IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 1, February 2026, pp. 725∼743
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i1.pp725-743

Explainable deep learning for scalable record linkage: a TabNet-based framework for structured data integration

Fatima Zahrae Saber[1], Ali Choukri[1], Mohamed Amnai[1], Abderrahim Waga[2]
[1]Department of Computer Science, Faculty of Science, Ibn Tofail University, Kenitra, Morocco
[2]School of Digital Engineering and Artificial Intelligence, Euromed University of Fes, Fez, Morocco

Article Info

Article history:

Received Apr 30, 2025
Revised Oct 30, 2025
Accepted Nov 8, 2025

Keywords:

Big data
Data quality
Deep neural networks
Record linkage
TabNet

ABSTRACT

Record linkage is considered a fundamental process for ensuring data quality and reliability, with critical applications in domains such as healthcare, finance, and commerce. A machine learning-based approach for optimizing record linkage in structured datasets is presented in this paper. By integrating hybrid blocking methods (combining standard blocking and sorted neighborhood approaches) with advanced similarity measures, computational overhead is significantly reduced while high accuracy is maintained. The performance of TabNet, a deep learning model designed for tabular data, is compared with traditional deep neural networks (DNNs) in the classification phase. Experimental results on a synthetic dataset of 5,000 records demonstrate that comparable precision and recall are achieved by TabNet to DNNs while execution time is reduced by over 79%. The scalability and efficiency of the proposed method are highlighted by these findings, making it well-suited for large-scale data management tasks. Practical and computationally efficient solutions for record linkage in the era of big data are contributed to by this work.

Corresponding Author:

Fatima Zahrae Saber
Department of Computer Science, Faculty of Science, Ibn Tofail University
Kenitra, Morocco
Email: fatimazahrae.saber@uit.ac.ma

1. INTRODUCTION

Data plays a crucial role in many aspects of daily life. Ensuring high quality data often involves the use of record linkage techniques, which aim to identify and remove duplicate entries referring to the same entity as shown in Figure 1. This process contributes to improved data integrity by reducing redundancy and minimizing errors. However, as databases increase in size and complexity, record linkage becomes increasingly challenging. Traditional methods, such as probabilistic record linkage [1], tend to be time consuming and resource intensive. In the context of big data [2], new challenges arise, including high processing demands, increased hardware costs, and difficulties in accurately determining whether records truly match. The record linkage process can be divided into four main steps [3]: data preprocessing, indexing, comparison, and classification of records. In the first step, tasks such as standardizing and normalizing data are performed to create a uniform database. The second step involves building an index of record pairs that may match, which helps reduce the time required for comparison. Only records within the same group are compared. For large databases, different indexing methods are employed, such as locality sensitive hashing (LSH) and sorted block indexing [4], each with its advantages and disadvantages. In the third step, similarity scores are calculated between the values of each record pair, resulting in scores for all pairs. The final step involves classification, where the record pairs are labeled as matching or not matching based on the calculated scores.

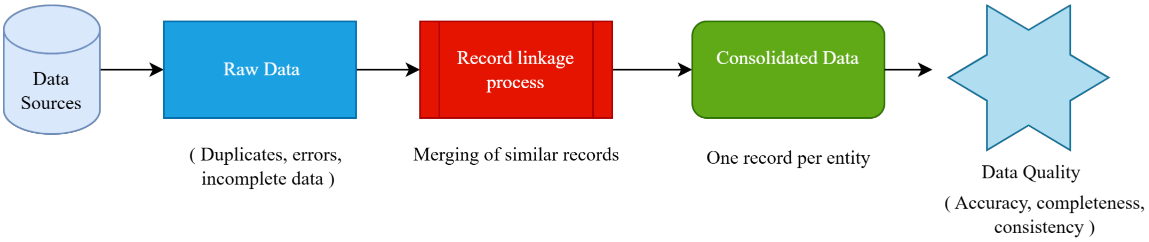

Data Sources
Raw Data
(Duplicates, errors, incomplete data)
Record linkage process
Merging of similar records
Consolidated Data
One record per entity
Data Quality
(Accuracy, completeness, consistency)

Figure 1. Data consolidation process

Some methods for the pair classification phase utilize machine learning algorithms, such as support vector machines (SVM) and XGBoost [5], while others are supported by deep neural networks (DNNs) [6]. DNNs have been recognized as powerful tools in this domain due to their ability to learn complex patterns from large amounts of structured data. To better understand the current challenges and advancements in the field, previous studies that have addressed the record linkage issue are reviewed. Record linkage, also known as entity resolution, is regarded as a critical task for the integration and deduplication of large datasets across diverse domains. Over the years, various methodologies have been proposed to address the challenges of scalability, accuracy, and privacy in record linkage processes.

This paper proposes a novel hybrid blocking technique integrated with TabNet, a deep learning model specifically designed for tabular data [7]. Our approach optimizes the record linkage process by reducing computational overhead, improving execution time, and maintaining high accuracy. Through experiments conducted on synthetic datasets, we demonstrate the effectiveness and scalability of our method, highlighting its potential for large scale applications in data management.

2. RELATED WORK

Record linkage, or entity resolution, is a critical task in data integration. The goal is identifying and merging records that refer to the same entity across different datasets. Over the years, various approaches have been developed to address challenges related to scalability, accuracy, and privacy in record linkage processes.

2.1. Traditional methods

Traditional probabilistic models, such as the Fellegi-Sunter model, have long dominated record linkage [8], focusing on probabilistic scoring to match records based on similarity thresholds. Recent advancements have incorporated ensemble methods and machine learning algorithms, as demonstrated in probabilistic record linkage for families (PRLF), an open source Python based tool. PRLF employs generalized linear models and machine learning to improve accuracy under challenging conditions, such as data degradation and missing fields, offering robust performance across synthetic and real world datasets.

2.2. Machine learning approaches

Heydari *et al.* [9] propose a distributed record linkage method applied to healthcare data using Apache Spark and its MLlib library. Their approach utilizes machine learning algorithms, such as regression and SVM, to match records based on preprocessed features like names, dates of birth, and zip codes. This study is notable for its use of stratified sampling to address the common issue of imbalanced datasets in record linkage, as well as its rigorous model validation, ensuring robust performance. The results demonstrate remarkable accuracy (up to 96.71% for regression), highlighting the scalability offered by Spark in handling massive data environments. This method showcases the effectiveness of a distributed approach in addressing challenges related to scalability and accuracy, although it focuses primarily on healthcare specific data.

2.3. Deep learning based methods

An innovative solution [10] introduces a scalable deep learning-based approach designed for big data scenarios. This method builds an artificial neural network (ANN), specifically a Siamese network, to efficiently encode records for faster similarity computations. By leveraging the cosine similarity metric, the network classifies record pairs as either matched or unmatched. The use of Apache Spark further enhances the scalability of this method, enabling parallel processing of large datasets and reducing computational overhead. This integration of deep learning and distributed computing makes it particularly suitable for handling large-scale data integration tasks.

Application of deep learning on record linkage is one of the major research area that seeks to address scalability and inflexibility problems of conventional rule-based approaches. Yulianton and Santi [11] present a deep learning approach for e-commerce product matching based on Sentence-BERT. Using lightweight transformer embeddings and cosine similarity with a fixed threshold, their method effectively captures semantic similarities between heterogeneous product titles. Evaluated on the Pricerunner dataset, the approach achieves high accuracy and perfect precision, demonstrating that efficient SBERT-based models are well suited for large-scale product matching tasks.

In the meantime, newer models like transformers [12] hold considerable promise for matching. In the related map matching field, a transformer model achieved F1-scores of over 96%, setting very high levels of efficiency for sequence matching problems. This piece confirms the line of deep learning models to address complex contextual information, and it also highlights the need for solutions that are still effective and powerful in addressing real-world data problems. Table 1 presents a comparative study of record linkage methods using deep learning for tabular data.

