat. Kajian ini membentangkan satu kerangka kerja dua langkah yang mengautomasikan pemilihan nilai ambang serta penjanaan data latihan melalui pendekatan pembelajaran mesin tidak diselia serta kaedah pengelompokan berasaskan data. Dalam langkah pertama, model *Unsupervised Random Forest* (URF) digunakan untuk mengira skor kesamaan antara pasangan rekod tanpa memerlukan data berlabel. Empat algoritma penentuan nilai ambang semulajadi, iaitu Fisher (F), Jenks (J), Head/Tail (HT), dan *K*-means (KM), diaplikasikan pada skor kesamaan tersebut untuk menentukan nilai ambang atas dan bawah yang akan memisahkan pasangan rekod sepadan dan tidak sepadan. Pasangan rekod ini kemudiannya akan digunakan bagi membina data latihan berkualiti tinggi melalui pemilihan secara "atas" dan "tidak seimbang", bagi menyokong latihan pengelas dalam langkah kedua. Pendekatan ini dinilai menggunakan 13 set data yang merangkumi set data penanda aras umum

serta data sintetik yang diubah secara terkawal. Keputusan menunjukkan bahawa *Random Forests* (RF) dengan pemisah F atau J mencapai kestabilan tertinggi bagi kumpulan data numerik, manakala *Support Vector Machine* (SVM) dengan pemisah J menunjukkan prestasi terbaik bagi kumpulan data rentetan. Bagi set data campuran, kedua-dua RF dan SVM memberikan prestasi yang baik, bergantung kepada perkadaran atribut teks berbanding numerik. Apabila dibandingkan dengan kaedah asas Fellegi-Sunter melalui pendekatan *Expectation Maximisation* and EpiLink, pendekatan yang dicadangkan mencapai keputusan skor F1 yang setara atau lebih baik dalam kebanyakan kes. Walaupun prestasi menurun bagi set data dengan tahap ralat yang tinggi atau kaedah imputasi yang lemah, kerangka kerja ini menunjukkan bahawa pautan berkualiti tinggi masih boleh dicapai tanpa perlunya kejuruteraan ciri yang kompleks atau penggunaan *deep embeddings*. Hal ini menonjolkan potensinya sebagai kaedah pautan yang berskala, boleh ditafsirkan, dan tidak bergantung kepada set data berlabel bagi data berstruktur.

Kata kunci: pautan rekod; hutan rawak tidak diseslia; sempadan semula jadi

1. Introduction

In today's data-driven landscape, accurate and efficient record linkage is essential for integrating information from disparate sources. Record linkage involves identifying and merging records that refer to the same entity across different datasets. It plays a pivotal role in fields ranging from marketing and fraud detection to public health and social sciences. In administrative contexts, effective record linkage informs critical decision-making processes, enabling organisations to improve customer relationship management, enhance fraud detection mechanisms, and optimise government administration. Beyond administrative uses, this approach supports research in epidemiology, health systems, and social sciences, contributing to public policy development and scientific advancements (Gu *et al.* 2003). However, achieving accurate linkage remains challenging due to growing data volumes, complex data structures and variations across datasets.

Traditionally, record linkage required significant manual effort and was prone to errors. As datasets expanded, the need for scalable and dependable linkage methods intensified. Manual approaches consumed time and introduced inconsistencies, making automation essential for improving efficiency, accuracy, and scalability (Nentwig *et al.* 2017; Rochner & Rothlauf 2024).

Efforts to automate record linkage date back to the 1950s, coinciding with the advent of computers. Newcombe *et al.* (1959) introduced one of the earliest probabilistic linkage approaches, allowing entity matching without unique identifiers. Fellegi and Sunter (1969) formalised probabilistic linkage using decision-based thresholds, later enhanced by Winkler (1993) through the Expectation Maximisation (EM) algorithm to improve probability estimates. Although effective on structured datasets, probabilistic models require manual threshold tuning, which limits their scalability in large-scale data integration tasks.

To address scalability concerns, machine learning techniques gained traction in the early 2000s, enabling data-driven similarity scoring and classification. Supervised models, such as Random Forests and Support Vector Machines, improved linkage accuracy by learning optimal matching rules from labelled data, with applications in domains such as author disambiguation (Treeratpituk & Giles 2009) and financial entity resolution (Kim & Giles 2016). Over the following decade, research expanded toward hybrid and semi-supervised approaches aimed at reducing dependence on labelled data, including frameworks like ReLiC (Varma *et al.* 2019)

and ZeroER (Wu *et al*. 2020). While these studies advanced scalability and reduced manual effort, they continued to rely on the availability of high-quality training datasets, which remain difficult to obtain in many domains where labelled records are scarce or costly to generate.

More recently, deep learning approaches, such as Siamese neural networks and Graph Neural Networks (GNNs), have demonstrated superior accuracy by leveraging complex feature representations and relational dependencies (Barlaug & Gulla 2021; Liu *et al.* 2023). Despite their theoretical advantages, these methods require a large volume of annotated data, extensive computational resources, and expert model tuning, making them impractical for many real-world applications. Additionally, deep learning methods often lack interpretability, complicating their validation in regulatory or sensitive environments. This contrast underscores the need for lighter, interpretable alternatives capable of delivering competitive performance without heavy reliance on labelled data or computationally intensive models.

To overcome these challenges, this paper presents a two-step framework for automating record linkage by integrating unsupervised similarity measures with natural thresholding strategies. The Unsupervised Random Forests (URF) model was utilised as a similarity measure, leveraging its built-in proximity function to compute record similarity. High-quality training data are generated using natural breaks algorithms, including Fisher (F), Jenks (J), Head/Tail (HT), and *k*-means (KM), following the "top" and "imbalanced" constructions (Ali Omar *et al.* 2023). These training datasets support classification tested on four models: Random Forests (RF), Support Vector Machine (SVM), Logistic Regression (LR), and Naive Bayes (NB).

The contributions of this study are threefold:

(1) It proposes a two-step framework that integrates URF with natural breaks algorithms for threshold optimisation and high-quality training data generation.
(2) It provides evidence that the proposed approach achieved equal or better performance than the Fellegi-Sunter model with Expectation Maximisation (FSM-EM) and EpiLink (FSM-Epi) algorithms in nearly all cases.
(3) It offers a scalable, interpretable, and label-efficient solution suitable for structured datasets, reducing the need for manual intervention and labelled data.

The remainder of this paper is structured as follows. Section 2 elaborates on the dataset preparation used in this study, along with the proposed approach and relevant background theory on natural breaks algorithms. In particular, the proposed approach extends previous work that established the top and imbalanced constructions for training data selection. Section 3 presents a detailed examination of the experimental results, focusing on key aspects such as threshold identification, training data quality assessment, and classifier performance comparison. Section 4 discusses the findings in depth, analysing their implications for record linkage methodologies and highlighting potential areas for improvement. Finally, Section 5 concludes the paper by summarising the key contributions and suggesting future research directions to further advance automated record linkage techniques.

2. Materials and Methods

2.1. *Dataset preparation*

This study utilised five primary datasets, which were further processed to generate a total of 13 datasets. The dataset selection combined both real-world benchmarks and synthetic data to ensure comparability and systematic evaluation under controlled error conditions.

