AIP Conference Proceedings

scitation.org/journal/apc

Volume 2418 A

International Conference on Research in Sciences, Engineering & Technology

Warangal, India • 12–13 February 2021

Editors • I. Rajasri Reddy and Kommabatla Mahender

INTERNATIONAL CONFERENCES

SUMATHI REDDY
INSTITUTE OF TECHNOLOGY FOR WOMEN
Learning at its best
Affiliated to JNTUH - Approved by AICTE - Accredited by NBA
Warangal Urban, Telangana, India-506371

AIP Publishing

ICCII-2020

4th International Conference on Computational Intelligence & Informatics

Organized by

Department of Computer Science and Engineering

JNTUH College of Engineering Hyderabad

(Under TEQIP-III)

Conference: 14th - 15th Feb, 2020

VENUE

Seminar Hall, Class Room Complex, JNTU Hyderabad

www.iccii.net

4th International Conference on

Computational Intelligence and Informatics

ICCII - 2020

(Under TEQIP-III)

14th & 15th FEBRUARY, 2020

SOUVENIR

Organized by

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

JNTUH COLLEGE OF ENGINEERING HYDERABAD

AN EVALUATION OF EMOTION RECOGNITION USING MACHINE LEARNING ALGORITHMS ON DIFFERENT SPEECH DATABASES

K.Raghu[1]*, M.Sadanandam[2] and V Kamakshi Prasad[3]

[1]Research Scholar , Kakatiya University, Warangal, raghukuphd@gmail.com
[2]Asst.Professor, Kakatiya University, Warangal, sadanb4u@yahoo.co.in
[3]Professor, JNTU College of Engineering, Hydearabad, kamakshiprasad@jntuh.ac.in

The speech recognition system plays a vital role in understanding the emotions of natural language. The identification of emotions from speech is a challenging task. The performance of the speech recognition system is effects on the speech signals. The speech contains different emotions feelings. Many researchers introduced different emotion recognition techniques. However, these techniques achieved better performance but unsatisfied in identify emotion of natural languages. This paper proposed a novel speech recognition system, which identify the emotions based on the speech signals. The Mel Frequency Cepstral Coefficients (MFCC) features. On the resultant features of speech applied cross validation using the test emotions. The performance of the proposed system verify with the SVM and other two classifiers. The proposed emotion recognition system achieves better performance. The empirical results shows that the proposed system outperforms when compare with different classifiers and databases.

Key Words: Corpora, Features, LPCC, MFCC,LR,SVM,HMM,GMM.

ABOUT CJITS

Christu Jyothi Institute of Technology And Science was established in the year 1998 and is situated in 54 acres campus in Yeshwantapur, Jangaon which is about 100KMs for Hyderabad with a built up area of 21463Sq.M. The institute is affiliated to JNT University,Hyderabad and approved by AICTE, New Delhi. The college offers quality technical education and offers B. Tech in CSE, ECE, EEE, civil & MECH and M. Tech. in CSE, SE, ME & EEE and in Polytechnic ECE ,EEE,ME & CIVIL

ABOUT ICRMIET-19

The objective of ICRMIET-19 is to present the latest research and results of scientists (preferred students, post graduate Students, Research Scholars and post-doc scientists) related to Electrical, Electronics Communication Engineering and Computer Science Engineering, Mechanical Engineering, Civil Engineering, Basic Sciences, Management. The conference will feature traditional paper presentations as well as keynote speeches by prominent speakers who will focus on related state-of-the-art technologies in the areas of the conference.

Now-a-days the academia and researchers are not only pondering but also experiencing the overwhelming outcomes of interdisciplinary researches. Moreover, it has been ubiquitously encouraged by the governments, research agencies and by the academic institutions.The intent behind the multidisciplinary international conference is to provide a common platform, where academia, delegates from industry and nominees from various Government and Private Universities and Institutions can sit together, and cherish about achievements so far, as well as deliberate upon futuristic approaches along with major bottlenecks. The deliberations will not only encompass all avenues of electrical, electronics, computer science and information technology but also through spotlight on positive and inadvertent impact of modern technologies on society.

The context of the conference is to foster as well as exaggerate the research culture among academia and industry facilitated by sprinkled out ideas by exchange of the intellect during conduct of the conference.