Table 1. Comparative table of existing deep learning approaches for record linkage

Category	Method/study	Approach/technique	Key features/strengths	Mentioned performance	Reference
Deep learning	Sentence-BERT (MiniLM)	Transformer-based sentence embeddings with cosine similarity and threshold-based matching	Lightweight transformer enabling semantic product matching with high precision and low computational cost; scalable to large e-commerce datasets	Accuracy: 98.10%, Precision: 100%, Recall: 91.84%, F1-score: 95.74%	Yulianton and Santi [11]
Deep learning	Neural ER (Tuple embeddings)	DNNs for learning distributed representations of structured entity attributes	Effective for complex entity matching tasks on heterogeneous structured data, including medical and product datasets	F1-score up to 94%	Peeters and Bizer [13]
Deep learning	Transformer (Seq2Seq)	Uses transfer learning with a transformer architecture.	Shows high potential for sequence matching tasks, though related to "map matching".	F1-score: >96% (at segment level)	Jin *et al.* [12]

2.4. Privacy-preserving record linkage based methods

The method of Wang *et al.* [14] seeks to enhance bloom filter-based privacy-preserving record linkage (PPRL). Their "(Hash)-A" hashing approach tackles information loss by coding q-gram frequency to more effectively differentiate between records and thus increase matching accuracy. To protect privacy, the "utility-optimized bloom filter" (UBF) approach utilizes user-level differential privacy (ULDP) to subject only a subset of bits recognized as sensitive to intense perturbation. This selective protection provides an improved trade-off between utility (linkage accuracy) and privacy than current methods. Ranbaduge *et al.* [15] presents the inaugural multi-party PPRL protocol to combine deep learning into a federated learning paradigm. The database owners initially encode their records into bloom filters, to which differential privacy noise is injected to provide provable protection for privacy. Local deep learning models are then trained separately by each party on feature vectors (similarity/distance scores) derived from such noisy bloom filters. Lastly, the local models are submitted to a secure aggregator that ensembles them into a global model, which a linkage unit uses to classify unlabeled data.

Table 2 summarizes the categories of record linkage methods, with the advantages and disadvantages of each. Our proposed solution optimizes record linkage through the combination of a hybrid blocking strategy and a TabNet classifier, a deep learning model specifically designed for tabular data. Through this combination, a new computation-accuracy tradeoff for medium to large datasets is introduced. Firstly, the number of pairs that need to be compared is significantly reduced by the hybrid blocking technique, lowering the workload computation while maintaining high recall of potential matches. This is followed by several critical advantages of the TabNet model: extremely accurate classification is possible, enhanced interpretability is facilitated through its attention mechanism, and most significantly, it is extremely efficient, with execution time reducing over 79% compared to a standard DNN.

Table 2. Comparative table of record linkage approaches

Category	Method/study	Approach/technique	Key features/strengths	Mentioned performance	Reference
PPRL	Enhanced bloom filter PPRL	"(Hash)-A" hashing with q-gram frequency and UBF with differential privacy.	Selective protection for a better accuracy-privacy trade-off.	Improved trade-off	Wang *et al.* [14]
Machine learning	Distributed approach	Regression and SVM on Apache Spark (MLlib).	High scalability; handles imbalanced data.	Up to 96.71% accuracy	Heydari *et al.* [9]
Probabilistic	PRLF	Ensemble methods, generalized linear models, and machine learning.	Open-source Python tool; robust against data degradation and missing fields.	Robust performance	Prindle *et al.* [8]
PPRL	Federated PPRL with deep learning	Multi-party protocol using noisy bloom filters to train and aggregate local models.	First protocol combining DL and federated learning for PPRL.	Robust performance	Ranbaduge *et al.* [15]
Deep learning	Transformer model	Transformer (Seq2Seq) with transfer learning.	Very promising; high efficiency for sequences.	F1-scores of over 96%	Jin *et al.* [12]
Deep learning	Siamese network on Spark	ANN (Siamese network) to encode records.	Scalable, designed for big data.	Efficient for reducing computation time.	Wolcott *et al.* [10]

3. METHOD

In this paradigm, record linkage becomes a supervised learning issue. It starts with leveraging the freely extensible biomedical record linkage 2 (FEBRL 2) dataset that comprises 5,000 synthetic records with pre-determined duplicates, thereby serving as training and validation labeled data. For each candidate record pair, a vector of similarities is calculated by applying stable measures such as the Jaro-Winkler (JW) and Levenshtein distances. This feature vector is then used to train a deep learning-based classifier, TabNet, in which duplicate pairs (matches) are learned to be distinguished from non-duplicate pairs. New, unseen record pairs can then be predicted to be new or not by the learned model. Several inherent challenges of this supervised learning task are directly addressed by our methodology:

i) Noisy data: real-world data is also infamous for having entry errors, format variations, and missing values. Our preprocessing stage of uppercase conversion, removal of irrelevant symbols, and numeric field cleaning guarantees correct results. Furthermore, the JW distance was chosen especially so that common typographical errors could be tolerated, making the method robust to noise.

ii) Class imbalance: class imbalance is famously a critical issue in record linkage, as was the case in our earlier work where this problem caused very low precision. Our approach utilizes deep learning models that perform well on imbalanced tabular data, as is evident from the high precision and recall values achieved.

iii) Domain adaptation: the fact that the synthetic data might not capture the complexity of real data is cited as a primary limitation. Therefore, as a key future exercise, the generalizability of our model will be evaluated on a large, real-world dataset—the North Carolina voter registration (NCVR)—so that it can be validated to generalize well to a number of production environments.

Artificial intelligence (AI) models, and deep learning techniques in particular, are better equipped to handle the ambiguity inherent in real-world data, thereby outperforming their classical rule-based counterparts. While classical rule-based techniques are traditionally described as stiff, greater flexibility and better performance in handling noisy or missing data are offered by our AI technique. Instead of being fixed and binary rules being applied, the model is trained from a rich similarity vector. The similarity level is quantified in terms of similarity scores, which are calculated based on JW and Levenshtein distances, such that variations, typos, and other forms of error can be managed by the model. The complex interaction among the diverse scores is then set up by the TabNet model so that it can make a probabilistic decision—a much more sophisticated task than what could possibly be accomplished by a set of rules. This ability for evidence to be dynamically weighted and for the most relevant features for any prediction to be established, as shown in our analysis of interpretability, is why uncertainty is best dealt with by AI.

As previously mentioned, the process is composed of four main steps. First, the data is preprocessed to clean and normalize it [3]. Next, a hybrid blocking method is employed to reduce the number of comparisons by dividing the data into smaller, more manageable blocks. Two techniques, sorted neighborhood and standard blocking, are used to create an index of candidate record pairs. These pairs are subsequently compared using similarity measures such as the Levenshtein distance and the JW distance. The resulting similarity scores are then fed into a classification model. To evaluate performance in terms of execution time and accuracy, TabNet and a DNN model were utilized, aiming to determine the best trade off between speed and precision [7], [16]. As shown in Figure 2.

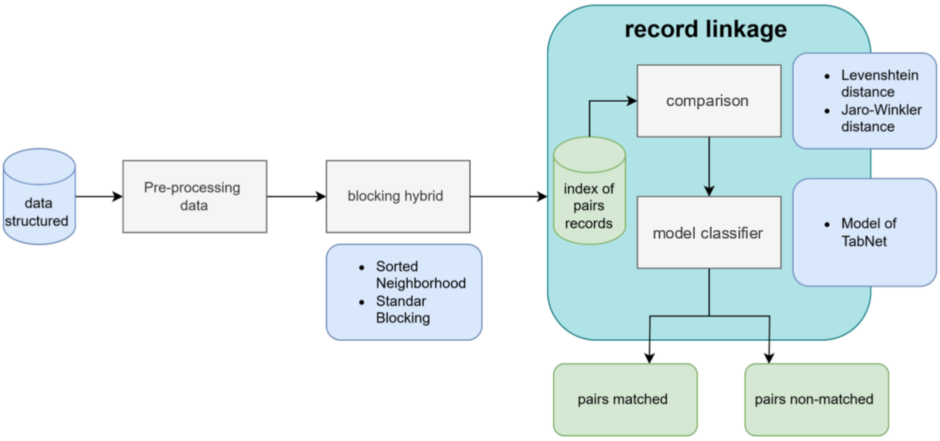


Figure 2. Proposed record linkage process

3.1. Training and validation dataset

The training and validation dataset used for this study is the FEBRL 2, which consists of fictitious records simulating personal information typically found in structured databases. It contains 5,000 rows, including 4,000 original records and 1,000 duplicate records. The dataset comprises six columns, each representing a specific attribute related to individuals, such as first name, last name, address, and other personal details. This dataset is employed to test and validate the proposed record linkage method by replicating real world conditions encountered in large scale databases.

The main columns include both textual and numerical information, as illustrated in Table 1. These columns represent the types of data commonly found in administrative or commercial databases and present typical challenges such as input errors, missing data, and format inconsistencies. In this context, data preprocessing was essential to normalize certain columns and address inconsistencies, as detailed in the next section. This step is critical for improving the quality of record matches. Table 3 provides a detailed description of each column in the dataset, along with concrete examples and remarks regarding the specific characteristics of each field.