The RLdata500 dataset, available in the `RecordLinkage` package in R, is a widely recognised benchmark in record linkage research. It contains 500 records, with 10% duplicated to simulate realistic data entry errors such as typographical and formatting inconsistencies. However, the degree of attribute-level corruption within those duplicates is not specified in the source. Each record represents an individual with attributes for first name, last name, and date of birth (split into year, month, and day). As RLdata500 is already standardised and free from missing values, no additional preprocessing was required.

The Restaurant dataset, available in the R `restaurant` package, was selected as another benchmark containing heterogeneous attributes and duplicate entries from two data sources (Fodor and Zagat). The sources originate from real-world data and contain natural duplicates caused by inconsistent data entries, typographical differences, and incomplete information. As this dataset is not synthetically corrupted, its attribute-level error rate is unspecified. The dataset underwent an initial cleaning process guided by a set of predefined cleaning rules (see Appendix), which standardised attribute formats, resolved inconsistencies, and address certain missing values.

A further cleaning step produced the Restaurant_clean version by refining string-based attributes, removing punctuation, special characters, and spaces, and converting all text to lowercase. This process enhanced text consistency and reduced noise during similarity computation. To isolate the influence of textual attributes, subsets containing only string attributes (Restaurant_str and Restaurant_str_clean) were derived from their respective versions.

The Cora dataset, available in the R `cora` package, is a bibliographic corpus with a high duplication rate and a high proportion of missing values. It was included to expand the diversity of the string-only category and to capture the complexity of citation linkage tasks, specifically illustrating a one-to-many linkage scenario in which multiple variant records correspond to a single true entity. Only attributes with fewer than 3% missing values were retained for the linkage process. The missing entries in these attributes were imputed with a fixed placeholder string to ensure consistency during similarity computation.

To evaluate robustness under controlled conditions, synthetic datasets were generated using GeCo (Tran *et al.* 2013), a web-based tool that introduces realistic corruption to personal data[a]. GeCo provides several corruption functions, including value swapping and missing values, as well as character edits, keyboard errors, optical character recognition (OCR) errors, and phonetic substitutions. These error types emulate common duplication mechanisms in administrative datasets, such as typographical variations, incomplete entries, and inconsistent spellings.

The attribute-wise error rate used in this study follows Garza *et al.* (2024) and is defined as the number of corrupted attribute values divided by the total number of attribute values across all records. This measure provides a consistent way to quantify the severity of corruption across the synthetic datasets.

In the GenData series, each dataset consisted of 550 records in total, comprising 500 original and 50 duplicated records, representing a 10% duplication rate. Corruption was applied only to the duplicated copies. For the first series (Gendata1 - Gendata5), each dataset contained eight attributes. GenData1 was created using GeCo, which targeted four attributes, using either value swapping, missing values, or keyboard edits. During the encoding process, which involved converting the string values to their numerical representations, each record was ensured to have exactly one corrupted attribute, with missing values imputed to an erroneous value. This resulted in 50 affected cells out of 4,400 total attribute values, leading to an attribute-wise error

[a] ANU Online Personal Data Generator and Corruptor (GeCo) is available at https://dmm.anu.edu.au/geco/

rate of 1.14%, as previously defined. In GenData2–GenData5, one additional attribute was corrupted in each subsequent dataset, producing 100, 150, 200, and 250 corrupted cells, respectively. These correspond to attribute-wise error rates of 2.27%, 3.41%, 4.55%, and 5.68%. These incremental corruption levels enabled systematic assessment of linkage robustness as data quality declined.

The second series (GenData4_0 and GenData4_Mod) was slightly different from the first series, as it only consisted of six attributes. Value swapping and missing value corruption were intended to be introduced to four attributes per duplicate. In practice, however, only 93 cells were affected, corresponding to an attribute-wise error rate of 2.82%. Upon closer inspection, we observed that value swapping was not implemented and that only missing values were applied. Missing entries were subsequently replaced with zeros in GenData4_0 or with the mode of the affected attribute in GenData4_Mod. This second series was primarily used to assess the effect of simple imputation strategies on linkage quality. Table 1 summarises the datasets, including record count, attribute type, and the pre-processing steps applied.

For analysis, the datasets were grouped into three categories:

(1) Mixed Data Type (Numeric + String): RLdata500, Restaurant, and Restaurant_clean.
(2) String-Only Datasets: Restaurant_str, Restaurant_str_clean, and Cora.
(3) Numeric-Only Datasets: GenData1 - GenData5, GenData4_0, and GenData4_Mod.

Collectively, these datasets encompass diverse data types, duplication levels, and corruption patterns, forming a comprehensive testbed for evaluating the robustness and scalability of the proposed framework, particularly its adaptability to varying data quality and structure.

Table 1: Summary of datasets with treatment information

Dataset Group	Dataset Name	No. of Records	No. of Attributes	Attribute Data Type	No. of Duplicates	Additional Information
Mixed	RLdata500	500	5	Integer, Character	50	
	Restaurant	864	7	Integer, Character	112	• Fodor – 331 records, • Zagat – 533 records.
	Restaurant_clean	864	7	Integer, Character	112	• Same as Restaurant dataset. • Character attributes underwent additional cleaning.
String	Restaurant_str	864	4	Character	112	• Subset of the Restaurant dataset containing only character attributes.
	Restaurant_str_clean	864	4	Character	112	• Same as Restaurant_str dataset. • All attributes underwent additional cleaning.
	Cora	1879	2	Character	64,578	• Title and author attributes contain fewer than 3% missing values. • The missing values are replaced with a placeholder string.

Table 1 (Continued)

Integer	GenData1	550	8	Integer, Numeric	50	• One attribute was corrupted per record (gender, married status, state, or income). • Error rate: 1.14% • The state attribute follows the 50 U.S. states.
	GenData2	550	8	Integer, Numeric	50	• Two attributes corrupted per record (including those from GenData1 plus day attribute). • Error rate: 2.27%
	GenData3	550	8	Integer, Numeric	50	• Three attributes corrupted per record (including those from GenData2 plus month attribute). • Error rate: 3.41%
	GenData4	550	8	Integer, Numeric	50	• Four attributes corrupted per record (including those from GenData3 plus year attribute). • Error rate: 4.55%
	GenData5	550	8	Integer, Numeric	50	• Five attributes corrupted per record (including those from GenData4 plus gender or marital status attribute). • Error rate: 5.68%
	GenData4_0	550	6	Integer, Numeric	50	• Four attributes were corrupted (intended as swapped or missing values, but only missing values were consistently applied). • Gender and state were most affected. • Error rate: 2.82% • The states attribute includes 54 categories, including blanks. • Missing values replaced with 0.
	GenData4_Mod	550	6	Integer, Numeric	50	• Same as GenData4_0, but missing values replaced with the mode of each attribute. • Mode for Gender: 1, State: 28

2.2. *The proposed approach*

The proposed approach adopts the two-step structure for automating record linkage described by Christen (2007), comprising training data generation as the first step and classification as the second. Our contribution lies in the training data creation step, where URF (Breiman 2001) was integrated as the similarity measure with natural break algorithms to automate threshold

selection. This integration enables the generation of high-quality training labels without requiring manual intervention.

