International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V73I8P101

Volume 73 Issue 8, 1-13, August 2025

Original Article

Multimodal Analytical Approach for Determination of Deduplication in Names for Identifying People during Emergency Situations

Nagesh Raykar[1], Prema Sahane[2], Sonali Rangdale[3], Dipmala Salunke[4], Pallavi Tekade[5], Pramod Patil[6]

[1]*Department of Information Technology, University of Jhunjhunu, Rajasthan, India.*
[2]*Department of Computer Engineering, JSPM's Rajarshi Shahu College of Engineering, Maharashtra, India.*
[3]*G H Raisoni International Skill Tech University, School of Engineering and Technology, India.*
[4,5]*Department of Information Technology, JSPM's Rajarshi Shahu College of Engineering, Maharashtra, India.*
[6]*Dr. D. Y. Patil Institute of Technology, Pune, India.*

[4]*Corresponding Author : dtsalunke_it@jspmrscoe.edu.in*

Received: 27 January 2025 Revised: 07 July 2025 Accepted: 21 July 2025 Published: 30 August 2025

Abstract - *Phonetic algorithms are developed to index words based on their pronunciation and are primarily developed for the English language. Demographic Data (DD) gives information about people according to certain attributes like name, age, gender, residence, occupation, etc. In hospitals, government record matching, and multilingual information retrieval systems during emergency situations, it becomes vital to quickly and accurately identify a person, and many times, confusion is created due to duplication in records. The records do not fetch the names if their alphabetical order is incorrect while writing the same names. Phonetic name identification also provides important statistics in web analysis. Though there are many existing studies handling the DD, no specific study has been done to deal with the Indian regional language. The proposed research compares the conventional regional names of format First Name (FN) and Last Name (LN) based on the phonetic rule. Research proposed a novel, efficient phonetic-based algorithm for the regional language. Attempts have been made to prevent name repetition and similar names, even with different alphabetical arrangements. There are emergency situations, especially while finding a next of kin, finding a person during national security issues, or in any emergency situation when not much information is available, but locating a person or informing the family about the situation is important. Many times, the person's information is available, but the database does not fetch it if there is a dissimilarity in the spellings of the names. This research is trying to apply multimodal approaches to combine NLP and machine learning approaches for identifying people during emergency situations. Results from the suggested approach are promising, and for a real-time environment, it can be applied.*

Keywords - *Demographic, Indexing, Indian regional languages, Machine Learning, Natural Language Processing (NLP), Phonetic algorithm.*

1. Introduction

1.1. Background Study

Traditional phonetic algorithms are made for English and don't work well on Indian languages because of their complicated phonetics and irregular spellings. Current techniques frequently only work with certain languages, such as Gujarati or Hindi, and have trouble with string comparisons. These algorithms do not work for words in other languages. Indian languages do not rely on sounds. Algorithms for English are sometimes not useful for indexing words in other languages like Marathi, French, Spanish, and German. DD gives information about people according to certain attributes like name, age, gender, residence, occupation, etc. DD plays an important role in web analysis by offering statistics. To obtain precise information, it is essential to eliminate duplicate data. In recent years, numerous researchers have made contributions to the field of demography [1]. When handling DD, removing duplicate data is vital. A reduction rule is necessary to aid the deduplication algorithm in Indian DD.

1.2. Literature Survey

In [2], the authors use a Deep learning approach for processing India-centric DD variations in terms of naming conventions, address, etc. The authors customize the DeepMatcher algorithm. In [3], the authors propose rules for designing an algorithm to remove duplication. It is based on Indian DD containing the FN and LN strings. These rules help to reduce NSs to generic strings. In [4], the authors present the parallel duplication removal algorithm FER- APARDA.

Probabilistic record linkage has been employed to detect duplication in datasets. In [5], the authors state that DD and data reduction are two techniques for the removal of redundant data. Many research papers are studied, and an attempt is made to summarize various aspects of the data deduplication process. Chunk-based DD techniques are elaborated. In [6], the authors elaborate on GRID3, which generates and uses Geospatial (GS) data on infrastructure, residential colonies, population and boundaries. The inputs from the government, academic institutions, and companies to design various GS solutions are combined in GRID3.

Recent research has focused on improving name matching and deduplication techniques across various domains. In [7], the author showed that match identification across jurisdictions was greatly enhanced when all known names for individuals were used in HIV monitoring data. In [8], the authors build a disease model using an updated deprivation index and DD. The results prove to be great for predicting chronic diseases. In [8], the authors use data mining techniques to analyze the relationships between various attributes. They implement ARIMA models. Monte Carlo Simulation is explained for target-based progress tracking. In [9], the authors state that DD is the information that extracts the characteristics. It can be used to study the population and, in turn, can do predictions. In [10], the researchers used a Convolutional Neural Network (CNN) for the extraction of features from profiles. An SVM is used to extract the characteristics of the customers. The proposed CNN-based method uses an Irish dataset. It extracts the demographic consumer information. In [11], the authors elaborate on the five most important methods for multimodal data analysis.

It includes 1) Multimodal Analysis, 2) Systemic Functional Discourse Analysis, 3) Mediated Discourse Analysis, 4) Multimodal Conversation Analysis and 5) Social Semiotics. The application of the method, data collection, analysis, and theoretical foundations have also been discussed. In [12], the authors propose a method to analyse and differentiate the introduction of Monoacetyldiglycerides (MAcDG) from acetylated Triglycerides (TG), a multimodal analytical approach. In [13], the authors proposed an efficient algorithm to deduplicate demographic data using phonetic-based reduction of name strings. Data deduplication has become an important aspect of data analytics. Deduplication can be performed on files, fixed-size blocks, or variable-size chunks. They also state that Content-Defined Chunking (CDC) can overcome the boundary-shifting problem and is hence mostly employed for deduplication. The authors combine computational models with multimodal frameworks for natural language processing, cloud computing, big data, and other related technologies. They suggest that merging these fields could enhance the effectiveness of computational tools and techniques [14-16]. Phonetic algorithms have shown promise in improving fuzzy matching and deduplication processes [17]. These algorithms can be tailored to specific languages and domains, such as Ukrainian surnames or medicinal names. Efficient deduplication methods often involve enrollment and deduplication, using phonetic reduction rules to create generic name strings [18]. Various similarity metrics and algorithms exist for detecting approximate duplicates, with techniques to enhance efficiency and scalability [19]. Individual names have unique characteristics that require consideration when applying matching techniques [20]. Experimental comparisons have indicated that there is no single best matching technique for all scenarios, highlighting the need for careful selection based on the specific application and data characteristics [21].

After an extensive literature survey, it is identified that though there are many existing studies handling the DD, no specific study has been done to deal with the Indian regional language. In the proposed research, the conventional regional names will be compared based on the phonetic rule. A novel, efficient phonetic-based algorithm for the regional language has been proposed. An attempt has been made to avoid redundancy from the Indian DD.

1.3. Objectives of the Paper

- To develop a robust algorithm for word extraction from Indian DD.
- To develop an algorithm to deduplicate data on the basis of similarity in the phonics.
- Compare the existing techniques with the proposed technique.

2. Methodology

2.1. Working of the Existing Algorithm

The phonetic algorithm focuses on the indexing of words and is based on pronunciation. Words from regional languages like Marathi, French, Spanish, and German may not be indexed by these algorithms since they were created for the English language. The following existing algorithms have been studied in depth.