Table 3. Description and specific features of the dataset used

Column name	Description	Data type	Example
given_name	First name	Text	SARAH
surname	Surname	Text	BRUHN
address_1	First line of address	Text	FORBES STREET
state	State	Text	VIC
date_of_birth	Date of birth (format YYYYMMDD)	Numeric	19300213
soc_sec_id	Unique social security number	Numeric	7535316

3.2. Experimental dataset

Three datasets were generated from the training dataset to experiment with and test the proposed method, as well as to evaluate the performance of the models and the execution time of each prediction. The execution time is considered an important criterion in this study, as the objective is to identify a method that reduces the time required for record comparison and duplicate prediction. Larger datasets were created to assess the models' performance at a larger scale. As shown in Table 4, the first dataset consists of 13,000 records, with 10,000 original records and 3,000 duplicates. The second dataset contains 16,000 records, including 12,000 original records and 4,000 duplicates. Finally, the third dataset includes 21,000 records, comprising 16,000 original records and 5,000 duplicates.

Table 4. Overview of datasets used for experimental model

Column name	Dataset 1	Dataset 2	Dataset 3
Total records	13,000	16,000	21,000
Original records	10,000	12,000	16,000
Duplicate records	3,000	4,000	5,000

3.3. Data preprocessing

To enhance data quality and facilitate comparisons during the record linkage process, several transformations were applied. First, columns containing textual data such as first names, surnames, and addresses were converted to uppercase to ensure consistency in information representation, regardless of variations in case. Next, irrelevant symbols and characters, particularly in address fields, were removed to refine matches and reduce potential inconsistencies [17]. For numeric fields, particularly zip codes, non-conforming (non-numeric) values were identified and removed to improve the accuracy of comparisons. Additionally, other specific columns underwent tailored cleaning operations, such as the standardization of abbreviations and the correction of typographical errors. These preprocessing steps are crucial for ensuring reliable results during the matching phase [18].

3.4. Indexing

Indexing is a critical step in the record matching process, designed to reduce the number of record pairs to be compared while maintaining a high level of accuracy in match detection [4]. Given the size of the dataset used in this study (5,000 records), the total number of potential comparisons without indexing would be extremely large, potentially reaching several million pairs. To address this challenge, a hybrid blocking approach was adopted, combining two methods: standard blocking and the sorted neighborhood method. The efficiency of the process is significantly enhanced by this combination, reducing the number of pairs to be compared while effectively identifying relevant matches.

3.4.1. Standard blocking

The first method employed is standard blocking [19], which involves dividing records into blocks based on one or more columns. For this study, records were blocked using the state column. This approach results in records being grouped by zip code, thereby restricting comparisons to within each block. While this technique effectively reduces the number of comparisons, limitations arise when dealing with missing or incorrect zip code values.

3.4.2. Sorted neighborhood

To address these limitations, standard blocking was combined with the sorted neighborhood method. This technique involves sorting records based on a sort key and then comparing each record with its neighbors within a fixed size window [20]. By using the surname as the sort key, this method captures matches that may not be grouped together in standard blocking due to minor variations in zip codes. The sliding window approach allows comparisons to be made only between neighboring records, significantly reducing the number of pairs to be compared. Figure 3 illustrates this process, where pairs of records (Record_a and Record_b) are compared after sorting. The lines represent potential matches between neighboring records, showing how the sorted neighborhood method limits the comparisons while capturing relevant matches.

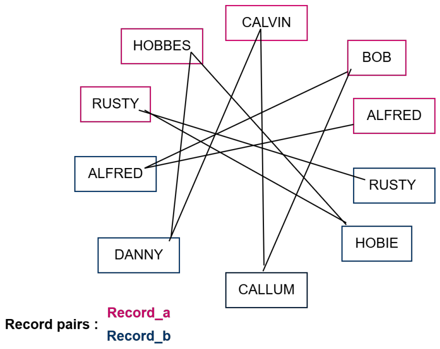


Figure 3. Sorted neighborhood algorithm to index record pairs

3.5. Hybrid blocking

The combination of standard state-based blocking and the sorted neighborhood algorithm forms a robust hybrid blocking approach. First, standard state based blocking reduces the number of pairs to be compared by excluding records that are geographically too distant. Second, the sorted neighborhood algorithm refines this process by performing comparisons between records sorted based on their surname, thereby capturing matches that might have been missed by standard blocking alone [21]–[23]. As illustrated in Figure 4, these two methods work together to improve the efficiency and effectiveness of record matching.

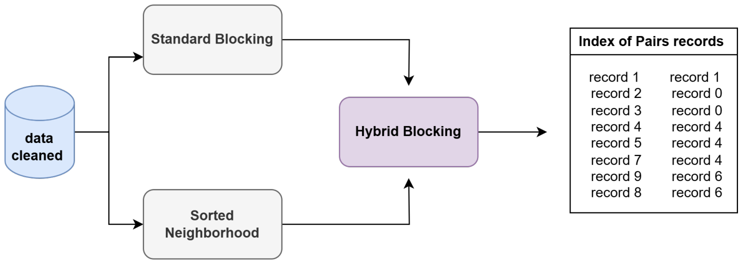


Figure 4. Hybrid blocking method was used

The hybrid approach offers several advantages. The complexity of comparisons is significantly reduced while maintaining a high degree of accuracy in matching records. It effectively handles minor variations in textual data, input errors, and missing values in specific columns. Following this step, an index of over 2.8 million record pairs is generated, which will be compared and classified as either matched or unmatched pairs. Table 5 presents the number of record pairs for each dataset, highlighting that as the number of records in a dataset increases, the number of record pairs for comparison also grows.

Table 5. Number of record pairs for each dataset

Dataset	Train/validation	Exp. dataset 1	Exp. dataset 2	Exp. dataset 3
Total records	5,000	13,000	16,000	21,000
Pairwise indexes	28,700	2,900,000	4,200,000	7,300,000

This comparison Table 6 different blocking strategies in terms of how well they perform in reducing the number of candidate pairs for record linkage. While a full index results in nearly 12.5 million pairs (0% reduction), the best performing method is the proposed hybrid blocking approach. It greatly minimizes the number of comparisons to just 28,702 pairs with an accomplishment of 99.77% reduction ratio (RR). This emphasizes the central contribution of the hybrid strategy to improving the computational efficiency of the record linkage pipeline.

Table 6. Comparison of blocking strategies by number of candidate pairs and RR

Method	Number of pairs	RR (%)
Full index	12,497,500	0.00
Blocking (state)	2,768,103	77.85
SortedNeighbour (surname)	75,034	99.40
Hybrid blocking	28,702	99.77

3.6. Comparison phase

In the comparison phase, similarity measures are applied to assess the correspondence between record pairs. Two well established methods, JW and Levenshtein distances, have been selected for this task. Each comparison produces a similarity score ranging from 0 to 1, reflecting the degree of correspondence between field values. These scores are then aggregated into a similarity vector, which summarizes the overall similarity between the two records. This similarity vector serves as the foundation for the subsequent classification phase, where it is determined whether the records represent the same entity.

3.6.1. Jaro-Winkler distance

The JW distance metric is especially effective for short strings, such as names. In this study, it was applied to several fields, including given_name, surname, address_1, and state. By taking into account both character matches and their order, the JW distance is sensitive to common typographical errors, making it particularly suitable for record matching tasks [24]–[26]. The JW improves the Jaro distance by adding a prefix scale:

$$JW = J + (l \times p \times (1 - J)) \tag{1}$$

In this equation, l is the length of the common prefix (up to 4 characters), and p is a scaling factor, usually 0.1. The adjustment favors strings that match from the beginning.

3.6.2. Levenshtein distance

The Levenshtein distance metric calculates the minimum number of operations (insertion, deletion, or substitution) required to transform one string into another. In the context of this study, Levenshtein distance was applied to numerical fields such as date_of_birth and soc_sec_id. This approach effectively quantifies the differences between records, even when there are variations in data entry, such as errors in date formatting or incorrect postal codes [27]. The Levenshtein distance, also known as the edit distance, measures the minimum number of single-character operations required to transform one string into another. The operations permitted include insertion, deletion, and substitution.

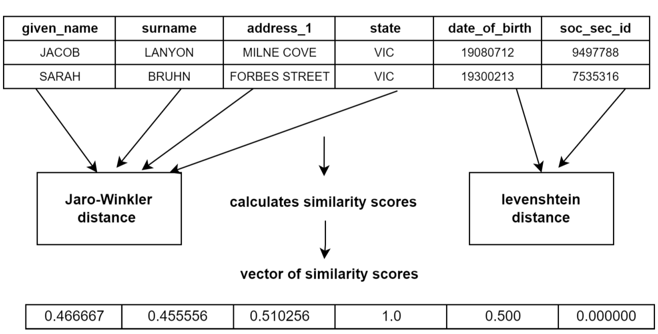
The Levenshtein distance measures the minimum number of single character edits required to change one string into another. The permitted operations include insertion, deletion, and substitution. The recursive formula for computing the Levenshtein distance $d(a, b)$ between two strings a and b is defined as:

$$d(a,b) = \begin{cases} \max(\text{len}(a), \text{len}(b)) & \text{if } \min(\text{len}(a), \text{len}(b)) = 0 \\ d(a{-}1, b{-}1) & \text{if } a = b \\ 1 + \min \begin{cases} d(a{-}1, b) \\ d(a, b{-}1) \\ d(a{-}1, b{-}1) \end{cases} & \text{otherwise} \end{cases} \tag{2}$$

- a and b are the two strings being compared.
- $d(a, b)$ is the minimum number of edit operations needed to convert string a into string b.
- The allowed operations are: i) insertion of a single character, ii) deletion of a single character, iii) substitution of one character for another
- len(a) and len(b) denote the lengths of the strings a and b, respectively.