2.2.1. *Step 1: Training data selection*

To construct high-quality training data, record pairs were first evaluated using the URF model, which generated similarity scores based on proximity measures. For two datasets, A and B, record pairs (i, j) are formed, where $i \in A$ and $j \in B$. The URF computed a similarity score $S\{i, j\}$ for each pair, producing a proximity matrix P that captured relationships among records. The similarity score ranged between *0* and *1*, with value $S\{i, j\} = 1$ indicating a perfect match, and value $S\{i, j\} = 0$ representing a definite non-match. The pairs with the perfect match and definite non-match were excluded in the threshold's selection. Using natural breaks algorithms (see Section 2.3), thresholds were applied to divide pairs into three categories:

(1) Match pairs (*M*): $S\{i, j\} \geq$ upper threshold (T_u) and labelled as matches.
(2) Non-match pairs (*NM*): $S\{i, j\} \leq$ lower threshold (T_l), labelled as non-matches, selected to satisfy the imbalanced ratio.
(3) Possible match pairs (*PM*): Pairs with $T_l < S\{i, j\} < T_u$, plus residual pairs below T_l not included in *NM*, requiring classification.

The training dataset T therefore consisted of M and NM as well as the perfectly matched pairs and is formally defined as:

$$T = \{(i, j) | S\{i, j\} \geq T_u\} \cup \{(i, j) | S\{\mathrm{i}, \mathrm{j}\} \leq T_l\}, \ \forall\, i < j \tag{1}$$

To maintain the desired imbalance between matches and non-matches, an adjusted imbalanced ratio r^* was introduced. The ratio leverages the number of identified matches in T, as a proxy for the true number of matches, which is unknown in practice. It is defined by:

$$r^* = \big||A| + |B| - |M|\big|/|M| \tag{2}$$

where $|A|$ and $|B|$ denote the number of records from two datasets respectively, and $|M|$ is the number of match pairs in T. Therefore, if A and B contain 100 records, and M is identified as 20, then $r^* = |100 + 100 - 20|/20$, which equals to $r^* = 9$.

2.2.2. *Step 2: Test data classification*

Once the training data was constructed, classification was applied to *PM* to determine whether they belonged to the *M* or *NM* category. Four machine learning classifiers were used to perform this task, which are RF, SVM, LR, and NB. These four classifiers are considered the state-of-the-art learning-based entity matching solution for traditional classifiers. The performance is also comparable to that of the deep learning approach for structured data (Mudgal *et al.* 2018).

Each classifier underwent 10-fold cross-validation, repeated three times, to provide stable performance estimates and to support hyperparameter tuning. All models were implemented using the `caret` package in R, which automatically optimises hyperparameters using grid search. For RF, `mtry` was optimised; for SVM, the cost parameter `C`; none in LR; and for NB, the `fL`, `usekernal`, and `adjust` parameters. Default settings were used for parameters not included in the tuning grid.

2.2.3. *Performance evaluation metrics*

Since each dataset included gold standard labels, training data quality was assessed as the percentage of correctly labelled pairs over the total labelled pairs (Christen 2007). Classification performance was evaluated using standard metrics: recall (*R*), precision (*P*), and *F*1 score. *R* measures the ability to correctly identify true match pairs. *P* indicates the proportion of predicted matches that are correct. High *R* but low *P* means the method successfully identified most true matches but also produced many false matches. Conversely, high *P* but low *R* indicates that most predicted matches are correct, but many true matches are missed. The *F*1 score provides a balanced single measure between *R* and *P*, rewarding methods that achieve both adequate coverage of true matches and control false matches.

Let *TP* denote the number of correctly identified match pairs (true positives), *FN* the number of missed match pairs (false negatives), and *FP* the number of incorrectly classified match pairs (false positives). The metrics for *R*, *P*, and *F*1 are defined as:

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

$$Precision = \frac{TP}{TP+FP} \tag{4}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \tag{5}$$

True negatives (*TN*) are not included in these measures because the number of non-matched record pairs grows combinatorially and does not correspond to a meaningful count of non-matched entities. As noted in prior work, any quality measure that incorporates *TN* can therefore produce deceptive results (Christen & Goiser 2007).

2.3. *Threshold optimisation using Natural Breaks*

Threshold optimisation in record linkage requires methods that adapt to the distribution of similarity scores. Traditional approaches often rely on fixed cut-offs or manual tuning, which can be arbitrary and dataset-specific. Natural break algorithms instead detect discontinuities in ordered data and place boundaries where groups of scores are most distinct. This makes them well suited for threshold determination, as thresholds emerge from the data itself rather than external rules.

To automate threshold selection, four natural break algorithms were employed, specifically F, J, HT, and KM. Their implementation was available in the `classInt` package in R. Each algorithm partitions the similarity scores distribution into intervals, from which T_u and T_l were derived. Specifically, T_u was defined as the minimum value of the highest interval, while T_l was defined as the minimum value in the lowest interval. This follows the assumption that higher similarity scores indicate a greater likelihood of a match, while lower scores indicate a greater likelihood of a non-match (Xu & Chan 2020). For F, J, and KM, the number of classes (*K*) was determined using Sturges' formula (Sturges 1926) when not otherwise specified. Default parameter settings were maintained throughout. This choice evaluates the framework's out-of-the-box utility, demonstrating its effectiveness without extensive hyperparameter tuning.

When applied, these algorithms determine thresholds by locating natural gaps in the similarity score distribution. By minimising within-group variance or recursively partitioning values, they position boundaries at the points of greatest separation between groups of scores. For example, the F and J optimisation minimises intra-class variance and maximises inter-class variance, thereby placing thresholds at natural discontinuities in the similarity distribution. The

output of each algorithm is a set of intervals that partition the similarity scores, from which the T_u and T_l thresholds are determined.

2.3.1. *Fisher break*

The F algorithm (Fisher 1958) minimises within-class variance while maximising between-class variances. It systematically explores all possible contiguous breakpoints with a dynamic programming search that guarantees the optimal partition for a given number of classes (K). For each prefix of the sorted data and class count, the algorithm records the minimum achievable within-class sum of squares. It then backtracks to recover the class boundaries that minimise the objective function:

$$min \sum_{k=1}^{K} \sum_{x_i \in C_k} (x_i - \mu_k)^2 \tag{6}$$

where K is the number of classes, C_k is the set of values in class k, x_i is an observation, and μ_k is the class mean. This method is deterministic and returns a globally optimal partition for the chosen K. In effect, the algorithm ensures that values within the same class are as similar as possible, while values in different classes are as distinct as possible.

2.3.2. *Jenks break*

The J natural breaks algorithm (Jenks & Caspall 1971) shares the same objective as the F algorithm, namely to minimise within-class variance while maximising between-class variances. Unlike F, which guarantees a global optimum through dynamic programming, J employs an iterative heuristic. It begins from an initial partition, usually based on equal counts or equal intervals for a given K, and then improves this partition through successive refinements. At each step, the algorithm examines the boundary between two adjacent classes and tests whether moving the cut point along the sorted data reduces the total within-class variance and increases the Goodness of Variance Fit (GVF). If the reassignment of values across the boundary improves GVF, the boundary is updated, and the process continues until no further improvement is achieved or a preset iteration limit is reached.