- Soundex Phonetic
- Daitch-Mokotoff
- Refined Soundex
- NYSIIS
- Cologne Phonetic
- Caverphone (1.0 & 2.0)
- Metaphone
- MRA (Match Rating Approach)

2.1.1. Soundex Phonetic

Soundex is a phonetic-based algorithm used to encode names according to English pronunciation by sound. It will give the indexing of the name. The Soundex algorithm generates a 4-digit string. The benefit of the Soundex technique is that it avoids or keeps away from most issues related to the misspellings of family names. The Soundex

method or system is a beneficial tool in searching for or penetrating forebears since the misspelling of family names was a usual event in official documents [13].

2.1.2. Daitch-Mokotoff Soundex

The Soundex algorithm created for Eastern European surnames is the most recent iteration. The algorithm is primarily used for discovering close or nearby matches, i.e. hereditary common names, which involve Jewish and Russian names. The Soundex algorithm ciphers it to a complete 6-digit cipher. The transformation guidelines of D-M Soundex are much more complex as they require groups or categories of characters to be enciphered. This algorithm is designed for the American Soundex to have significant accuracy or correctness in matching Yiddish and Slavic surnames. These algorithm surnames have similar pronunciations but spelling variation or differentiation, so the D-M Soundex algorithm is used [13].

2.1.3. Refined Soundex

It is a modified Soundex algorithm optimized for spell-checking. The Refined Soundex algorithm changes the letter bins from Soundex, allowing for a closer match. This algorithm Encodes tokens using an improved version of the index [20].

2.1.4. NYSIIS

This algorithm was developed for the New York State Intelligence and Identification System. This technique is used to translate a name to phonetic name coding up to six letters or characters. It will increase the accuracy by 2.7% as compared to the conventional Soundex algorithm. The NYSIIS is a part of the Criminal Justice Services Division of New York State. Before applying the algorithm, the input string should be translated to capital or upper case with all white space eliminated [20].

2.1.5. Cologne Phonetic

The Soundex algorithm gives similar codes to characters with the same sounds, making it a comparable technique. Finding a similarity in the midst of words or letters is possible with this approach. According to the Cologne phonetic algorithm, every letter in a word corresponds to a number between 0 and 8. The next most appropriate digit to use in the situation is chosen. Using the conditions, need to select the appropriate number for the final. A few rules are applied in a specific manner to the initial characters of words. By doing it this way, it sounds as if it assumes to allocate identical code [13].

2.1.6. Caver Phone

It is devised to identify the English names with their sounds. The Caverphone technique was developed in the Caversham Project. This algorithm has two versions- the initial version was developed in 2002, and the revised version was developed in 2004. This method was created for the current pronunciation of the way to present in the south of Dunedin, New Zealand, which is the research area [22]. There are two versions of this algorithm: Caverphone 2.0, which was developed in 2004, and Caverphone 1.0, which was developed in 2002 [23]. The common phonetic match is better suited for the Caverphone 2.0 Algorithm. This algorithm was freely available for use [24].

2.1.7. Metaphone

This algorithm is used to index words and symbols according to how they are spoken. This improved Soundex algorithm takes advantage of the English language variety and inconsistency in data. The pronunciation will result in a very accurate encoding and be helpful in matching words or characters, as well as names that are the same or of a similar sort. For a similar name, Soundex will assign the same keys to similar word sounds. This approach is accessible in several systems with built-in operators [25].

2.1.8. Double Metaphone

Philips produced a new version of the algorithm, and it was given the name Double Metaphone. Original Metaphone is limited to English phonetic encoding criteria. Double Metaphone is better than the Metaphone phonetic algorithmic technique [26].

2.1.9. Match Rating Approach

The MRA Phonetic Algorithm is developed for indexation and differentiation of names that are pronounced the same. Simple encoding methods and rules are used in this technique. By measuring the strings from left to right and then from right to left, the main mechanism-similarity differentiation-calculates the amount of unmatched or unequal words or characters and eliminates similar words or characters. After deducting this number from 6, a minimal threshold is measured. Another name for the ciphered name is Personal Identification Number (PIN). There can never be more than six alpha-only words or letters in the ciphered name [27, 28].

2.2. Procedure Followed in the Existing Method

2.2.1. Genetic Deduction Rule

The count of unique names can be significantly reduced in a genetic deduction-based method that just takes into account the phonetic portion. It benefits to minimize exploration space in the process. This component has several roles utilized to decode Indian names (surname and name) into general names. Examples such as 'Nitin','Niteen', and'Neetin' can be minimized to the general name 'Nitin'. A few phonetic acts of trimming on every name of the string can be explained to discover the general name. Phonetics is the part of linguistics that consists of the study of the sound of human speech. Phonetic sounds are divided into vowels and consonants. To change a Name String (NS) into a common NS Phonetic reduction laws, some rules have been used. The proposed reduction law can be explained with the following example: Dileep can be replaced by Dilip (similar phonetic reduction).

2.2.2. Creation of a Database

In the existing methodology, the rules are utilized to translate a minimized generic name and add it to the database. Different name strings might be converted into distinct generic names. A common name that carries the set of NSs can be used to explain a bin. Names can be the same, but IDs can differ.

Therefore, in the creation of an Singly Linked List (SLL), where each node has an NS with non-identical IDs, one can operate as the bin. Insertion and deletion can be done using a single-linked list.

2.2.3. Matching Strategy

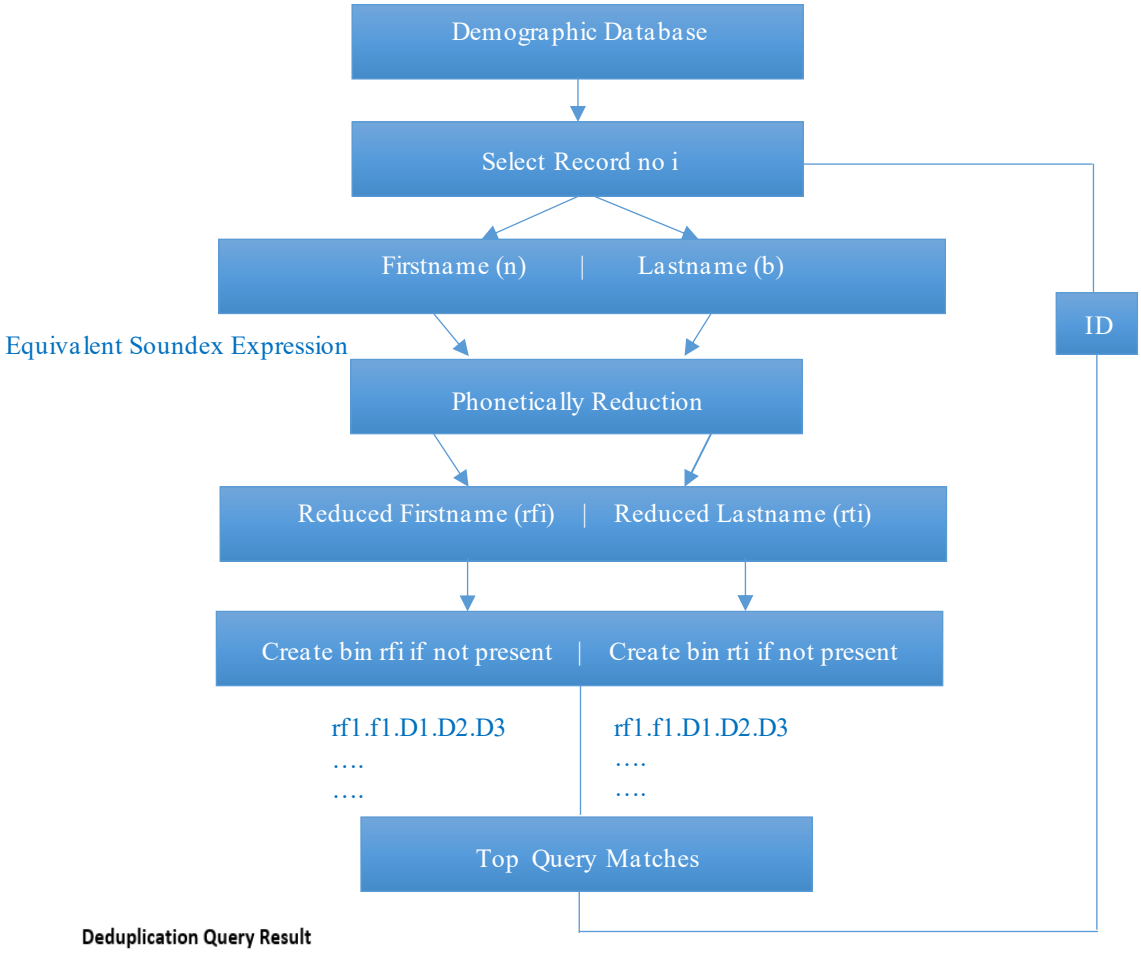
The Edit Distance (ED) algorithm is used to calculate the difference between two strings and perform a minimum number of operations, such as transposing, replacing, deleting, and inserting. Levenshtein distance is the case at the point of ED, which estimates the distance between two strings of characters. To calculate the normalized distance, the string is divided using a maximum of two NSs. NSs linked to the bins are regarded as the most likely applicants for the duplicates, as their standard Edit Distance from the query data is within the acceptable limit.