This algorithm is widely used in approximate string matching and natural language processing tasks, as it provides a quantifiable measure of similarity between two sequences based on their structural differences.

In Figure 5, each row represents a pair of records, and each column shows the score for the corresponding attribute. For instance, for the first record pair, the scores are 0.466667 for the first name (given_name_score), 0.455556 for the surname (surname_score), and so on. These individual attribute scores are combined to calculate an overall similarity score for each pair of records. The similarity measures are selectively applied to the record pairs generated in the previous step, which uses a hybrid blocking system. This technique reduces the number of comparisons required, optimizing the process while maintaining a high precision rate. As a result, similarity vectors are generated, where each pair of records is associated with a similarity score for each attribute.



given_name	surname	address_1	state	date_of_birth	soc_sec_id
JACOB	LANYON	MILNE COVE	VIC	19080712	9497788
SARAH	BRUHN	FORBES STREET	VIC	19300213	7535316

0.466667	0.455556	0.510256	1.0	0.500	0.000000

Figure 5. Comparison of record pair similarities using JW and Levenshtein distances

High scores indicate a strong similarity between the attribute values, suggesting a probable match between the records. This detailed scoring system offers greater flexibility in the final classification step. While traditional record linkage methods typically apply a global similarity threshold to determine matches, our method classifies record pairs as either matching or non-matching using TabNet, a deep learning model specifically designed for tabular data. This approach was selected to improve both accuracy and execution time.

3.7. Classification models

For our record linkage experiment, a pragmatic hyperparameter search strategy was utilized for two classification models: TabNet and DNN. Rather than exhaustive search, we took established parameters. We trained the TabNet model at learning rate 0.02, max_epochs of 6, and patience of 5 for early stopping. On the other hand, the DNN was regularized with binary cross entropy loss and employed dropout and early stopping as methods of regularization. The choice of model was based on a balance between performance as indicated by accuracy and execution time, and computational efficiency. The goal was to achieve the model offering the best balance for deployability at scale.

3.7.1. TabNet

TabNet is a deep learning model uniquely designed for the effective handling of tabular data. Unlike traditional neural network architectures [28], [29], TabNet employs an innovative approach that integrates attention mechanisms with a hierarchical structure to identify and extract relevant features from the data, see Figure 6. This model has demonstrated significant success in tasks involving structured datasets, due to its ability to focus on the most informative parts of the data while maintaining interpretability.

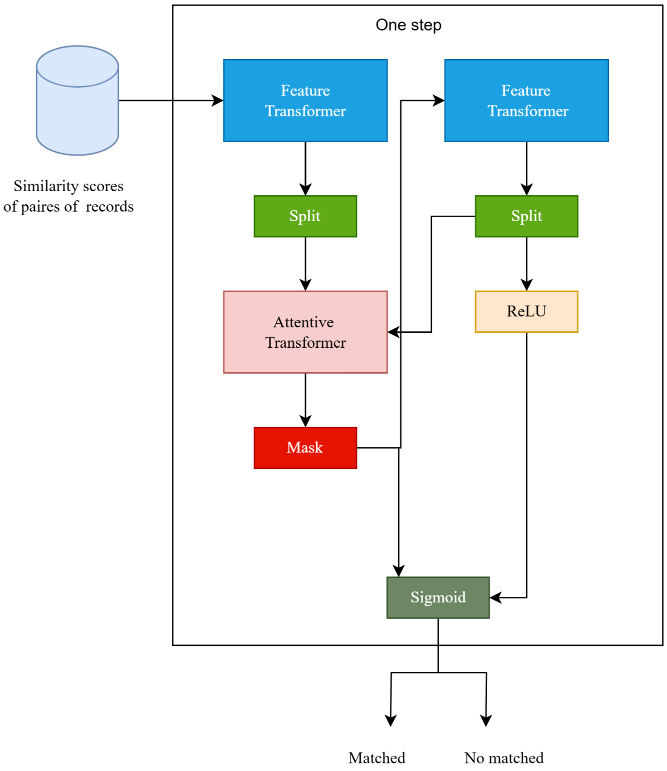


Figure 6. The TabNet model for record pair classification

The TabNet model begins with the feature transformer, which transforms the input variables into richer representations suitable for prediction tasks. This component consists of four layers: fully connected layers (dense layers) that integrate the variables, batch normalization to stabilize the learning process, and specific activation functions such as gated linear units (GLU) that dynamically select relevant information.

The primary purpose of the feature transformer is to extract complex, non-linear representations of the data, capture interactions between variables, and prepare these representations for the next phase the attentive transformer module.

Before progressing to the attentive transformer, the data is divided using a split mechanism into two parts. The first part produces a partial prediction result, while the second part is forwarded to the attentive transformer, which focuses on selecting relevant features. The attentive transformer leverages an attention mechanism to identify and emphasize the most important columns at each stage, capturing intricate relationships among them. This approach enables TabNet to dynamically select relevant combinations of columns, improving its efficiency and flexibility in prediction tasks involving structured data. Once the relationships between columns have been identified, the model dynamically selects the relevant columns at each step using a mask. This process is iterated 10 times to generate the final prediction for each record pair, determining whether they are a match or not. Due to its architecture, TabNet has proven to be an effective tool for classification tasks, particularly in the context of structured data.

3.7.2. Neural networks deep neural networks

Deep learning models, particularly DNNs, have become increasingly utilized for solving record linkage problems, including tasks such as record pair classification, record normalization, and similarity computation between records [6], [30], [31]. The DNN model used for record pair classification consists of three dense layers: an input layer with 256 nodes employing the ReLU activation function, followed by a dropout layer for regularization; a hidden layer with 128 nodes, also utilizing the ReLU activation function and a dropout layer; and an output layer with a single node using a sigmoid activation function for binary classification see Figure 7 (1 for matched records and 0 for unmatched records). Table 7 presents a comparison between the TabNet model and the DNN in terms of architecture, scalability, training time, performance, and other relevant factors. The advantages of the TabNet model over the previously employed DNN are clearly highlighted in the table.

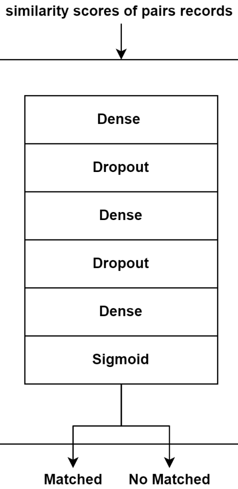


Figure 7. The DNN model for record pair classification

Table 7. Comparison between TabNet and DNNs

Criteria	TabNet	DNN
Data type	Optimized for tabular data	Can process various data types (images and text)
Architecture	Uses attention mechanisms and dynamic masks	Composed of fully connected layers
Activation functions	Sparse activation via attention	Non-linear functions such as ReLU, sigmoid
Interpretability	High due to the attention mechanism	Limited due to complex structure
Overfitting prevention	Integrated regularization techniques	Dropout, early stopping, etc.
Scalability	Efficient on large tabular datasets	Requires more resources for large datasets
Training time	Fast due to feature selection via attention	Can be long with deep architectures
Performance	Performs well on imbalanced tabular data	Requires tuning for optimal performance
Typical applications	Classification and regression on tabular data	Computer vision, NLP, and more

Algorithm 1 outlines the overall process of our record linkage procedure. It begins with preprocessing data for standardization. A hybrid blocking strategy is next used to strategically reduce the search space by selecting only the most likely candidate pairs. For each of these candidate pairs, a similarity vector is calculated with measures such as JW and Levenshtein. Finally, these vectors are used to train our TabNet classifier, which gives us the final output of whether to match the records.