The GVF is defined as:

$$GVF = 1 - \frac{\sum_{k=1}^{K} \sum_{x_i \in C_k} (x_i - \mu_k)^2}{\sum (x_i - \mu)^2} \tag{7}$$

where μ is the overall mean, and μ_k is the mean of class C_k. The GVF represents the proportion of total variance explained by the classification, with values closer to one indicating tighter and better-separated classes. As Jenks relies on local adjustments, it may converge to a local rather than global optimum.

2.3.3. *Head/Tail break*

HT break (Jiang 2013) is designed for heavy-tailed distributions where a few large values dominate many smaller values. It recursively partitions data into head (values > mean) and tail (values ≤ mean) segments, following the steps:

(1) Compute the mean of the dataset.
(2) Partition the data into:
    (i) Head: Values greater than the mean.

(ii) Tail: Values less than or equal to the mean.

(3) If the head contains more than 40% of the data, repeat the process recursively on the head.

This procedure produces a hierarchical set of classes without requiring a preset of *K*. At each step, the mean is recomputed for the current head segment and used as a split point. The iteration stops when the head is no longer sufficiently smaller than the tail. In this study, the stopping criterion was set to 40%, following Jiang's recommendation that a threshold around this level balances two goals: capturing the recursive structure of heavy-tailed data while avoiding excessive fragmentation. The 40% rule is not fixed by theory and can be tuned. It was adopted here to ensure comparability with prior studies and to provide a reproducible stopping condition for deriving the T_u and T_l.

2.3.4. *K-means clustering*

KM clustering (Macqueen 1967) partitions data into *K* clusters by minimising the sum of squared distances between data points and their cluster centroids. The objective function is:

$$min \sum_{k}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2 \quad (8)$$

where μ_k is the centroid of a cluster C_k. The algorithm alternates between assigning each point to its nearest centroid and recomputing the centroids as cluster means until assignments no longer change or a maximum number of iterations is reached. The result depends on the initial centroids and can converge to a local minimum rather than a global optimum.

KM was included with F, J, and HT because it provides a complementary perspective. F and J optimise variance across intervals, while HT recursively splits heavy-tailed data. KM groups similarity scores around centroids, with thresholds taken from the boundaries between clusters. This centroid-based approach complements the interval and recursive methods by capturing patterns they may overlook.

3. Results

This section presents the experimental results for threshold identification, training data quality assessment, and classifier performance evaluation.

3.1. *Thresholds and imbalanced ratio identification*

The T_u and T_l, the $|M|$ in the training data, and the r^* were derived using F, J, HT, and KM breaks. Across all datasets, the T_l remained consistent across break algorithms after rounding. However, the J break algorithm initially produced an insufficient number of match pairs in integer datasets with ≥2.27% data errors. To address this, the T_u was lowered by 0.1 to ensure an adequate training set. The r^* varied depending on the dataset and the number of matched pairs identified. Table 2 summarises the threshold values and their corresponding imbalanced ratios for each dataset.

Table 2: Upper threshold, number of matched pairs, and adjusted imbalanced ratios

Dataset	F			J			HT			KM		
	T_u	$\|M\|$	r*	T_u	$\|M\|$	r*	T_u	$\|M\|$	r*	T_u	$\|M\|$	r*
RLdata500	0.90	28	1:16	0.99	6	1:82	0.76	38	1:12	0.67	40	1:11
Restaurant	0.73	60	1:13	0.98	23	1:36	0.68	70	1:11	0.73	60	1:13
Restaurant_clean	0.78	45	1:18	0.99	22	1:38	0.72	67	1:11	0.62	89	1:8
Restaurant_str	0.79	34	1:24	0.98	22	1:38	0.62	86	1:9	0.58	113	1:6
Restaurant_str_clean	0.79	35	1:23	0.99	23	1:36	0.49	188	1:3	0.62	89	1:8
Cora	0.89	4828	1:1	0.99	4539	1:1	0.73	5747	1:1	0.87	4903	1:1
GenData1	0.77	26	1:20	0.97	5	1:109	0.60	36	1:14	0.77	26	1:20
GenData2	0.84	13	1:41	0.87	10	1:24	0.79	17	1:31	0.67	22	1:24
GenData3	0.70	12	1:44	0.71	12	1:44	0.62	16	1:33	0.57	22	1:24
GenData4	0.57	11	1:49	0.62	8	1:67	0.61	8	1:67	0.45	21	1:25
GenData5	0.43	11	1:49	0.46	7	1:77	0.39	23	1:22	0.36	36	1:14
GenData4_0	0.56	15	1:35	0.70	3	1:182	0.73	3	1:182	0.56	15	1:35
GenData4_Mod	0.85	8	1:67	0.98	4	1:136	0.66	18	1:29	0.55	28	1:18
Mean	0.74			0.86			0.65			0.62		
SD	0.14			0.18			0.11			0.13		

r* denotes the ratio of matched to non-matched pairs (*M:NM*) in the training dataset. For example, a value of 1:16 indicates that for every one matched pair, there are 16 non-matched pairs.

In general, the J break algorithm produced the highest T_u values (mean = 0.86), but with smaller $|M|$ and higher r^*. By contrast, HT and KM tended to yield lower T_u values, larger $|M|$, and lower r^* in several datasets. The F break algorithm often produced intermediate thresholds, with $|M|$ and r^* that were more stable across datasets. Notably, datasets with higher error rates (e.g., GenData4 and GenData5) showed more extreme imbalance under J, whereas HT and KM remained comparatively larger $|M|$ even in these challenging cases. Finally, across all methods, the Cora dataset stood out as an exception, as it consistently produced large training sets and near-balanced ratios regardless of threshold choice.

3.2. *Training data quality*

To evaluate the effectiveness of the training data selection process, the quality of $|M|$ and $|NM|$ was assessed for each natural break method. The quality of the definite non-match pairs denoted as $|P_0|$ was also evaluated at the dataset level. Table 3 summarises the results across datasets.

The quality of $|M|$ was generally high, particularly for F and J breaks, where half of the datasets were above 90% correctly labelled. However, HT and KM showed reduced $|M|$ quality in string-only and error-prone datasets, with the lowest values observed in Restaurant_str_clean (31.91% for HT) and GenData5 (30.56% for KM). GenData4_0 was an extreme case, where all methods produced 0% $|M|$.

The quality of $|NM|$ remained stable across methods, with nearly all selected non-matches correctly labelled at or near 100%. In contrast, $|P_0|$ showed slight but notable variation. While many datasets achieved 100% correctness, deviations were observed in string-only datasets

(e.g., Restaurant_str and Restaurant_str_clean) and in Cora (97.75%), as well as in numeric-only datasets with higher error rates (≥4.55%).