2.2.4. Working

The approach used by the existing demographic approach is shown in the flowchart of Figure 1.

2.2.5. Experimental Result for the Existing Strategy

The existing methodology uses bin density for approximate representation of numerical or categorical data distribution. The results obtained from the existing strategy are shown below in Table 1. The existing system shows loopholes in phonetic reduction and specific string matching, so the existing techniques are not efficient in recognizing duplicates.



Deduplication Query Result

Query	Top Matches	Score
Yash Mehta	Mehta Yash Dilip	0.06
Yash Mehta	Mehta Yash Mahavir	0.08
Yash Mehta	Mehta Yash Prashant	0.09

Varying Threshold Value:

Distance	D1	D2	D3
0.06	62.5	68.4	71.4
0.08	64.1	69.9	72.1
0.10	66.3	70.6	73.2

Fig. 1 Flowchart of existing algorithm

Table 1. Experimental results for the existing strategy

Head Details	FN	LN
No. of bins	43944	27185
Minimum Bin Size (BS)	1	1
Maximum of BS	15	14
Average of BS	1.27	1.28
Standard Deviation	0.80	0.79
Reduction Percentage	92	94

3. Proposed Methodology

The ED algorithm is used to find the edit distance between two strings of characters and perform a minimum number of operations, such as replace, insert, transpose, etc. Levenshtein distance is the case at the point of ED, which estimates the distance between characters in two strings. This interval is divided using two NSs to obtain the normalized distance. NSs are linked to the bins whose ED from the query data's NS is not upto the allowed threshold and are thought about as the likely applicant for the duplicates. Figure 2 shows the high-level architecture of the algorithm.

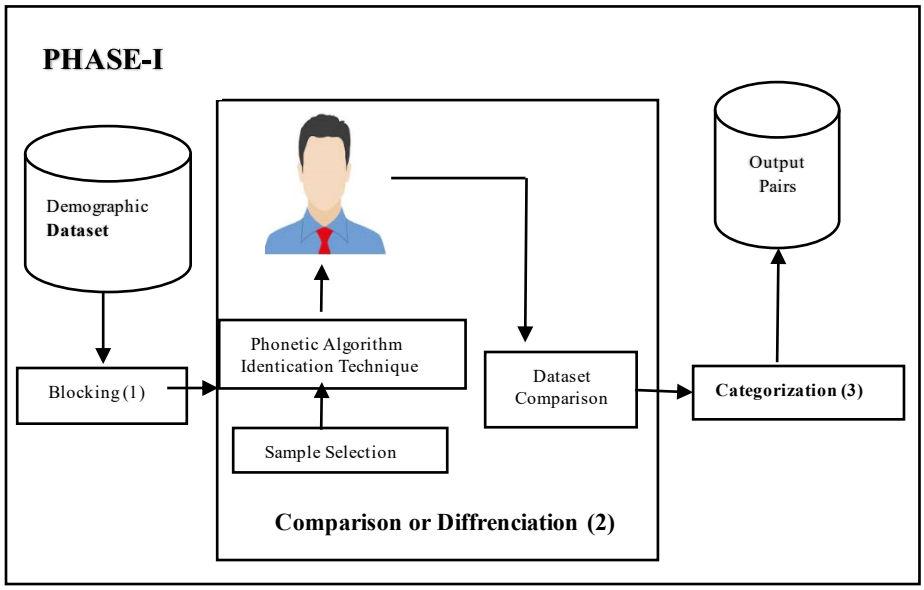


Fig. 2 The architecture of the proposed algorithm

The logical flow of the proposed methodology is as follows-

3.1. Demographic Dataset Preparation

In the Demographic, a customized dataset was prepared for this research. The dataset contains the FN, LN, and gender added and is always in running mode. This running mode dataset comes under the blocking stage.

3.2. Blocking Phase

This phase reduces the comparisons by grouping the data that shares common features. This creates training and an actual database.

3.3. Comparison Phase

The Comparison phase similarity function is applied to identify the strings of the same block. Finally, the Classification phase identifies matching and non-matching sets. In the comparison, the .csv or Excel dataset file is used to predict or analyze the demographic Information.

The user can supply the individual user demographic information. Accordingly, once the uploaded information passes through the developed phonetic algorithm. This developed phonetic technique translates the information to the encoded format, such as FN and LN.

3.3.1. Sample Selection Strategy

For testing purposes, the researcher needs to supply the sample inputs to the developed algorithm. So the sample selection strategy, such as individual upload or bulk file uploads, is done. If the proper data is not loaded, the algorithm will show the error log.

3.3.2. Phonetic Algorithm Identification Technique

Sample inputs are provided and converted to algorithmically encoded format.

3.3.3. Dataset Comparison

Dataset comparison means that the testing sample selection input compares with the trained input samples.

3.4. Categorization

In the Categorization phase, the trained dataset value matches the testing input sample accordingly, showing matching and non-matching input value details.

3.5. Output Pair

It contains the non-matching and matching pair details.

The proposed methodology is divided into four phases. The phases are as follows-

Phase I: Categorization, Phase II: Proposed algorithm application, Phase III: Apply ML processing and Translate Ciphered Key, and Phase IV: Classification.

3.5.1. Phase I - Categorization

The categorization phase has been depicted in Figure 3 as follows:

Categorization is carried out in phase I. The demographic information is supplied in CSV or XLSX format. The selection strategy is chosen. The file is uploaded. This input is given to the next Phase, i.e Phase-II.