Algorithm 1 Record linkage pipeline with hybrid blocking and TabNet

```
Input: dataset A, dataset B
Output: set of matched pairs $M$
// Step 1: preprocessing
for each record $r$ in dataset A and B do
    Convert text fields to uppercase
    Remove irrelevant symbols
    Clean numeric fields
end for
// Step 2: Candidate pair generation (hybrid blocking)
$P_{\text{standard}} \leftarrow$ StandardBlocking($A, B,$ on='state')
$P_{\text{sorted}} \leftarrow$ SortedNeighborhood($A, B,$ on='surname')
$P_{\text{candidates}} \leftarrow P_{\text{standard}} \cup P_{\text{sorted}}$
// Step 3: Similarity vector calculation
$S \leftarrow$ Initialize empty list for similarity vectors
for each pair $(r_a, r_b)$ in $P_{\text{candidates}}$ do
    $sim\_vector \leftarrow$ CalculateSimilarityScores$(r_a, r_b)$        ▷ Jaro-Winkler, Levenshtein
    Append $sim\_vector$ to $S$
end for
// Step 4: Classification
$TabNet_{\text{model}} \leftarrow$ TrainTabNetModel(labeled_similarity_vectors)
$M \leftarrow$ PredictMatches$(TabNet_{\text{model}}, S)$
return $M$
```

3.7.3. Evaluation metrics

The classification performance of our model is evaluated using four key metrics: accuracy, precision, recall, and F1-score. These measures are calculated based on the elements of the confusion matrix, true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN).

- Accuracy: accuracy represents the proportion of correctly classified pairs, including both duplicates and non-duplicates. The formula for accuracy is:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

  Where TP is number of correctly matched record pairs (i.e., actual duplicates correctly identified). FP is number of non-duplicate pairs incorrectly classified as duplicates. FN is number of duplicate pairs that the model failed to identify. TN is number of correctly identified non-duplicate pairs.

- Precision (positive predictive value - PPV): precision measures the proportion of predicted matches that are actually correct:

$$\text{Precision} = \frac{TP}{TP + FP}$$

- Recall (sensitivity or true positive rate - TPR): recall indicates how many actual duplicates were correctly identified:

$$\text{Recall} = \frac{TP}{TP + FN}$$

– F1-score: the F1-score is the harmonic mean of precision and recall, balancing both FP and FN:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The methodological choices presented above are not only technically sound but are also seen as contributing to the novelty and practical relevance of the proposed approach. The originality of the proposed approach is demonstrated through the integration of hybrid blocking techniques with TabNet, a deep learning model specifically designed for tabular data. Unlike existing methods that are often dependent on complex distributed infrastructures or conventional classifiers with limited scalability, a more efficient and accessible alternative is offered. A significant reduction in candidate record pairs is achieved by combining standard blocking with the sorted neighborhood method, while preserving high matching quality. TabNet is employed to enhance interpretability and to reduce prediction time by over 79%, thus enabling practical deployment in real-time or near real time scenarios. As a result, this approach is particularly well suited for use cases such as healthcare data integration, fraud detection, and customer database deduplication, especially in environments where computational resources are limited.

4. RESULTS AND DISCUSSION

This section presents the results related to the performance and prediction time of each model. The models were initially trained using 80% of the FEBRL 2 dataset, with the remaining 20% reserved for validation. This division ensured proper model training and evaluation. Afterward, the models were tested on larger datasets across three distinct scenarios.

The timely impact of our hybrid blocking mechanism can be considered an ablation study on computational load. As outlined in Table 6, not including the blocking factor (the "full index" scenario) would result in a comparison of nearly 12.5 million candidate pairs. Our hybrid approach drops this significantly to just 28,702, with a 99.77% reduction in computational load. This result is the primary motivation behind adopting the blocking step, as classifying the full set of pairs would be computationally inefficient, thus proving the efficiency of our suggested pipeline.

4.1. Training and validation phase

As shown in Table 8, the models exhibit a notable difference in speed while maintaining high classification performance. On a dataset of 500,000 pairs, the TabNet model processed the data in just 17.23 seconds, significantly faster than the DNN model which required 82.06 seconds, underscoring TabNet's superior efficiency for large-scale or real-time processing. In terms of performance metrics, both models proved to be highly effective, achieving a precision of 0.99, a recall of 0.98, and an F1-score of 0.98. These results suggest that while both models are well-suited for applications where accuracy is paramount, TabNet offers a considerable advantage in execution time.

Table 8. Performance metrics and execution time for TabNet and DNN models

Model	Accuracy	Precision	Recall	F1-score	Time (seconds)
DNN	0.99	0.98	0.98	0.99	82.06
TabNet	0.99	0.98	0.97	0.99	17.23

The side-by-side comparison of the results is shown in Figures 8 and 9 that both the TabNet and DNN models are performing an excellent level on the record linkage task. Even though the DNN shows a slight edge of pure performance with the ideal area under the curve (AUC) score of 1.000 and only 4 missed duplicates, compared to 17 for TabNet (AUC of 0.995), both models performed exceptionally by having zero false positives, which is of critical importance for data integrity. This marginal performance difference must be weighed, however, against TabNet's enormous advances in computational efficiency and inherent interpretability. As a result, TabNet stands as the most pragmatic and well-rounded method for large-scale deployment, offering near-perfect performance while ensuring efficiency and transparency.

While this research demonstrated the effectiveness of our model, testing our framework on the synthetic FEBRL dataset is a principal limitation because such data may fail to capture the complex noise and diverse distributions of real-world data. One principal area of future research is thus to validate our model's generalizability and insensitivity in a naturalistic setting. With this goal in mind, we plan to evaluate and test our TabNet-motivated approach on the vast-scale NCVR dataset. This significant move will subject the actual-world applicability of our solution to testing, checking whether its combination of high performance, efficiency, and interpretability translates effectively to production environments.

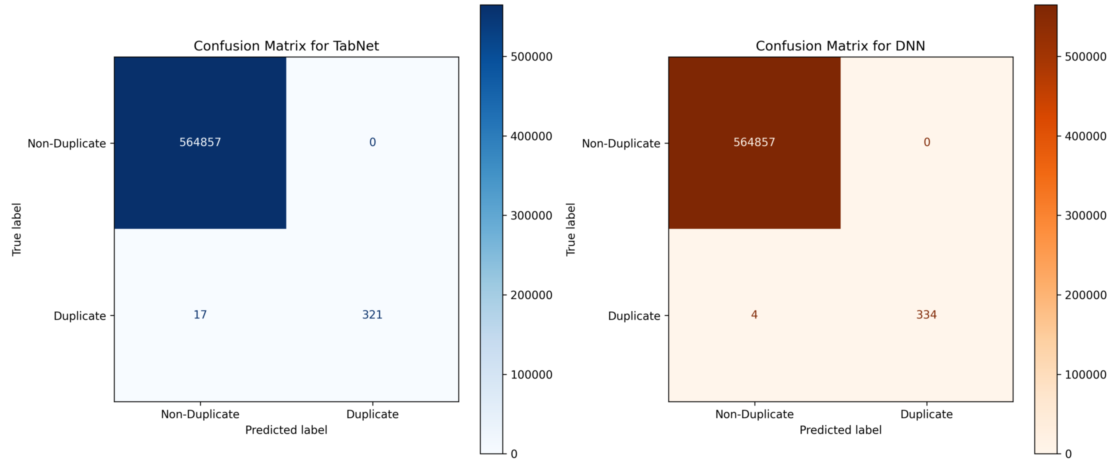


Figure 8. Confusion matrices for TabNet and DNN

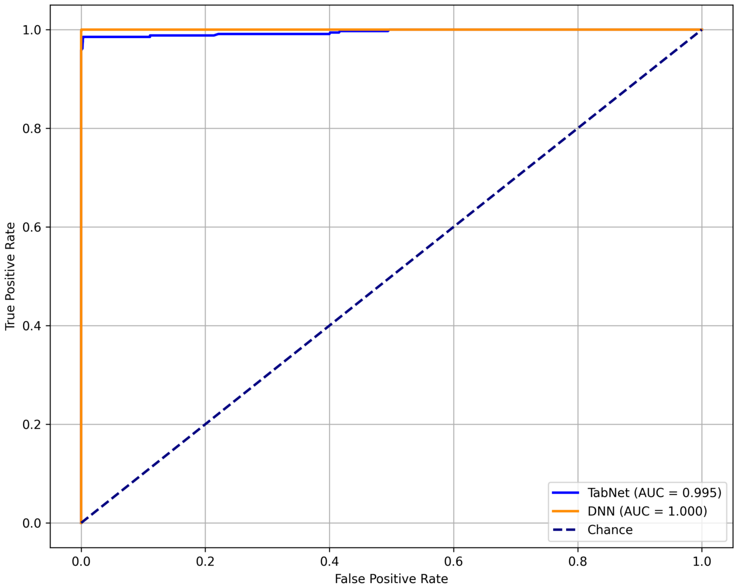


Figure 9. ROC curves for TabNet and DNN

4.2. Experimental phase

To further assess performance of the TabNet and DNN models, three additional tests were conducted using larger datasets than those employed in the initial evaluations. The first dataset (dataset 1) contains 13,000 records, the second dataset (dataset 2) consists of 16,000 records, and the third dataset (dataset 3) includes 21,000 records. These tests were designed to evaluate the models' robustness and scalability when faced with progressively larger datasets.