Table 3: Training data quality for matched pairs, non-matched pairs, and true non-matched pairs

Training Set	$\|M\|$				$\|NM\|$				$\|P_0\|$
Break	F	J	HT	KM	F	J	HT	KM	
Dataset									
RLdata500	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Restaurant	93.33	100.00	91.43	93.33	100.00	100.00	100.00	100.00	100.00
Restaurant_clean	91.00	100.00	91.00	82.00	100.00	100.00	100.00	100.00	100.00
Restaurant_str	79.41	95.45	61.63	50.44	100.00	100.00	99.87	99.56	99.99
Restaurant_str_clean	77.14	95.65	31.91	52.81	100.00	100.00	99.65	99.86	99.99
Cora	94.88	95.02	93.37	94.76	99.86	99.85	99.88	99.86	97.75
GenData1	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
GenData2	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
GenData3	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
GenData4	90.91	87.50	87.50	76.19	100.00	100.00	100.00	100.00	99.99
GenData5	45.45	42.86	34.78	30.56	100.00	100.00	100.00	100.00	99.99
GenData4_0	0.00	0.00	0.00	0.00	100.00	100.00	100.00	100.00	100.00
GenData4_Mod	100.00	100.00	72.22	53.57	100.00	100.00	100.00	99.80	100.00
Median	93.33	100.00	91.00	82.00	100.00	100.00	100.00	100.00	100.00

3.3. *Classifier performance evaluation*

The classification performance of RF, SVM, LR, and NB was evaluated using R (Eq. (3)), P (Eq. (4)), and $F1$ score (Eq. (5)) as the main metrics across datasets and threshold selection methods. Table 4 compares the best break-classifier combinations with the best optimal method, with linkage quality (LQ) labels following Zhu *et al.* (2015).

RF consistently achieved the highest $F1$ scores, particularly when combined with F and J breaks, and was the dominant performer in mixed and numeric datasets. SVM also performed strongly in string-only datasets, where J-SVM achieved high recall (e.g., Restaurant_str: R = 0.95, $F1$ = 0.90), in some cases surpassing the optimal FSM baseline. In contrast, LR and NB rarely appeared among the best combinations. LR only emerged in Cora, where both break-based and optimal methods performed poorly ($F1 \approx 0.40$).

Comparisons with optimal methods showed that break-based approaches often matched or exceeded baseline performance in clean or moderately noisy datasets (e.g., RLdata500, Restaurant, GenData1, GenData2, GenData4_Mod). However, in high-error datasets such as GenData5 and GenData4_0, break-based methods deteriorated sharply ($F1$ = 0.71 and 0.14, respectively), while optimal methods maintained stronger performance, including perfect classification in GenData4_0.

Table 4: Performance comparison of the best Break-Classifier combinations and the best optimal method

Dataset	Best Break-Classifier	*P*	*R*	*F*1	LQ	Best Optimal Method	*P*	*R*	*F*1	LQ
RLdata500	F-RF/J-RF	1.00	0.98	0.99	Very good	FSM-EM/FSM-Epi	0.98	0.94	0.96	Very good
Restaurant	J-RF	0.90	1.00	0.95	Very good	FSM -Epi	0.91	0.95	0.93	Very good
Restaurant_clean	J-SVM	0.90	0.96	0.94	Very good	FSM -Epi	0.90	0.96	0.93	Very good
	J-RF	0.91	0.99	0.94						
Restaurant_str	J-SVM	0.60	0.95	0.90	Very good	FSM -Epi	0.95	0.74	0.83	Fair
Restaurant_str_clean	J-SVM	0.79	0.77	0.91	Very good	FSM -Epi	0.85	0.84	0.85	Good
Cora	HT-LR	0.87	0.14	0.40	Very poor	URF[b]	0.49	0.37	0.42	Very poor
GenData1	F-RF/ HT-RF/ KM-RF	0.98	1.00	0.99	Very good	FSM -EM	0.98	1.00	0.99	Very good
GenData2	F-RF/ HT-RF	1.00	0.96	0.98	Very good	FSM -EM/URF	0.96	0.88	0.92	Very good
GenData3	KM-RF	0.98	0.90	0.96	Very good	FSM -EM	0.82	0.84	0.83	Fair
GenData4	HT-RF	0.93	0.86	0.88	Good	FSM -Epi	0.63	0.44	0.52	Poor
GenData5	J-RF	0.56	0.88	0.71	Poor	FSM -Epi	0.64	0.28	0.39	Very poor
GenData4_0	J-RF/ HT-RF	0.26	0.10	0.14	Very poor	FSM -EM/ FSM -Epi	1.00	1.00	1.00	Very good
GenData4_Mod	F-RF/J-RF	1.00	1.00	1.00	Very good	FSM -EM/ FSM -Epi	1.00	1.00	1.00	Very good

[b]The URF listed under Best Optimal Method refers to a single-step linkage using the URF proximity scores paired with manually optimised threshold.

A box plot was used to visually compare classifier and break performance across dataset groups, focusing on the *F*1 score (Figure 1). In mixed data type datasets, RF with J break achieved the highest *F*1 scores, performing near perfectly. SVM performed slightly lower, while LR and NB struggled with low *F*1 scores due to low precision. For string-only datasets, SVM with J break outperformed other methods, demonstrating its adaptability in handling text-based attributes. In numeric-only datasets, RF with F and J breaks consistently achieved the highest F1 scores, while HT and KM breaks resulted in greater variability.

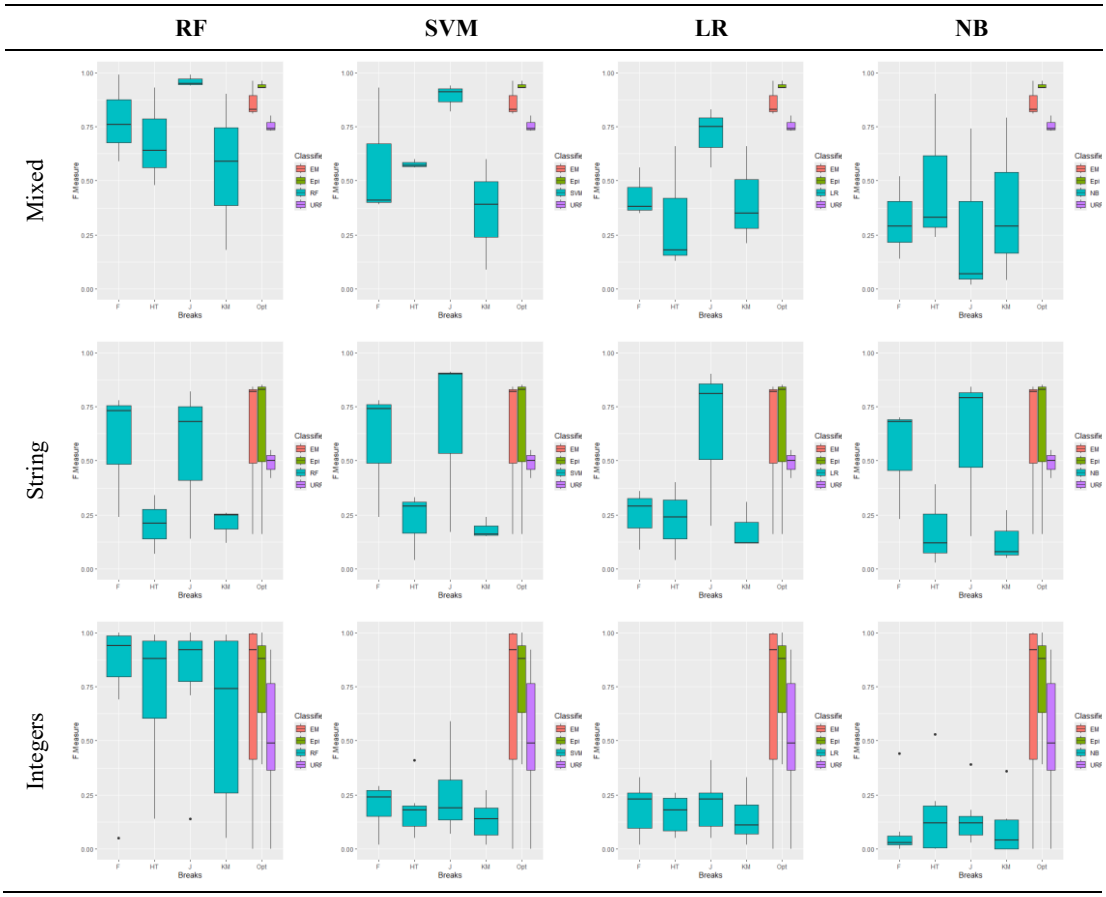

Figure 1: Performance comparison of the best Break-Classifier combinations and the optimal method by dataset group