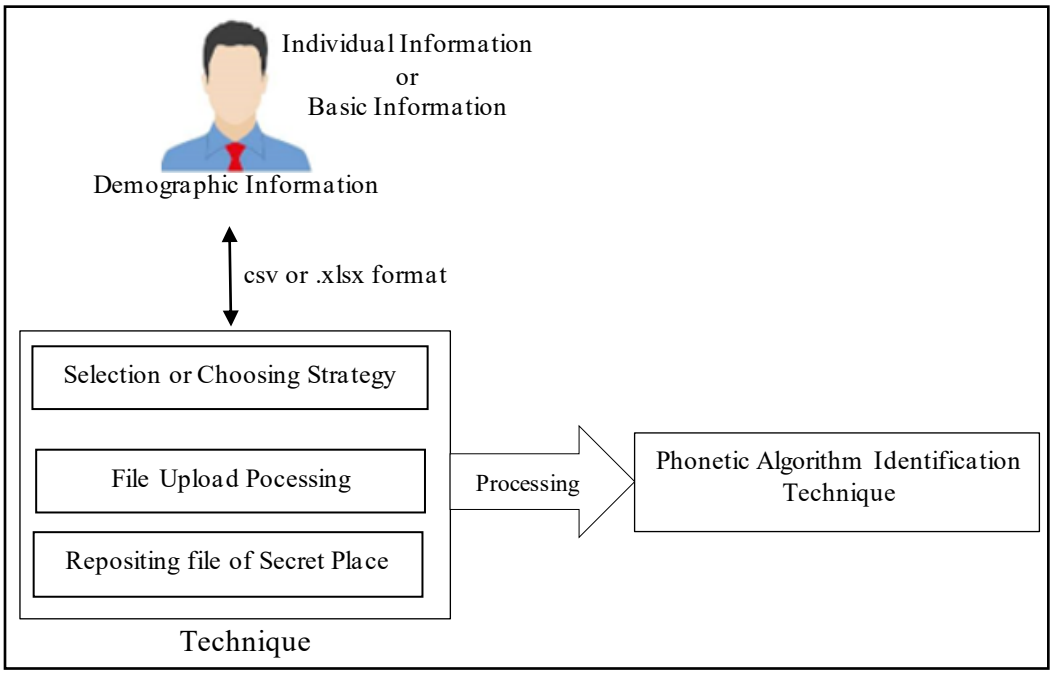


Fig. 3 Phase I - categorization

3.5.2. Phase II- Applying the Proposed Algorithm

Phase II: Phonetic Algorithm has been shown in Figure 4, and the steps are as follows:

- Supply the Inputs: In the supply input, the individual input or bulk input upload process is carried out.
- Uppercase Conversion of Inputs: Supplied inputs need to be converted into uppercase. For example, the name "Neelima' will be converted as 'NALIMA'
- Eliminate repeated words: once the name is converted to the uppercase format. The repetitive words need to be removed, and one letter in the repetitive word is marked.
- Alphabet is Vowel or Consonant: This part consists of conditional checking, such as whether the alphabet is a vowel or a consonant, so accordingly it will be categorized and will move on to the specific category, such as the alphabet is a vowel or alphabet is a consonant section. As per the condition, it will be processed for the next phase/process.

Alphabet is a Vowel

If alphabet is a vowel, then it will come to this vowel section. In the vowel section, vowels contain the following families, so accordingly applying the phonetic rules.

- Monophthongs
- Diphthongs
- Special Cases

Monophthongs: They consist of a vowel that has a single perceived auditory quality. It contains one or a single sound. It contains a pure vowel sound. It contains 12 pure sounds of monophthongs, e.g. 'New'

Diphthongs: They are two vowel sounds that are pronounced together to make one sound, for example, the aII sound in 'fine'.

- Special Case: In this special case scenario, sometimes 'W' and 'Y' are treated as the vowel.
- Apply Phonetic Rule: As per categorization of the Monophthongs, Diphthongs, or special case, apply the phonetic rule.
- Replace letter as A: For the vowel alphabet, replacing the letter as 'A', this is applicable for all vowels such as 'A', 'E', 'I', 'O, 'U' are considered as 'A'.
- Generate Code: Against the vowel, it generates the code.

Alphabet is a Consonant

A consonant is a kind of speech sound that is not a vowel.

Non-Vocalize consonant: Individual consonant mapped with equivalent encipher value. It will assign individual consonant-wise encipher values.

Repetitive consonant: If any repetitive consonant is present, it will be treated as a single letter.

- Apply phonetic Rule: As per the pronunciation of the alphabet, it will be enciphered.
- Enciphered consonant alphabet: In a special case scenario, the alphabet will be enciphered as per a special consonant condition, such as silent pronunciation and double consonant pronunciation. So, the condition will be handled.
- Generate code: As per consonant-wise, generate the Enciphered code.
- Final output Enciphered Encoded Value: Display the Name-wise enciphered value in the expected output.

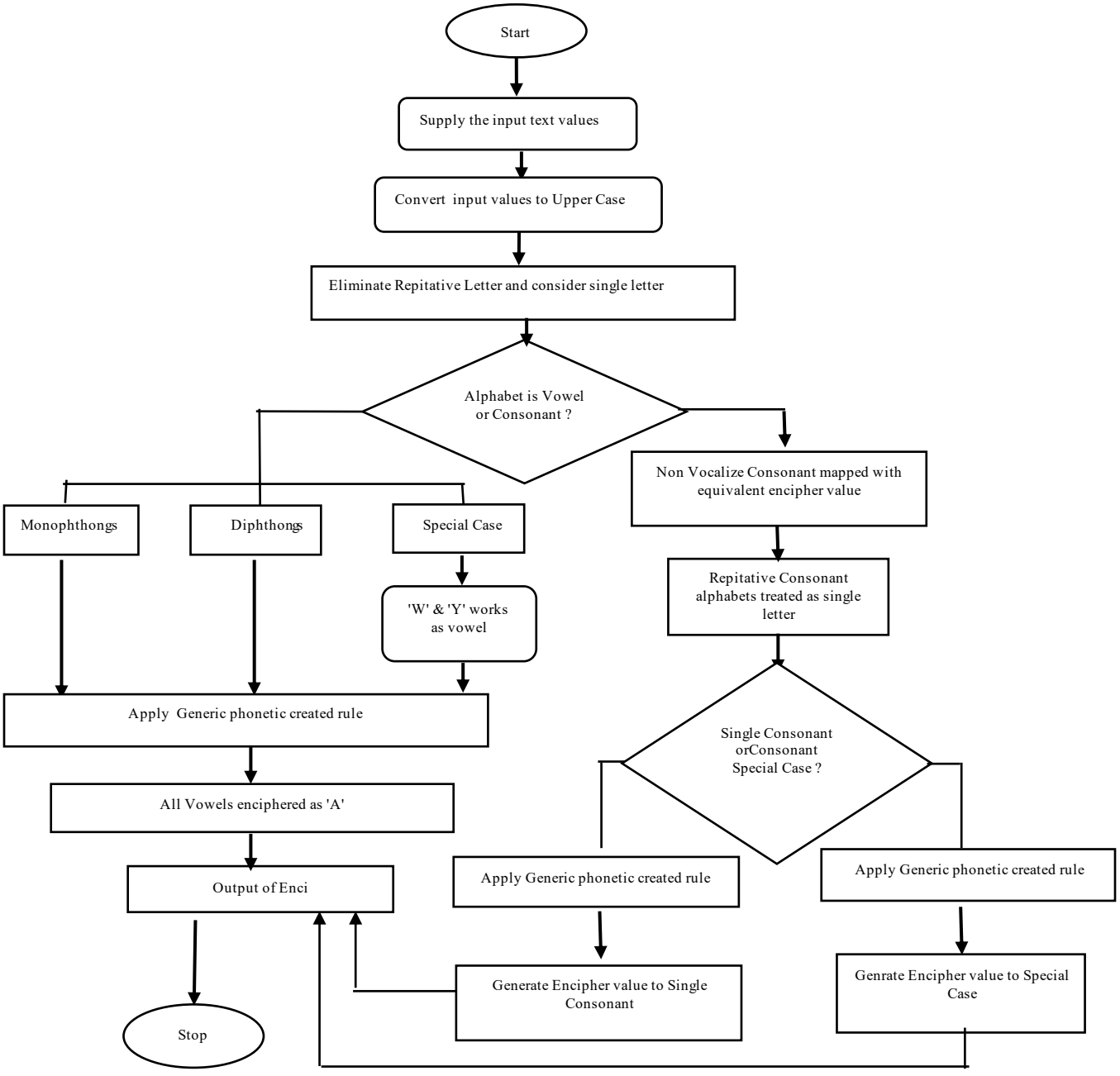


Fig. 4 Flowchart of phonetic algorithm

Enciphered encoded value vowel/alphabet is given as an input to the ML processing algorithm in phase III.