The evaluation and testing of the TabNet and DNN models yielded generally satisfactory results as shown in Figures 10 to 15, with high precision, recall, and F1-score metrics for both models. Specifically, both models achieved a precision of 0.99, recall of 0.98, and an F1-score of 0.98. A comparative analysis, presented in the figures, highlights the differences in execution time and accuracy between the two models. Notably, TabNet significantly outperforms DNN in terms of computation time, reducing processing time by approximately 79%. For a dataset of 500,000 pairs, TabNet completed the task in 17.23 seconds, compared to 82.06 seconds for DNN. This time advantage is crucial for applications that process large volumes of data, particularly in real time or near real time environments such as epidemiological surveillance systems or commercial database integration platforms.

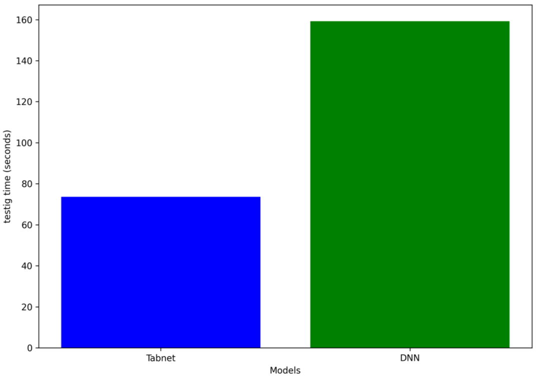


Figure 10. Comparison of execution times for dataset 1

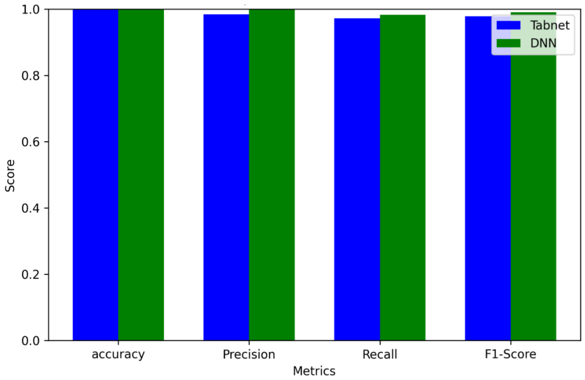


Figure 11. Testing phase metrics for dataset 1

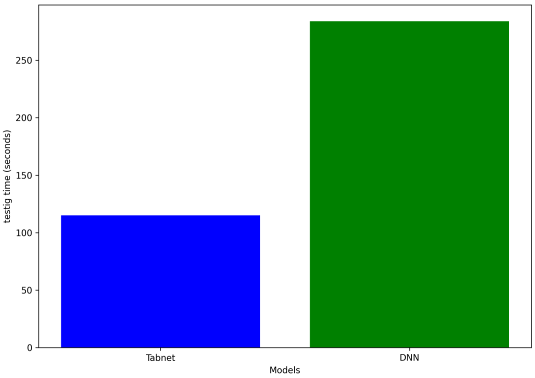


Figure 12. Comparison of execution times for dataset 2

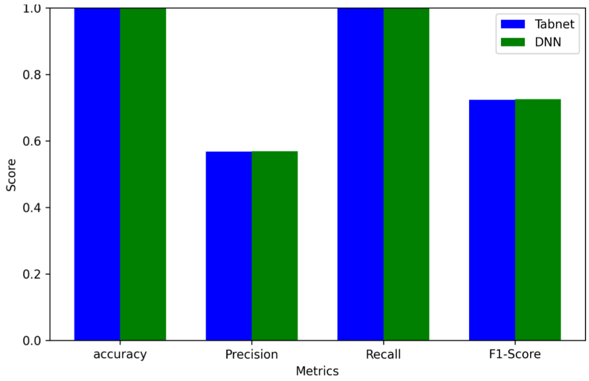


Figure 13. Testing phase metrics for dataset 2

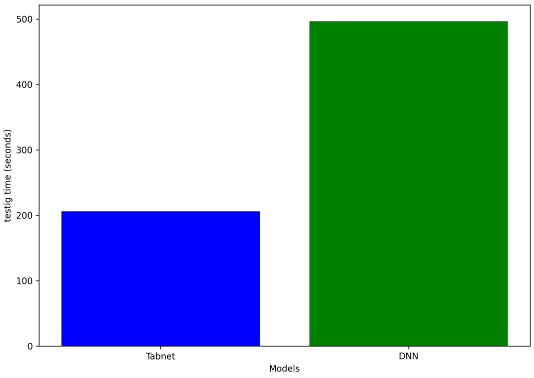


Figure 14. Comparison of execution times for dataset 3

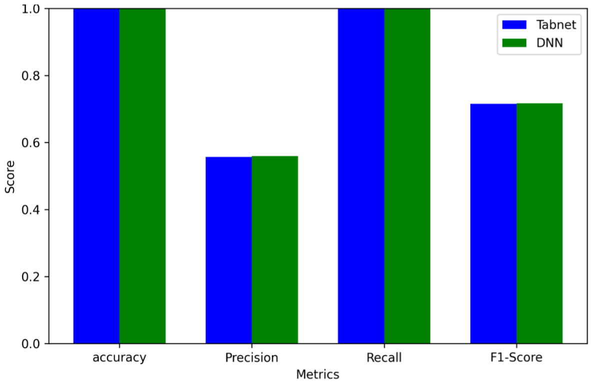


Figure 15. Testing phase metrics for dataset 3

In terms of accuracy and recall, TabNet's performance is comparable to that of DNN, with a slight decrease in accuracy as data size increases. This suggests that further fine tuning of hyperparameters could improve model stability when working with larger datasets. Additionally, an error analysis reveals that both models are sensitive to noisy data and textual inconsistencies, highlighting the importance of rigorous data preprocessing to enhance input data quality and improve classification performance. The effectiveness of the similarity measures used, specifically the JW distance for textual data and Levenshtein distance for numerical data, played a key role in achieving a high rate of exact matches. These measures effectively address typographical errors and variations in data formats, ensuring accurate pair comparisons.

When compared to traditional probabilistic methods, such as the Fellegi-Sunter model, the machine learning-based approach offers greater adaptability and better performance in handling noisy or incomplete data. Furthermore, when compared to distributed approaches using Apache Spark, as proposed by [9], our method stands out for its simplicity of implementation and efficiency on moderately sized datasets. Although distributed approaches are more suitable for very large datasets, their reliance on complex and costly computing infrastructures remains a notable limitation. In conclusion, these findings suggest that TabNet is the preferred choice for handling large datasets due to its faster execution time, while still maintaining competitive accuracy. However, further optimizations in hyperparameter tuning and data preprocessing are essential to ensure model stability and accuracy as data volumes continue to increase.

4.3. Interpretability analysis

One of the advantages of TabNet lies in its inherent interpretability, which allows us to track the model's decision-making. The global feature importance as calculated by the trained TabNet model is presented in Figure 16. We observe _birth_score to be the most common feature, with a huge majority of the model's predictions relying on it. This indicates that the model has learned to rely significantly on the date of birth as the primary predictor in discovering duplicate records. While this works efficiently on the FEBRL dataset, this high reliance on a single feature puts into perspective the importance of data quality for this specific attribute. The model employs soc_sec_id_score and address_1_score as secondary predictors, with the other features contributing minimally. This transparency is such a big deal, because it makes explicit suggestions as to how the model behaves, as opposed to typical 'black-box' techniques like the DNN.