4. Discussion

4.1. *Impact of threshold selection on training data quality and classification*

Threshold selection determines how similarity scores are partitioned, which explains the patterns observed in Table 2. F and J breaks consistently produced high-quality |M| and |NM| because they both minimised within-group variance and maximised between-group variance (Fisher 1958; Jenks & Caspall 1971). This variance-based design creates sharp separations in the similarity distribution, clarifying why these methods yielded stable thresholds and well-separated training data. Similar findings were reported by Henckaerts *et al.* (2018) and Saleh (2024), who showed that variance-minimising segmentation improves classification accuracy.

The J break, however, often produced the highest T_u values. This reflects its iterative optimisation, which tends to push the upper threshold closer to the extreme end of the similarity distribution. While this improves the purity of match labels, it also reduces the number of admitted pairs. In datasets with errors ≥2.27%, the J break often set the upper threshold too high, admitting too few match pairs and, in some cases, none at all. This triggered classifier warnings. F breaks were less extreme, producing intermediate thresholds that consistently

yielded sufficient match pairs, making them more dependable in numeric datasets with missing values.

HT breaks behaved differently. Their recursive partitioning is designed for heavy-tailed distributions (Jiang 2013). This characteristic explains why HT performed well when similarity scores were skewed, as the method naturally aligned with the data's scaling properties. Lin (2013) also showed that HT breaks are effective in hierarchical data, which supports this observation. However, in datasets without heavy-tailed structures, HT thresholds became unstable. This instability explains the sharp drop in $|M|$ quality in string-only datasets such as Restaurant_str_clean.

KM breaks were the most inconsistent. The reliance of KM on centroid placement means that thresholds are sensitive to initialisation and the assumption of spherical clusters (Macqueen 1967; Nigro & Cicirelli 2023). In skewed or noisy datasets, centroids fail to align with the true similarity structure, producing unstable thresholds and high false positives. This explains the poor $|M|$ quality observed in numeric datasets such as GenData4_Mod. Prior work (Melnykov & Zhu 2019) confirms that KM struggles with skewed or clustered data, which matches our findings.

Despite their methodological differences, all four natural break methods effectively segmented similarity scores into meaningful intervals, reinforcing their utility in automating threshold selection for record linkage. Their ability to eliminate arbitrary cutoffs makes them valuable in real-world applications. However, effectiveness varied by dataset type. In numeric-only datasets, no single method consistently outperformed the others, particularly when error rates were high (≥4.55%), underscoring the persistent challenge of selecting robust thresholds in noisy data (Bohara 2020; Salimi *et al.* 2024). Even so, F break remained the most reliable across numeric distributions, consistently producing sufficient match pairs without requiring additional adjustments.

Another aspect observed was the decline in $|P_0|$ quality in string-only datasets and in numeric datasets with higher error rates (≥4.55%). This reflects a limitation of the URF clustering process when dealing with sparse string features and heavily corrupted values. Limited diversity in attributes reduces the discriminative power of URF, and in cases of severe corruption, some true matches are misclassified with a similarity score of zero. This effect is consistent with the downward trend of T_u across GenData1 to Gendata5, where all break algorithms adjusted to progressively lower thresholds in response to the degraded similarity score distributions.

4.2. *Classifier robustness and performance*

Threshold selection is critical for record linkage. This holds true not only for the traditional FSM model but also for the proposed framework. As discussed in Section 4.1, the F and J breaks generally produced high T_u, which in turn generated the most reliable training data. These higher-quality labels directly supported more stable and accurate classification outcomes. By contrast, when T_u values were set too low, as observed with HT and KM in certain datasets, noisy training labels reduced classification stability, as illustrated in Table 4 and Figure 1 across the dataset groups. The RF model most clearly demonstrates the interplay between threshold determination, training data quality, and classifier stability.

The experimental analysis and box plot evaluations underscore that the RF classifier remains exceptionally robust. RF consistently outperformed other models when trained on data partitions derived from F and J breaks, supporting findings from Ali Omar *et al*. (2023). Its ensemble structure reduces variance and bias (Breiman 2001; Treeratpituk & Giles 2009) and handles imbalanced ratios effectively (Li *et al*. 2020b). This explains why RF maintained high

*F*1 scores in numeric and mixed datasets, even when corruption reached 4.55% in GenData4. However, RF's resilience collapsed in GenData4_0, where missing values were imputed with zeros, producing *F*1 = 0.14. This was predominantly caused by the failure in the initial training data selection step (as shown in Table 3), where all natural break algorithms yielded 0% quality for the matched training set (*M*). Mode imputation in GenData4_Mod restored performance to *F*1 = 1.0, confirming that preprocessing choices are decisive. These results align with Fernando *et al.* (2021), Harron *et al.* (2020) and Farhangfar *et al.* (2008), who emphasised the importance of appropriate imputation strategies.

SVM performed strongly in string-only datasets, particularly when trained on high-quality labels from J breaks. Kernel functions enabled SVM to model non-linear relationships in textual features, allowing it to handle small inconsistencies, such as variations in spacing or spelling, more effectively than linear models. This explains why J-SVM surpassed FSM baselines in Restaurant_str and Restaurant_str_clean, where string inconsistencies dominated. These findings are consistent with Acharya (2024), who reported that SVM outperforms Logistic Regression when data are not linearly separable, owing to its ability to construct flexible decision boundaries. However, as error levels increased in numeric datasets, SVM's recall deteriorated because corrupted or imputed attributes reduced class separability in the feature space.

The LR and NB remained consistently weaker even on datasets with relatively high-quality training labels, such as Rldata500, GenData1–3, and GenData4_mod under F and J breaks. LR's reliance on linear separability limited its ability to capture complex decision boundaries (Acharya 2024), while NB's conditional independence assumption was undermined by correlated attribute patterns (Zhang 2004). Their underperformance in these settings indicates that their weaknesses are structural rather than solely a consequence of noisy training data.