Phase III: Apply ML processing and translate the Ciphered Key Phase III is divided into two sub-phases: Phase A: In this phase, the ML algorithm is applied.

Phase B: Translation of Ciphered Key is carried out in this phase.

Phase III A: The ML algorithm process followed is as depicted in Figure 5.

Construct a Wordbook based on Encoded Pair

It consists of encoded (key/values names array dictionary)

Key : Encoded Text
Values : Name Arrays

3.5.3. Phase III B: Translate Ciphered Key

Key value pairs of names obtained in phase II are given as input in phase III, i.e. translating the ciphered key. In the conversion process of these encoded keys, the keys are translated into an array representation. This can be explained by using the example given below. Consider the name "NALIMA". The corresponding array will look as shown in Table 2. The following steps are followed for the translation.

- Splitting Dataset: generated input values are split with the help of a cluster.
- Assembled LSTM-Based Processing: In clustering input processing with the assembled LSTM. LSTM helps us with cluster-wise input separation.

Input : Three-dimensional array (Ciphered Keys)
Output : n * n (n: In cluster, unique number of ciphered keys) and Store LSTM Model.

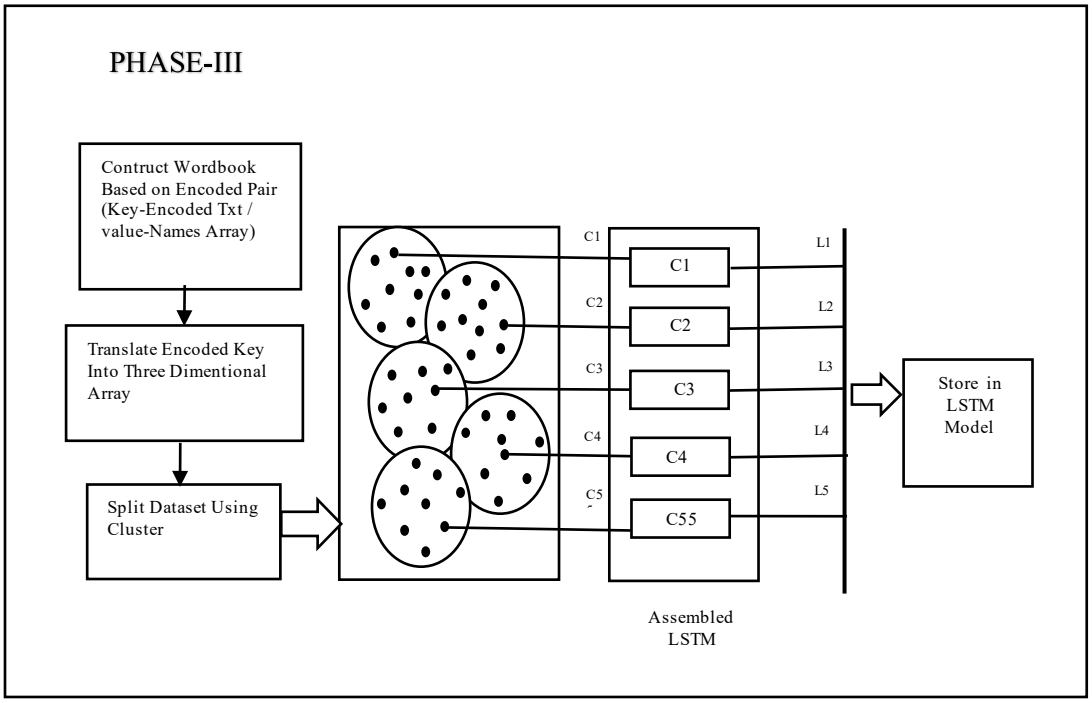


Fig. 5 Phase III-ML processing

3.5.4. Phase IV: Classification Technique

The last phase of the proposed algorithm is classification. The classification is shown in Figure 6.

The following steps are carried out in order to classify the names.

- Classify model: The Classify model is used for a detection model for workbook input value. As per the classification, it will find the match.
- Supply the Inputs: It supplies the input value.
- Encoded pair: The Input value should be in the encoded pair format
- Translate Ciphered Key: In the conversion process, these encoded keys are translated y into the three-dimensional array. Consider the string " NALIMA"; the array will look the same.

Find nearest Cluster: In the Classify technique, once the encoded pair is identified, the nearest cluster in the group will be found.

Load Cm LSTM Model: The Cm LSTM Model is loaded. In that model, inputs are received from the stored LSTM Model. Once the matching is done, the next process will be called, i.e., 10 Nearest Encoded Text.

Table 2. Array representation of "NALIMA"

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
													1												
1																									
											1														
								1																	
												1													
1																									