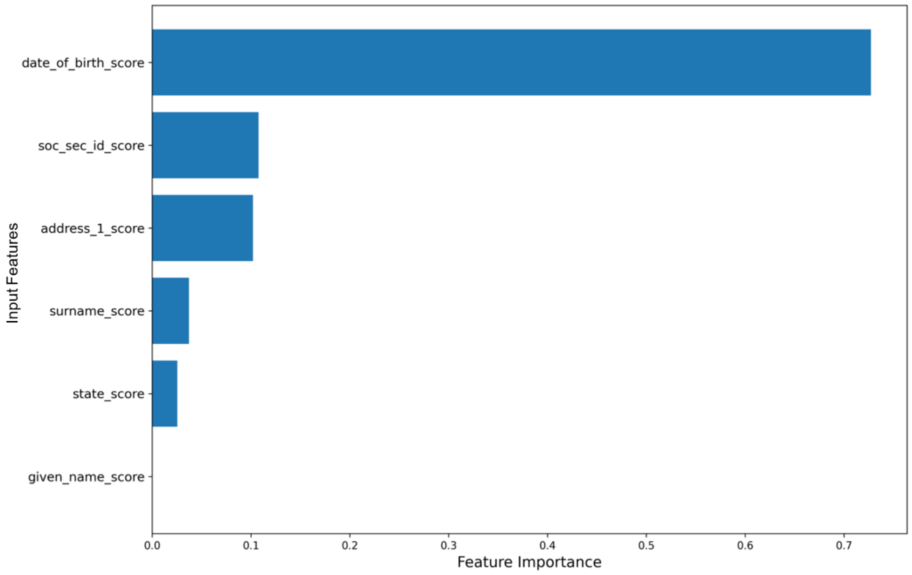
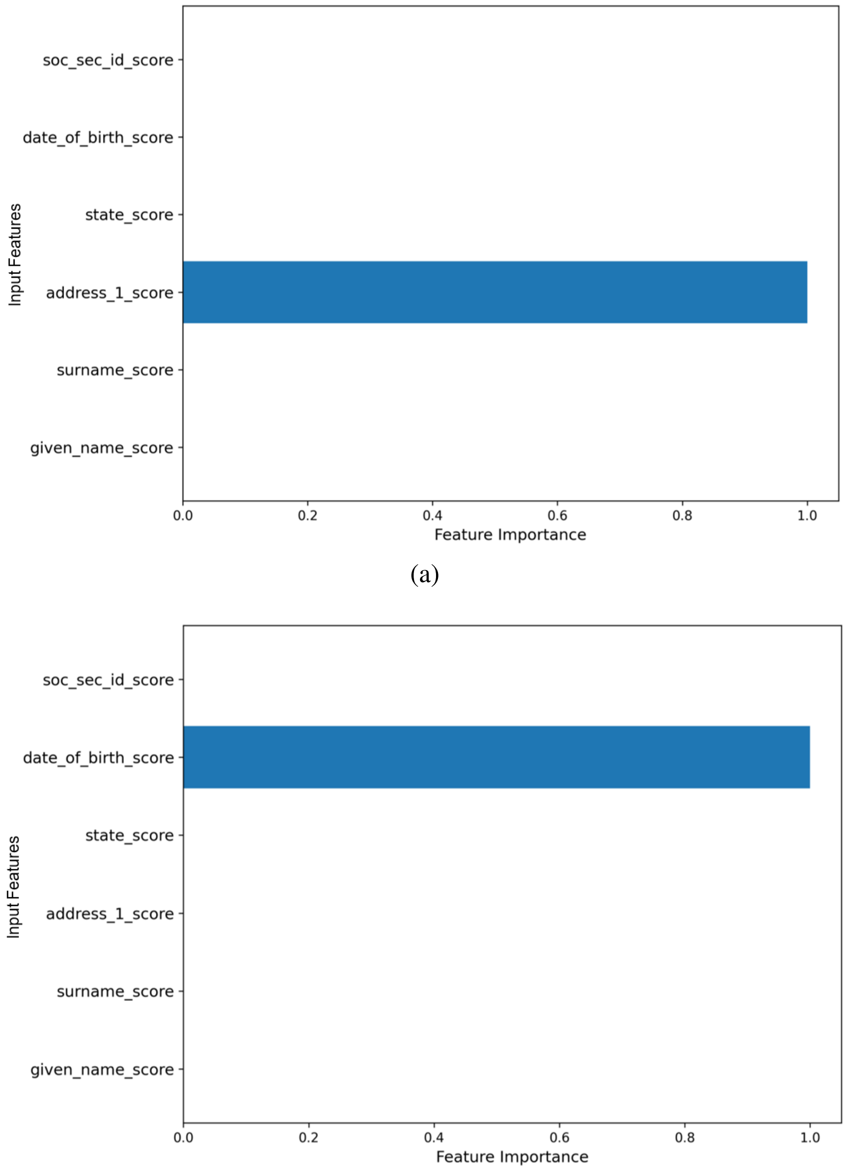


Figure 16. Global feature importance

Figure 17 illustrates model case-by-case logic. Figure 17(a) to confirm actual duplicate (case 1), it firmly focuses on a single piece of strong similarity evidence (in this case, the address). Conversely, Figure 17(b) to eliminate a non-duplicate (case 2), it sensibly picks one important difference (date of birth) to make its decision, even though other fields might be similar. This capability of adjusting its strategy based on the use of one prominent feature for every prediction illustrates clear and effective behavior.



(a)

(b)

Figure 17. Local interpretability: analysis of individual predictions, correctly identified of (a) duplicate, and (b) non-duplicate

5. CONCLUSION AND FUTURE WORK

The evaluation of the TabNet and DNN models highlights their effectiveness in addressing record linkage tasks, as evidenced by their high performance across precision, recall, and F1-scores, achieving values of 0.99, 0.98, and 0.98, respectively. While both models demonstrated comparable accuracy, TabNet distinguishes itself with its remarkable computational efficiency, reducing prediction time by approximately 79% when compared to DNN. This advantage makes TabNet particularly suitable for real-time applications and large-scale data processing. Furthermore, this study emphasizes the critical role of preprocessing and robust similarity measures such as JW and Levenshtein in enhancing model performance, particularly when dealing with noisy or inconsistent data. In comparison to traditional probabilistic models like the Fellegi-Sunter model, and even distributed solutions based on Apache Spark, the machine learning based approach proposed here demonstrates superior adaptability and ease of implementation for medium-sized datasets. Future work could explore the integration of TabNet into distributed infrastructures, such as Apache Spark or Hadoop, to better address big data challenges. Additionally, the development of semi-supervised or unsupervised methods to reduce reliance on labeled data, along with evaluating the approach on real world and diverse datasets, would be valuable steps for validating its generalizability. These efforts will further promote the adoption of TabNet and enhance its robustness for large-scale applications in big data environments. Future work will also focus on exploring the transferability of learned representations and domain adaptation techniques so that model generalizability can be enhanced. Inspiration will be taken from the success of transfer learning in related sequence matching tasks and an experiment will be done to see if feature representations learned by TabNet can be transferred to speed up training on new domains, especially where there is limited labeled data. Furthermore, development using semi-supervised or unsupervised methods is actually formally proposed as a main domain adaptation technique, through which dependence on labeled data is intended to be reduced and use of our solution in real-world heterogeneous environments like healthcare and finance made more convenient.

FUNDING INFORMATION

No funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Fatima Zahrae Saber	✓	✓	✓	✓	✓	✓	✓	✓	✓					
Ali Choukri		✓	✓	✓		✓			✓	✓	✓	✓	✓	✓
Mohamed Amnai	✓	✓							✓	✓	✓	✓	✓	✓
Abderrahim Waga	✓				✓	✓	✓	✓		✓				

C	: **C**onceptualization	I	: **I**nvestigation	Vi	: **Vi**sualization
M	: **M**ethodology	R	: **R**esources	Su	: **Su**pervision
So	: **So**ftware	D	: **D**ata Curation	P	: **P**roject Administration
Va	: **Va**lidation	O	: Writing - **O**riginal Draft	Fu	: **Fu**nding Acquisition
Fo	: **Fo**rmal Analysis	E	: Writing - Review & **E**diting		

CONFLICT OF INTEREST STATEMENT

No conflict of interest.

INFORMED CONSENT

We have obtained informed consent from all individuals included in this study.

DATA AVAILABILITY

The data that support the findings of this study are available on request from the corresponding author, [FZS]. The data, which contain information that could compromise the privacy of research participants, are not publicly available due to certain restrictions.

REFERENCES

[1] A. Sayers, Y. B.- Shlomo, A. W. Blom, and F. Steele, “Probabilistic record linkage,” *International Journal of Epidemiology*, vol. 45, no. 3, pp. 954–964, 2016, doi: 10.1093/ije/dyv322.

[2] A. Prabhugouda and S. Asra, “A review on big data applications and their challenges,” *Journal of Information and Knowledge Management*, vol. 23, no. 6, 2024, doi: 10.1142/S0219649224300018.

[3] S. F. Zahrae, C. Ali, and A. Mohamed, "Record linkage approaches in big data: a comprehensive review," *2024 International Conference on Intelligent Systems and Computer Vision (ISCV)*, 2024, doi: 10.1109/ISCV60512.2024.10620076.

[4] K. O'Hare, A. J.- Loughrey, and C. de Campos, "A review of unsupervised and semi-supervised blocking methods for record linkage," in *Linking and Mining Heterogeneous and Multi-view Data*, Cham, Switzerland: Springer, 2019, pp. 79–105, doi: 10.1007/978-3-030-01872-6_4.

[5] R. Mohamed, A. El-Bastawissy, E. Nasr, and M. Gheith, "Comparative study of record linkage approaches for big data," *Walailak Journal of Science and Technology*, vol. 18, no. 2, pp. 1–22, 2021, doi: 10.48048/wjst.2021.7221.