ELSEVIER Scopus® Science Citation Index Expanded THOMSON REUTERS ज्ञान - विज्ञान विमुक्तये

Christu Jyoti Institute of Technology & Science
Yeshwanthapur,Jangaon
Telangana 506167
www.cjits.ac.in | www.icrmiet.com

ICRMIET- 19

3rd INTERNATIONAL CONFERENCE ON RESEARCH AND MODERN INNOVATIONS IN ENGINEERING & TECHNOLOGY

LABTECH INNOVATIONS

PROCEEDING ICRMIET-19

3rd INTERNATIONAL CONFERENCE ON RESEARCH AND MODERN INNOVATIONS IN ENGINEERING & TECHNOLOGY

30th - 31st JANUARY 2019

CHRISTU JYOTI INSTITUTE OF TECHNOLOGY & SCIENCE
YESHWANTHAPUR, JANGAON
TELANGANA - 506167

Organised By

CHRISTU JYOTI INSTITUTE OF TECHNOLOGY & SCIENCE
JANGAON, TELANGANA
www.cjits.ac.in

LABTECH INNOVATIONS

In association with
Labtech Innovations™
www.labtechinnovations.com

ICRMIET –2019

3rd International Conference on Research and Modern Innovations in Engineering & Technology

30th - 31st January, 2019

Organized by:

CHRISTU JYOTI INSTITUTE OF TECHNOLOGY & SCIENCE
Colombonagar, Yeshwanthapur, Jangaon, Warangal,
Telangana-506 167

In Association with

Labtech Innovations™

TITLES AND AUTHORS

3rd International Conference on Research and Modern Innovations in Engineering & Technology (ICRMIET-2019)

CJITS, Jangaon, Telangana

30th - 31st January, 2019

ABSTRACTS

Organized by:

Christu Jyothi Institute of Technology & Science,
Jangaon, Warangal-506 167, Telangana
In Association with **Labtech Innovations™**

A STUDY OF SENTIMENT ANALYSIS METHODS, TOOLS AND CHALLENGES.

Pranay Kumar BV
Deptt of Computer Science and Engineering
Christu Jyothi Institute of Technology and Science
ColomboNagar , Jangaon, Telangana State India

Dr M Sadanandam
Dept. of Computer Science and Engineering
Kakatiya University
Warangal Telangana

Abstract

Sentiment Analysis is the process of computationally identifying and categorizing opinions expressed in a piece of text especially in order to determine the writers' attitude towards a particular topic, product etc, is positive, negative or neutral. Evolution of new generation technologies and fair access to the internet in countries like India, public opinions over social media, expression of sentiment on products and services are fast and furious in present days. These opinions have great value for companies to materialize profits and understand the market for their future strategic decisions. Presently sentimental analysis has wide scope in understanding customer experience, market research and consumer insights. Other applications include social media analytics, emotion and customer predictions.

Keywords: Sentiment Analysis, classification, Feature selection, emotion detection, transfer learning, prediction, machine learning, building resources.

Random search technique to find the dialects of Telugu language

Cite as: AIP Conference Proceedings **2418**, 020018 (2022); https://doi.org/10.1063/5.0081717
Published Online: 24 May 2022

S. Shivaprasad and M. Sadanandam

View Online

Export Citation

AIP Conference Proceedings **2418**, 020018 (2022); https://doi.org/10.1063/5.0081717 **2418**, 020018

Random Search Technique to Find the Dialects of Telugu Language

S. Shivaprasad[1, 2, a)] and M. Sadanandam[3]

[1]Kakatiya University, Warangal, Telangana, India
[2]SR University, Warangal, Telangana,India
[3]KU College of Engineering and Technology (Kakatiya University), Warangal, india

[a)] Corresponding author: s.shivaprasad@sru.edu.in

Abstract. Today everything was automated and things are getting easy. Text based searching and processing is given less importance than before. To use the text-based systems also there exists a large set of algorithms with tolerable complexity. These days speech is given more importance than text. Our idea is to develop a model for searching the audio files based on the characteristics that each audio has. Based on the searching it also says where the particular audio resides in the system. For this purpose, we created standard database of Telugu language dialects i.e., Telangana, Coastal Andhra and Rayalaseema. To find out required dialects, we get the different parameters from the speech sample and compare with dataset audio features. Where we get the most parameters matched that belongs to particular dialect. By this search, method performance is increased and time complexity is reduced. We got an overall accuracy of 95% when applied for the testing samples.

Keywords: Dialects, Telangana, Andhra, Rayalaseema, Random search, Telugu language, SoX.