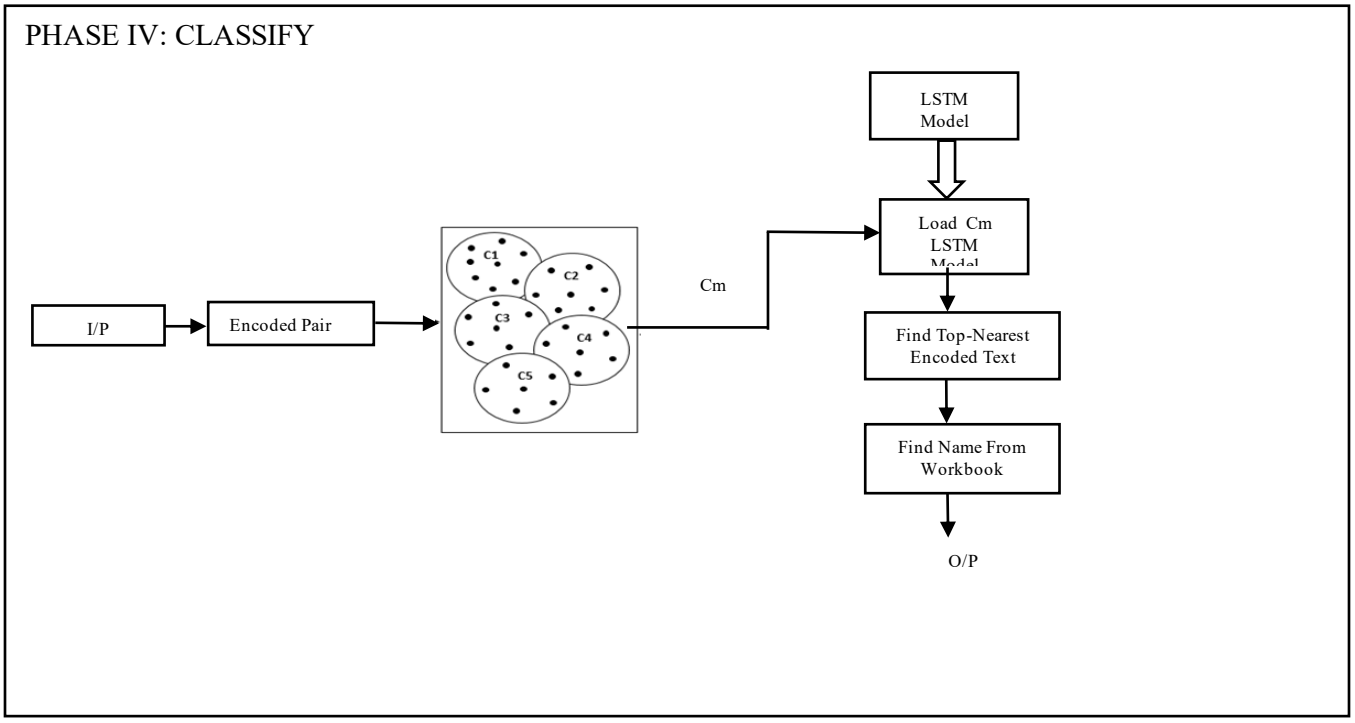


Fig. 6 Phase IV - classification technique

Find Top 10 Nearest Encoded Text: This phase contains the best 10 matches encoded results, displaying/showing to the end user.

- Find Name from Workbook: In the workbook, the details are getting the name with a possible combination.
- Output: Name Arrays (Top K match)

4. Results and Analysis

The results are compared with the existing algorithm. The existing algorithms have been compared with the proposed algorithm, which has been compared with Soundex, Refined Soundex, Daitch-Mokotoff Soundex, Nysiis, Caverphone2, Caverphone1, Caverphone, Cologne, Match Rating Approach Encoder, Double Metaphone, and Metaphone. Space, time complexity, recall, precision and accuracy have been calculated for all the algorithms. An accuracy comparison between the based algorithm and all other phonetic algorithms is shown in Figure 7.

The proposed phonetic-based algorithm has maximum accuracy. The next highest accuracy is shown by Refined Soundex. The accuracy of the existing and proposed algorithms is calculated for different numbers of records ranging from 100K to 500K.

The results are shown in Figure 8. From the above Figure, it is concluded that the proposed phonetic-based algorithm gives consistent accuracy with any number of records. For 100K, it shows an accuracy of 98%. The accuracy is maintained even for 500K records; the recall has been calculated. The results are shown in Figure 9 as follows:

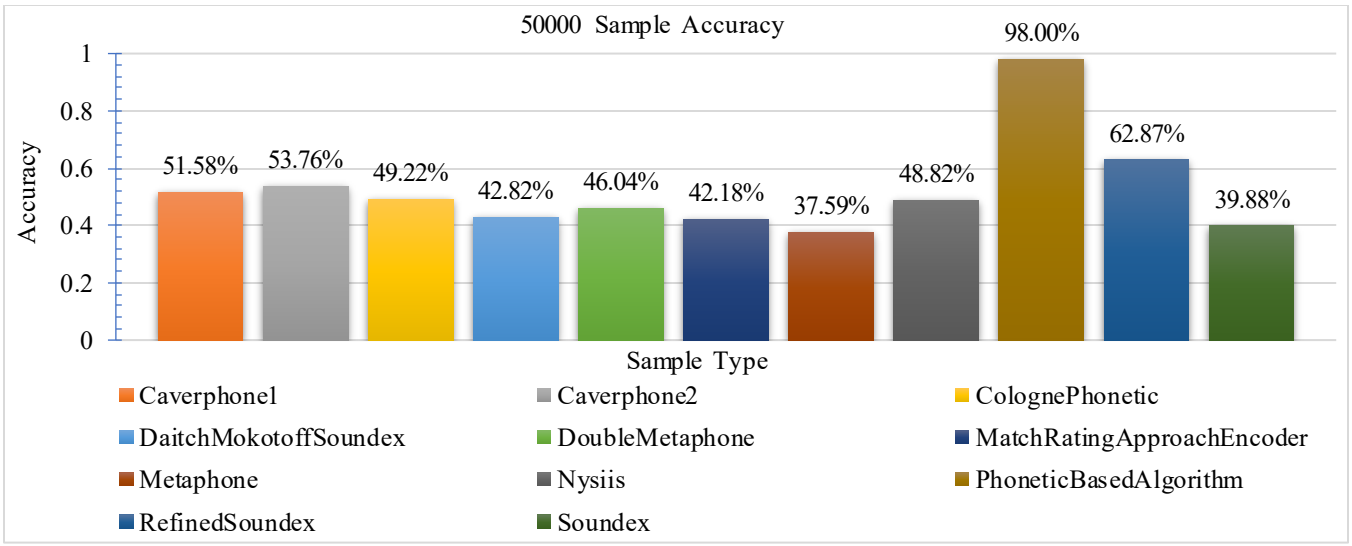


Fig. 7 Comparison of the accuracy of the phonetic-based algorithm with the existing algorithm