The results collectively indicate that there is a need for dataset-type-aware classifier selection in automated record linkage, rather than a one-size-fits-all approach. The proposed framework does not yield uniform improvements across all dataset types. In numeric datasets, RF showed the best overall performance with all four breaks algorithms achieving the highest *F*1 scores in certain datasets, but F or J breaks showed greater stability. In string-only datasets, SVM with J break was superior, however, the results for Cora dataset reveal a challenge to the FSM as well as the proposed framework. This highlights a boundary for automated linkage when dealing with complex one-to-many linkage relationships and sparse text features without advanced embeddings. Mixed datasets benefited from both RF and SVM, with the optimal break-classifier combination depending on the proportion of textual versus numeric attributes.

The implications are twofold. First, the framework reliably improves record linkage in clean or moderately noisy datasets, reducing reliance on manual tuning. Second, in datasets with extreme corruption or poor imputation (e.g., GenData4_0), its advantage diminishes. In such cases, even high thresholds cannot fully compensate for degraded similarity distributions, and FSM methods may remain competitive. This highlights that, although the proposed method reduces manual intervention and improves scalability, its gains are most pronounced in structured or moderately corrupted datasets.

4.3. *Comparison with other approaches*

Compared with state-of-the-art methods benchmarked on the Restaurant dataset (Eibeck *et al*. 2024), which achieved *F*1 scores of 1.0, the proposed approach differs in three key respects: methodological design, reliance on feature engineering, and empirical performance. One important methodological difference lies in feature generation. Traditional pipelines first construct record pairs and then expand them through extensive feature extraction. For example,

the Magellan framework (Mudgal *et al*. 2018) begins with six attributes and expands them to 86 through distance calculations and encoding transformations, enabling methods like ZeroER (Wu *et al*. 2020) to effectively differentiate true matches from non-matches.

In contrast, the proposed methodology reverses this order by first using the URF model to compute similarity through clustering original records against synthetically generated data, and only then constructing record pairs. This design constrains the URF model to operate on the features directly available in the data. However, additional attributes derived through feature encoding can still be incorporated if needed.

Furthermore, the proposed approach diverges from competitors in its treatment of text attributes. CollaborEM (Ge *et al*. 2023) leverages sentence-BERT for contextual similarity, while DeepMatcher (Mudgal *et al*. 2018) employs FastText embeddings. Additionally, DITTO (Li *et al*. 2020a) integrates transformer-based models with domain insights, and DeepER (Ebraheem *et al*. 2018) uses distributed word representations. Our framework relies on conventional standardisation and basic cleaning functions rather than advanced text processing.

As for empirical performance, although the proposed approach does not yield a perfect F1 score for the Restaurant data, like the aforementioned methods, the linkage quality is still considered "Very good" ($F1 = 0.95$). Crucially, this demonstrates that high-quality linkage can be achieved without heavy feature engineering or deep embeddings, underscoring the novelty and scalability of the proposed framework.

5. Conclusion

This study introduced a two-step framework for automated record linkage that integrates a URF model with natural breaks algorithms to generate high-quality training data without manual threshold tuning. Across 13 datasets, the framework demonstrated that RF with F or J breaks consistently excelled in numeric data, while SVM with J break was most effective in string-only datasets. These findings confirm that dataset-type-aware classifier selection is essential for achieving robust linkage performance.

The framework advances the field by reducing reliance on labelled data and complex feature engineering while still matching or surpassing established baselines, such as FSM-EM and FSM-Epi in most scenarios. Its streamlined reliance on standard attribute cleaning and encoding further enhances scalability and interpretability, making it suitable for structured datasets in real-world applications.

Nonetheless, several limitations should be acknowledged. Performance deteriorated in datasets with extreme corruption or poor imputation, where FSM methods remained competitive. The approach is currently restricted to structured datasets and has not been validated on unstructured or semi-structured data. Threshold stability varied across algorithms, with HT and KM producing inconsistent results in skewed or sparse distributions. Additionally, complex one-to-many linkage relationships like those in the Cora dataset remain a challenge for both the proposed framework and traditional models. Finally, classifier performance was sensitive to preprocessing choices, particularly imputation strategies, which highlights the necessity of careful data preparation.

Future research should focus on adaptive thresholding strategies for highly corrupted data, the integration of more sophisticated imputation methods, and the extension of the framework to unstructured or semi-structured domains. Addressing these challenges will further strengthen the applicability of URF with natural breaks as a practical, scalable, and interpretable solution for automated record linkage.

Acknowledgment

The authors gratefully acknowledge the financial support provided by Universiti Kebangsaan Malaysia (UKM) with project code TAP-K017073.

References

Acharya B.B. 2024. Comparative analysis of machine learning algorithms: KNN, SVM, decision tree and logistic regression for efficiency and performance. *International Journal for Research in Applied Science and Engineering Technology* **12**(11): 614–619.

Ali Omar Z., Zamzuri Z.H., Mohd Ariff N. & Abu Bakar M.A. 2023. Training data selection for record linkage classification. *Symmetry* **15**(5): 1060.

Barlaug N. & Gulla J.A. 2021. Neural networks for entity matching: A survey. *ACM Transactions on Knowledge Discovery from Data* **15**(3): 52.

Bohara B. 2020. Adaptive threshold for online object recognition and re-identification tasks. *arXiv*. https://arxiv.org/abs/2012.14305

Breiman L. 2001. Random forests. *Machine Learning* **45**(1): 5–32.

Christen P. 2007. A two-step classification approach to unsupervised record linkage. *Proceedings of the Sixth Australasian Conference on Data Mining and Analytics,* pp. 111–119.

Christen P. & Goiser K. 2007. Quality and complexity measures for data linkage and deduplication. In Guillet F. & Hamilton H. (eds.). *Quality Measures in Data Mining*: 127–151. Berlin, Heidelberg: Springer.

Ebraheem M., Thirumuruganathan S., Joty S., Ouzzani M. & Tang N. 2018. Distributed representations of tuples for entity resolution. *Proceedings of the VLDB Endowment* **11**(11): 1454–1467.

Eibeck A., Zhang S., Lim M.Q. & Kraft M. 2024. A simple and efficient approach to unsupervised instance matching and its application to linked data of power plants. *Journal of Web Semantics* **80**: 100815.

Farhangfar A., Kurgan L. & Dy J. 2008. Impact of imputation of missing values on classification error for discrete data. *Pattern Recognition* **41**(12): 3692–3705.

Fellegi I.P. & Sunter A.B. 1969. A theory for record linkage. *Journal of the American Statistical Association* **64**(328): 1183–1210.

Fernando M.P., Cèsar F., David N. & José H.O. 2021. Missing the missing values: The ugly duckling of fairness in machine learning. *International Journal of Intelligent Systems* **36**(7): 3217–3258.

Fisher D.W. 1958. On grouping for maximum homogeneity. *Journal of the American Statistical Association* **53**(284): 789–798.

Garza M.Y., Williams T.B., Ounpraseuth S., Hu Z., Lee J., Snowden J., Walden A.C., Simon A.E., Devlin L.A., Young L.W. & Zozus M.N. 2024. Comparing Medical Record Abstraction (MRA) error rates in an observational study to pooled rates identified in the data quality literature. *BMC Medical Research Methodology* **24**(1): 304.

Ge C., Wang P., Chen L., Liu X., Zheng B. & Gao Y. 2023. CollaborEM: A self-supervised entity matching framework using multi-features collaboration. *IEEE Transactions on Knowledge and Data Engineering* **35**(12): 12139–12152.