[6] A. J.- Loughrey, "Deep learning based approach to unstructured record linkage," *International Journal of Web Information Systems*, vol. 17, no. 6, pp. 607–621, 2021, doi: 10.1108/IJWIS-05-2021-0058.

[7] S. Arık and T. Pfister, "TabNet: attentive interpretable tabular learning," *35th AAAI Conference on Artificial Intelligence (AAAI)*, vol. 8A, pp. 6679–6687, 2021, doi: 10.1609/aaai.v35i8.16826.

[8] J. Prindle, H. Suthar, E. P.- Hornstein, and R. Foust, "Probabilistic record linkage for families (PRLF): a discussion of the development and validation of this open-source linkage tool," *International Journal of Population Data Science*, vol. 9, no. 5, 2024, doi: 10.23889/ijpds.v9i5.2763.

[9] M. Heydari, R. Sarshar, and M. A. Soltanshahi, "Distributed record linkage in healthcare data with apache spark," *arXiv:2404.07939*, 2024.

[10] L. Wolcott, W. Clements, and P. Saripalli, "Scalable record linkage," *Proceedings - 2018 IEEE International Conference on Big Data, Big Data*, pp. 4268–4275, 2018, doi: 10.1109/BigData.2018.8622516.

[11] H. Yulianton and R. C. N. Santi, "Product matching using sentence-BERT: a deep learning approach to e-commerce product deduplication," *Engineering and Technology Journal*, vol. 9, no. 12, 2024, doi: 10.47191/etj/v9i12.14.

[12] Z. Jin, J. Kim, H. Yeo, and S. Choi, "Transformer-based map matching model with limited ground-truth data using transfer-learning approach," *arXiv:2108.00439*, 2021.

[13] R. Peeters and C. Bizer, "Supervised contrastive learning for product matching," *WWW '22: Companion Proceedings of the Web Conference 2022*, pp. 248-251, 2022, doi: 10.1145/3487553.3524254.

[14] Y. Wang, M. Xu, and S. Zhong, "Enhancing privacy in lightweight data encoding for sensitive applications," *IEEE Access*, vol. 13, pp. 118406–118422, 2025, doi: 10.1109/ACCESS.2025.3580958.

[15] T. Ranbaduge, D. Vatsalan, and M. Ding, "Privacy-preserving deep learning based record linkage," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 6839–6850, 2024, doi: 10.1109/TKDE.2023.3342757.

[16] V. Borisov, T. Leemann, K. Sebler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep neural networks and tabular data: a survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 6, pp. 7499–7519, 2024, doi: 10.1109/TNNLS.2022.3229161.

[17] M. Zhekova, "An algorithm for exploratory analysis and normalization of big data with pandas," *Comptes Rendus de L'Academie Bulgare des Sciences*, vol. 76, no. 11, pp. 1716–1723, 2023, doi: 10.7546/CRABS.2023.11.09.

[18] V. N. G. Raju, K. P. Lakshmi, V. M. Jain, A. Kalidindi, and V. Padma, "Study the influence of normalization/transformation process on the accuracy of supervised classification," *Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 729–735, 2020, doi: 10.1109/ICSSIT48917.2020.9214160.

[19] F. Azzalini, S. Jin, M. Renzi, and L. Tanca, "Blocking techniques for entity linkage: a semantics-based approach," *Data Science and Engineering*, vol. 6, no. 1, pp. 20–38, 2021, doi: 10.1007/s41019-020-00146-w.

[20] A. Samiei and F. Naumann, "Cluster-based sorted neighborhood for efficient duplicate detection," *IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 202–209, 2016, doi: 10.1109/ICDMW.2016.0036.

[21] Z. Peng and H. Qin, "A single-individual based variable neighborhood search algorithm for the blocking hybrid flow shop group scheduling problem," *Egyptian Informatics Journal*, vol. 27, 2024, doi: 10.1016/j.eij.2024.100509.

[22] T. C. Ong, L. M. Duca, M. G. Kahn, and T. L. Crume, "A hybrid approach to record linkage using a combination of deterministic and probabilistic methodology," *Journal of the American Medical Informatics Association*, vol. 27, no. 4, pp. 505–513, 2020, doi: 10.1093/jamia/ocz232.

[23] Y. Jiao *et al.*, "A new hybrid record linkage process to make epidemiological databases interoperable: application to the GEMO and GENEPSO studies involving BRCA1 and BRCA2 mutation carriers," *BMC Medical Research Methodology*, vol. 21, no. 1, 2021, doi: 10.1186/s12874-021-01299-6.

[24] S. C. Cahyono, "Comparison of document similarity measurements in scientific writing using Jaro-Winkler distance method and paragraph vector method," *IOP Conference Series: Materials Science and Engineering*, vol. 662, no. 5, 2019, doi: 10.1088/1757-899X/662/5/052016.

[25] I. C. Bu'ulolo, M. I. Siregar, and C. F. Simanjuntak, "Comparison between Jaro-Winkler distance algorithm and Winnowing algorithm in detecting word similarities in Indonesian documents," *AIP Conference Proceedings*, vol. 2658, 2022, doi: 10.1063/5.0111181.

[26] F. Friendly, "Jaro-Winkler distance improvement for approximate string search using indexing data for multiuser applications," *Journal of Physics: Conference Series*, vol. 1361, no. 1, 2019, doi: 10.1088/1742-6596/1361/1/012080.

[27] P. J. Rao, K. N. Rao, S. Gokuruboyina, and K. N. Neeraja, "An efficient methodology for identifying the similarity between languages with Levenshtein distance," *Proceedings of the 6th International Conference on Communications and Cyber Physical Engineering*, vol. 1096, pp. 161–174, 2024, doi: 10.1007/978-981-99-7137-4_15.

[28] I. A. Fares and M. A. Elaziz, "Explainable TabNet transformer-based on Google Vizier optimizer for anomaly intrusion detection system," *Knowledge-Based Systems*, vol. 316, 2025, doi: 10.1016/j.knosys.2025.113351.

[29] V. C. Ta *et al.*, "TabNet efficiency for facies classification and learning feature embedding from well log data," *Petroleum Science and Technology*, vol. 42, no. 25, pp. 4610–4625, 2024, doi: 10.1080/10916466.2023.2223623.

[30] N. Barlaug and J. A. Gulla, "Neural networks for entity matching: a survey," *ACM Transactions on Knowledge Discovery from Data*, vol. 15, no. 3, 2021, doi: 10.1145/3442200.

[31] Z. Zhang, S. Xu, S. Zhang, T. Qiao, and S. Cao, "Attention based convolutional recurrent neural network for environmental sound classification," *Neurocomputing*, vol. 453, pp. 896–903, Sep. 2021, doi: 10.1016/j.neucom.2020.08.069.

BIOGRAPHIES OF AUTHORS

Fatima Zahrae Saber is currently a Ph.D. student at Ibn Tofail University, Faculty of Science, Kenitra, Morocco. Her thesis focuses on the optimization of record linkage approaches in big data, which she began in 2023. She received her bachelor's degree in Mathematics and Computer Science in 2020, followed by a master's degree in big data and cloud computing in 2022. Her research interests lie in data integration, machine learning, and scalable computing. She is passionate about scientific research and actively engaged in developing efficient solutions for large-scale data processing. She can be contacted at email: fatimazahrae.saber@uit.ac.ma.

Ali Choukri he is a professor at the Faculty of Sciences, Ibn Tofaïl University, Kenitra, Morocco. He specializes in software engineering, artificial intelligence, and algorithms. His research interests include intelligent systems, optimization techniques, algorithm design, and machine learning. He has contributed to several scientific publications and actively supervises Ph.D. students in the fields of artificial intelligence and big data. He is committed to advancing education and innovation in computer science. He can be contacted at email: ali.choukri@uit.ac.ma.

Mohamed Amnai is a professor at the Faculty of Sciences, Ibn Tofaïl University, Kenitra, Morocco. His areas of expertise include artificial intelligence, data structures, and data mining. He is actively involved in academic research and has published numerous scientific papers in international journals and conferences. His work focuses on intelligent systems, knowledge discovery, and the design of efficient algorithms. He also supervises graduate and postgraduate students in computer science and artificial intelligence. He can be contacted at email: mohamed.amnai@uit.ac.ma.

Abderrahim Waga holds the position of assistant professor in Computer Science and Artificial Intelligence at the Euro-Mediterranean University of Fez (UEMF) and is a researcher specializing in mobile robotics. His academic journey, which began with a bachelor's degree in Mathematics and Computer Science and a master's in Embedded Systems, culminated in a Ph.D. in Computer Science from Moulay Ismail University in 2025. His research focuses on mobile robot navigation through deep learning. He has developed end-to-end autonomous navigation systems based on innovative hybrid models that combine convolutional neural networks (CNNs) for visual perception with machine learning for decision-making. He can be contacted at email: a.waga@ueuromed.org.