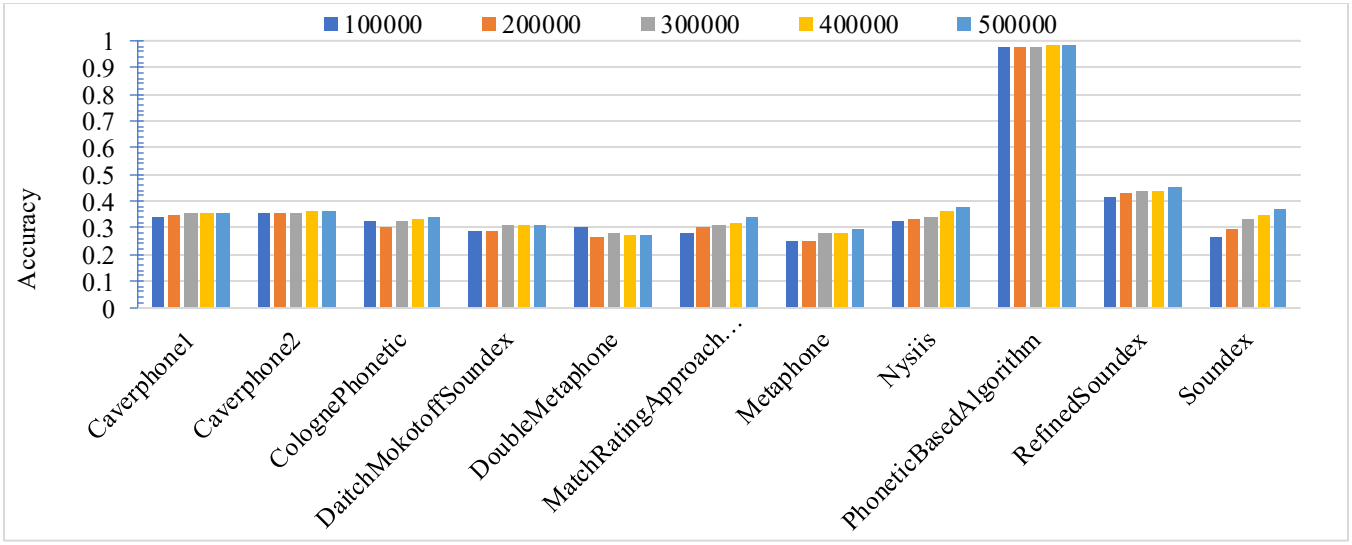


Fig. 8 Accuracy comparison for different numbers of records

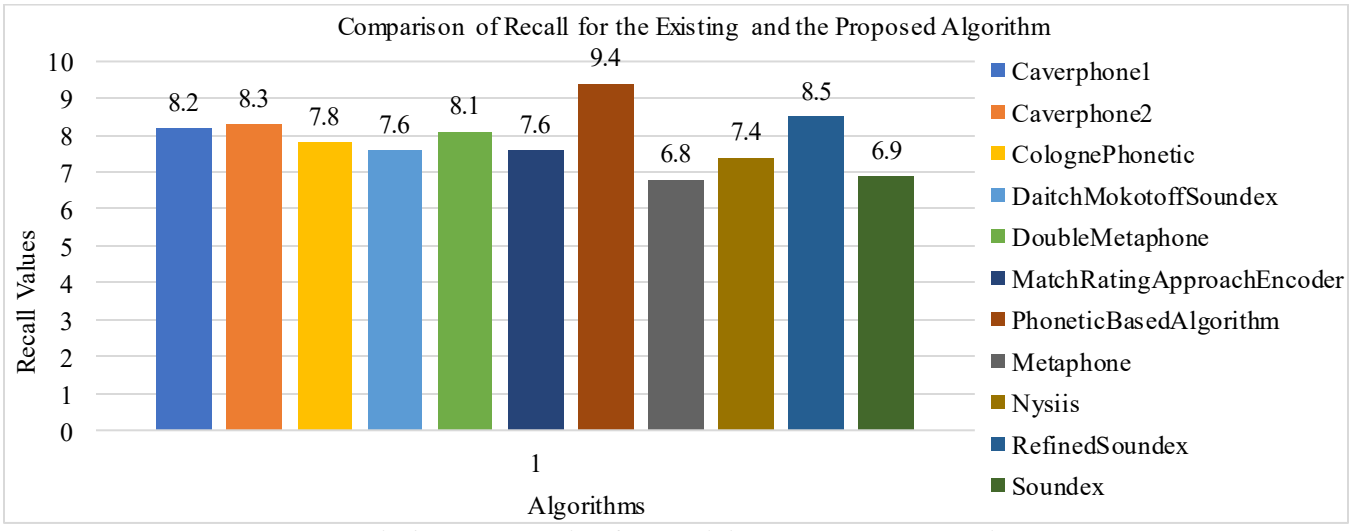


Fig. 9 Recall comparison for the existing and the proposed algorithm

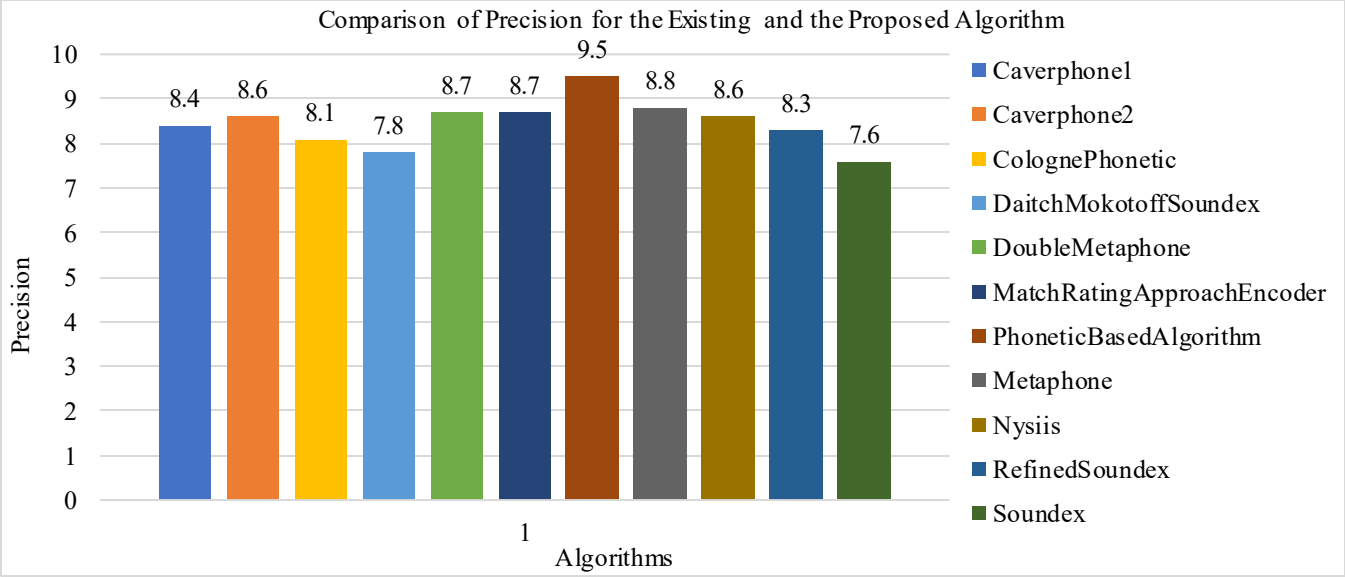


Fig. 10 Comparison of precision for the existing and the proposed algorithm

The proposed phonetic-based algorithm shows the highest recall of 9.4, followed by the highest recall by Refined Soundex. Precision has been calculated to gauge the performance of the proposed algorithm and the existing ones. The results are shown in Figure 10. The proposed phonetic-based algorithm works with the highest precision of 9.5, followed by the highest precision by Metaphone.

5. Discussion

In existing algorithms, the author uses bin density for the approximate representation of the distribution of numerical or categorical data. It has been observed that when the existing algorithm is applied to the Indian Name system, the reduction percentage values for the FN are 91% while those of LN are 95%. From this, it is evident that the existing system shows loopholes in phonetic reduction and specific string matching, so the existing techniques are not efficient in recognising phonetic-based duplicates. After executing the proposed algorithm on the database containing FN and LN format, the reduction percentage values for the FN are 98% while those of LN are 99% in the case of the proposed algorithm.

The existing algorithms have also been compared for accuracy, recall, and precision. The proposed algorithm has shown an accuracy of 98.00 %, while Refined Soundex shows an accuracy of 62.87%, which is quite low compared to the proposed algorithm. Among the existing phonetic-based algorithms, the proposed algorithm shows the highest accuracy. The recall for all the existing algorithms, as well as the proposed algorithm, has been calculated. The proposed algorithm has shown a recall of 9.4. The refined Soundex algorithm shows the next highest recall of 8.5. The proposed algorithm has shown the highest recall of 9.4. The precision of the existing algorithms as well as the proposed algorithm, has been calculated. The proposed algorithm shows a precision of 9.5. Metaphone shows the second-highest precision of 8.8. The proposed algorithm has shown the highest precision of 9.5. The time complexity is the computational complexity, i.e., the amount of time taken by the algorithm to run on a computer. It directly depends on the number of operations performed by the algorithm. It is assumed that each basic operation takes the same time. In the proposed algorithm, the data is input in the form of a string. The proposed algorithm has a time complexity of O(n).

The space complexity of an algorithm is used to identify the amount of memory required by an algorithm to complete execution and output the result. It is the summation of actual memory and auxiliary memory. As in the algorithm, input is given in the form of a string, and the space complexity is O(n).