Gu L., Baxter R., Vickers D. & Rainsford C. 2003. *Record Linkage: Current Practice and Future Directions*. CMIS Technical Report No. 03/83. CSIRO Mathematical and Information Sciences.

Harron K., Doidge J.C. & Goldstein H. 2020. Assessing data linkage quality in cohort studies. *Annals of Human Biology* **47**(2): 218–226.

Henckaerts R., Antonio K., Clijsters M. & Verbelen R. 2018. A data driven binning strategy for the construction of insurance tariff classes. *Scandinavian Actuarial Journal* **2018**(8): 681–705.

Jenks G.F. & Caspall F.C. 1971. Error on choroplethic maps: Definition, measurement, reduction. *Annals of the Association of American Geographers* **61**(2): 217–244.

Jiang B. 2013. Head/tail breaks: A new classification scheme for data with a heavy-tailed distribution. *The Professional Geographer* **65**(3): 482–494.

Kim K. & Giles C.L. 2016. Financial entity record linkage with random forests. *Proceedings of the 2016 ACM SIGMOD International Conference on Management of Data*.

Li Y., Li J., Suhara Y., Doan A. & Tan W.C. 2020a. Deep entity matching with pre-trained language models. *Proceedings of the VLDB Endowment* **14**(1): 50–60.

Li Y.S., Chi H., Shao X.Y., Qi M.L. & Xu B.G. 2020b. A novel random forest approach for imbalance problem in crime linkage. *Knowledge-Based Systems* **195**: 105738.

Lin Y. 2013. A comparison study on natural and head/tail breaks involving digital elevation models. Bachelor Thesis. University of Gävle.

Liu X., Li X., Fiumara G. & De Meo P. 2023. Link prediction approach combined graph neural network with capsule network. *Expert Systems with Applications* **212**: 118737.

Macqueen J. 1967. Some methods for classification and analysis of multivariate observation. *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability,* pp. 281–297.
Melnykov V. & Zhu X. 2019. An extension of the K-means algorithm to clustering skewed data. *Computational Statistics* **34**(1): 373–394.
Mudgal S., Li H., Rekatsinas T., Doan A., Park Y., Krishnan G., Deep R., Arcaute R. & Raghavendra V. 2018. Deep learning for entity matching: A design space exploration. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 19–34.
Nentwig M., Hartung M., Ngonga Ngomo A.C. & Rahm E. 2017. A survey of current Link Discovery frameworks. *Semantic Web* **8**(3): 419–436.
Newcombe H.B., Kennedy J.M., Axford S.J. & James A.P. 1959. Automatic linkage of vital records. *Science* **130**(3381): 954–959.
Nigro L. & Cicirelli F. 2023. Improving K-means by an Agglomerative Method and Density Peaks, pp. 343–359.
Rochner P. & Rothlauf F. 2024. Using machine learning to link electronic health records in cancer registries: On the tradeoff between linkage quality and manual effort. *International Journal of Medical Informatics* **185**: 105387.
Saleh M. 2024. Evaluation of Jenks natural breaks clustering algorithm for changepoint identification in streaming sensor data. *IEEE Sensors Letters* **8**(10): 1–4.
Salimi Y., Domingo-Fernández D., Hofmann-Apitius M. & Birkenbihl C. 2024. Data-driven thresholding statistically biases ATN profiling across cohort datasets. *The Journal of Prevention of Alzheimer's Disease* **11**(1): 185–195.
Sturges A.H. 1926. The choice of a class interval. *Journal of the American Statistical Association* **21**(153): 65–66.
Tran K.N., Vatsalan D. & Christen P. 2013. GeCo: An online personal data generator and corruptor. *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management,* pp.2473–2476.
Treeratpituk P. & Giles C.L. 2009. Disambiguating authors in academic publications using random forests. *Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital libraries*, pp. 39–48.
Varma S., Sameer N. & Chowdary C.R. 2019. ReLiC: Entity profiling using random forest and trustworthiness of a source. *Sādhanā* **44**(9): 200.
Winkler W.E. 1993. Improved decision rules in the Fellegi-Sunter model of record linkage. *Proceedings of the Section on Survey Research Methods,* pp. 274–279.
Wu R., Chaba S., Sawlani S., Chu X., Thirumuruganathan S. 2020. ZeroER: Entity resolution using zero labelled examples. *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data,* pp. 1149–1164.
Xu X. & Chan W.K.V. 2020. A stable algorithm based on pareto optimal and maximum difference for record matching with errors. *Proceedings of the 6th International Conference on Big Data and Information Analytics,* pp. 180-185.
Zhu Y., Matsuyama Y., Ohashi Y. & Setoguchi S. 2015. When to conduct probabilistic linkage vs. deterministic linkage? A simulation study. *Journal of Biomedical Informatics* **56**: 80–86.
Zhang H. 2004. The optimality of naive Bayes. *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference,* pp. 562–567.

Appendix

Restaurant Cleaning Rules

Rule	**Example**
Field: name	
Remove dots "."	s.p.q.r → s p q r
Remove apostrophe "'"	joe's → joes
Remove dash "-"	hotel bel-air → hotel bel air
Change abbreviation "&" to full form "and"	gotham bar & grill → gotham bar and grill
Remove prefix "restaurant"	restaurant katsu → katsu
Remove additional information within a bracket "(…)"	le chardony (los angeles) → le chardony
Remove article "the" which was placed at the end of name and place it at front	palm the → the palm
Field: addr	
Remove additional information that start with a preposition "between/near/off/at/in/just"	2450 broadway between 90th and 91st sts. → 2450 broadway
Change the spell out ordinal number to a digit format	seventh → 7th

Field: city	
Remove prefix "st."	st. hermosa beach → hermosa beach
Remove prefix "w."	w. hollywood → hollywood
Change any short form to full form	boyle hts. → boyle heights
Change "la" and "west la" to "los angelas"	la → los angelas
Field: phone number	
Separate into three new fields. phn_1 consist of the first three digits. phn_2 consist of the middle three digits. phn_3 consist of the last four digits	310/246-1501 → phn_1: 310, phn_2: 246, phn_3: 1501
Lookup Dive! restaurant phone number	310/788- → phn_1: 310, phn_2: 788, phn_3: 3483
Field: type	
Remove additional phone number information	and 212/614-9345 asian → Asian
Remove additional information within a bracket "(…)"	american (new) → American
Remove additional type	cajun/creole → cajun
Change the same meaning type	coffee bar → coffee shops coffee houses → coffee shops
Remove any prefix	only in las vegas → las vegas
Lookup restaurant information with "NA" type	NA → russian

Mathematics Computer Graphics Programme
Faculty of Science and Technology
Universiti Malaysia Sabah
88400 Kota Kinabalu
Sabah, MALAYSIA
*E-mail: zatur@ums.edu.my**

Department of Mathematical Sciences
Faculty of Science and Technology
Universiti Kebangsaan Malaysia
43600 UKM Bangi
Selangor, MALAYSIA
E-mail: zamira@ukm.edu.my, aftar@ukm.edu.my, tqah@ukm.edu.my

Received: 20 June 2025
Accepted: 1 April 2026

*Corresponding author