6. Conclusion

A phonetic-based methodology has been proposed to accurately extract the names from the Indian Demographic dataset.. The proposed methodology has been compared with Soundex, Refined Soundex, Daitch-Mokotoff Soundex, Nysiis, Caverphone, Caverphone1, Caverphone2, Cologne, Metaphone, Match Rating Approach Encoder and Double Metaphone for the accuracy parameter. The proposed methodology shows the highest accuracy compared to existing algorithms, achieving an accuracy of 98% and the highest precision of 9.5. Other phonetic-based algorithms, however, perform less accurately when used with Indian names, underscoring the usefulness of the suggested method in this particular situation. The proposed algorithm proves to be more accurate in fetching the Indian names that are phonetically equal but differ in spelling. A novel, efficient phonetic-based algorithm has been proposed for the regional language. Efforts have been made to eliminate redundancy in names.

In case of an emergency, similar names with different alphabetical arrangements are identified to locate a person. In the future, efforts can be made to optimise the proposed work by reducing its space and time complexity. More algorithms will be tried out to find similarities and dissimilarities in names to locate people in emergency situations.

References

[1] Donald Treiman, Yao Lu, and Yaqiang Qi, "New Approaches to Demographic Data Collection," *Chinese Sociological Review*, vol. 44, no. 3, pp. 56-92, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[2] Krishnanjan Bhattacharjee et al., "A Novel Approach of Deduplication on Indian Demographic Variation for Large Structured Data," *Intelligent Sustainable Systems, Selected Papers of WorldS4 2021*, vol. 2, pp. 345-355, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[3] Vandna Dixit Kaushik et al., "Certain Reduction Rules Useful for De-Duplication Algorithm of Indian Demographic Data," *2014 Fourth International Conference on Advanced Computing & Communication Technologies*, Rohtak, India, pp. 79-84, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[4] Walter Santos et al., "A Scalable Parallel Deduplication Algorithm," *19th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD'07)*, Gramado, Brazil, pp. 79-86, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[5] Manogar Ellappan, and S. Abirami, "A Study on Data Deduplication Techniques for Optimized Storage," *2014 Sixth International Conference on Advanced Computing (ICoAC)*, Chennai, India, pp. 161-166, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[6] Io Blair-Freese, "Geo-Referenced Infrastructure and Demographic Data for Development," *2019 IEEE Global Humanitarian Technology Conference (GHTC)*, Seattle, WA, USA, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[7] Auntré D. Hamp et al., "Enhancing the ATra Black Box Matching Algorithm: Use of All Names for Deduplication Across Jurisdictions," *Public Health Reports*, vol. 138, no. 1, pp. 54-61. 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Olugbenga Iyiola, and Monika Akbar, "Demographic Data-Driven Deprivation Index for Predicting Chronic Diseases," *2021 IEEE International Conference on Big Data (Big Data)*, Orlando, FL, USA, pp. 4277-4286, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Manan Chawda, Rutuja Rane, and Srikanth Giri, "Demographic Progress Analysis of Census Data Using Data Mining," *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, Coimbatore, India, pp. 1894-1897, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[10] Jennifer Ferreira, *Demographic Data*, Encyclopedia of Big Data, pp. 1-4, Springer, Cham, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[11] Zhiyuan Tang et al., "Phonetic Temporal Neural Model for Language Identification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 134-144, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[12] Yi Wang, et al., "Deep Learning-Based Socio-Demographic Information Identification From Smart Meter Data," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 2593-2602, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[13] Ivan Amón, Francisco Moreno, and Jaime Echeverri, "Phonetic Algorithm to Detect Duplicate Text Strings in Spanish," *Engineering Magazine, University of Medellín*, vol. 11, no. 20, pp.127-138, 2012. [Google Scholar]

[14] Aditya Jain, Gandhar Kulkarni, and Vraj Shah, "Natural Language Processing," *International Journal of Computer Sciences and Engineering*, vol. 6, no. 1, pp. 161-167, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[15] Qinlu He, Zhanhuai Li, and Xiao Zhang, "Data Deduplication Techniques," *2010 International Conference on Future Information Technology and Management Engineering*, Changzhou, pp. 430-433, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[16] Tirapathi Reddy Burramukku, U. Ramya, and M.V.P. Chandra Sekhar, "A Comparative Study on Data Deduplication Techniques in Cloud Storage," *International Journal of Pharmacy & Technology*, vol. 8, no. 3, pp. 18521-18530, 2016. [Google Scholar]

[17] Zhengbing Hu, Volodymyr Leonidovich Buriachok, and Volodymyr Sokolov, "Deduplication Method for Ukrainian Last Names, Medicinal Names, and Toponyms Based on Metaphone Phonetic Algorithm," *Advances in Computer Science for Engineering and Education III*, vol. 1247, pp. 518-533, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[18] Vandana Dixit Kaushik et al., "An Efficient Algorithm for De-Duplication of Demographic Data," *Intelligent Computing Technology, 8th International Conference*, Huangshan, China, vol. 7389, pp. 602-609, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[19] Ahmed Elmagarmid, Panos Ipeirotis, and Vassilios S. Verykios, "Duplicate Record Detection: A Survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 1, pp. 1-16, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[20] Peter Christen, "A Comparison of Personal Name Matching: Techniques and Practical Issues," *Sixth IEEE International Conference on Data Mining - Workshops (ICDMW'06)*, Hong Kong, China, pp. 290-294, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[21] Sigrid Norris, Jesse Pirini, and Tui Matelau, *Multimodal Analysis*, The Palgrave Handbook of Applied Linguistics Research Methodology, Palgrave Macmillan, London, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[22] Raymond Thomas, Charles Manful, and Thu Pham, "A Multimodal Analytical Method to Simultaneously Determine Monoacetyldiacylglycerols, Medium and Long Chain Triglycerides in Biological Samples during Routine Lipidomics," *ResearchSquare*, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[23] Jiansheng Wei, Junhua Zhu, and Yong Li, "Multimodal Content Defined Chunking for Data Deduplication," *FAST'14: Proceedings of the 12th USENIX conference on File and Storage Technologies*, pp. 1-2, 2014. [Google Scholar]

[24] Kay O'Halloran, Gautam Pal, and Minhao Jin, "Multimodal Approach to Analysing Big Social and News Media Data," *Discourse, Context & Media*, vol. 40, pp. 1-32, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[25] D. Holmes, and M.C. McCabe, "Improving Precision and Recall for Soundex Retrieval," *Proceedings of the0 International Conference on Information Technology: Coding and Computing*, Las Vegas, NV, USA, pp. 22-26, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[26] Matthew E. Peters et al., "Deep Contextualized Word Representations," *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, New Orleans, Louisiana, vol. 1, pp. 2227-2237, 2018. [CrossRef] [Publisher Link]

[27] David Pinto et al., "The Soundex Phonetic Algorithm Revisited for SMS Text Representation," *International Conference on Text, Speech and Dialogue*, pp. 47-55, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[28] Rima Shah, and Dheeraj Kumar Singh, "Improvement of Soundex Algorithm for Indian Language Based on Phonetic Matching," *International Journal of Computer* Science, *Engineering and Applications*, vol. 4, no. 3, pp. 31-39, 2014. [Google Scholar] [Publisher Link]

[29] B.S Harish, and R. Kasturi Rangan, "A Comprehensive Survey on Indian Regional Language Processing," *SN Applied Sciences*, vol. 2, no. 7, pp. 1-16, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[30] Laura Conde-Canencia, and Belaid Hamoum, "Deduplication Algorithms and Models for Efficient Data Storage," *2020 24th International Conference on Circuits, Systems, Communications and Computers (CSCC)*, Chania, Greece, pp. 23-28, 2020. [CrossRef] [Google Scholar] [Publisher Link]