INTRODUCTION

Dialect is defined as the language, which is spoken by the people of an area for years. Dialect varies from language to language. Dialect recognition system increases the performance of automatic speech recognition systems. Telugu language is considered as native language for the people of Andhra, Telangana and Rayalaseema regions people. Telugu language is also spoken by some part Tamilnadu, Karnataka who are migrants from Telugu spoken regions. Telugu is designated as the classical language of India by country's government. Random search method which is a database dependent search technique which can be applicable to any language but in this paper, we considered Telugu language for our research. Now a days it is not difficult to search any text present in a group of text documents. As many algorithms that are more searching came into existence for solving the problems of text, searching anything became easy with respect to text. It is a big time taking process to convert audio into text format and sometimes it may not be possible too. Searching the audio by its filename is another aspect where sometimes the same audio may be replaced by another name by mistake. So, to search an audio file required it takes lots and lots of time to know whether it is present or not. To overcome this problem there is a process where extraction of features of audio came into existence. By the technique, we can decrease the duplicate audio that are saved with different filenames. There are no searching algorithms for audio. The technique is to search an audio based on the features a particular audio has. Many of the researchers has been working towards good dialect recognition system in order to improve the performance of the automatic speech recognition system. In this paper we considered the features of an audio that are unique for every single audio.

International Conference on Research in Sciences, Engineering & Technology
AIP Conf. Proc. 2418, 020018-1–020018-8; https://doi.org/10.1063/5.0081717
Published by AIP Publishing. 978-0-7354-4368-6/$30.00

LITERATURE SURVEY

Till now there are many researchers worked with different techniques for classifying dialects of a particular language. In [1], authors have used HMM and GMM techniques for classifying dialects of Telugu language and authors used mfcc feature extraction technique for extracting the features of the audio. In [13], classification of dialects is done by using neural fuzzy classifier to uniquely identify the vowel sound as it occurs in the acoustic speech signal more frequently. In[3], Dialect identification is done by using spectral and prosodic feature extraction for English language. SVM is used for identifying the dialects based on the feature extraction.

In[2], authors have analyzed the dialects based on the on the primitive differences between the dialects. Initially authors have experimented on spectral acoustic differences between the dialects which uses volume space analysis which is a 3D-model. In[7], user friendly prototype was developed which can be installed and used in NAO robots which can be communicated using speech. HMM-GMM models are used in this model. This prototype can further have trained to identify the dialects of different languages. In [9] Authors used MFCC and SVM. The speeches are collected from different people the class boundaries are cepstral coefficients and statistical parameters are used by support vector machine. Says SVM based dialect identification system outputs optimal results.

In[11],authors applied GMM and GMM-UBM for Assamese dialects. For identification of dialects of assamese language authors have created a dataset recorded for 13 hours and 30 minutes data of spontaneous speech. GMM-UBM model got an accuracy of 98.3% and GMM model got an accuracy of 85.7%. So authors concluded that GMM-UBM is better than GMM. In[12], authors suggested that we can group the dialects based on speech signs acoustic attributes. For identifying the dialects, authors used word level features. For the development of the elements vectors acoustic properties are extracted from the word level and for the extraction of the colossal models, SVM and tree-based XGB(xtreme Gradient boosting) group calculations were utilized to separate the speech[14].

In[6] authors have proposed a new model which contains feature level fusion of MFCC(Mel Frequency Cepstral Coefficient) and TEO(teager energy operator).Authors used SVM(Support Vector Machine) classifier is used for classification phase. Authors used Malayalam dataset for evaluation of the proposed model. Their dataset contains 4 dialects and each dialect contains 300 audio samples. The system accuracy for MFCC is 65% for TEO is 73.3% and combined system reported 78% accuracy[15,16].

PROPOSED METHODOLOGY

In this section , we are explained the database creation, proposed models used to identify the dialects and the working procedure is explained.

DIALECT DATABASE CREATION

The audios are created by taking the speech from different people who are native speakers from Andhra, Telangana, Rayalaseema regions. The audios are recorded using PRAAT tool. The audios are properly preprocessed i.e. removal of noise, elimination of unwanted backgrounds in audios and editing is done by using streaming audio editor tool. We have created a dataset having 400 audios for Andhra region, 350 audios for Rayalaseema region and 400 audios for Telangana region. The procedure followed in creation of database as shown in figure 1.

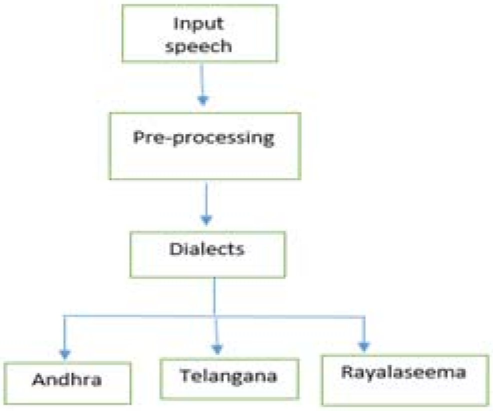


FIGURE 1. Database creation

PROPOSED MODELS

In this, we used two different methods to search the dialect of particular speech sample one is sox package and Random search method. These two are followed different approaches in searching process. Both methods provide the good accuracy in searching. Comparison of speech samples by using sox package as shown in figure2.

SEARCHING USING 'SOX' PACKAGE

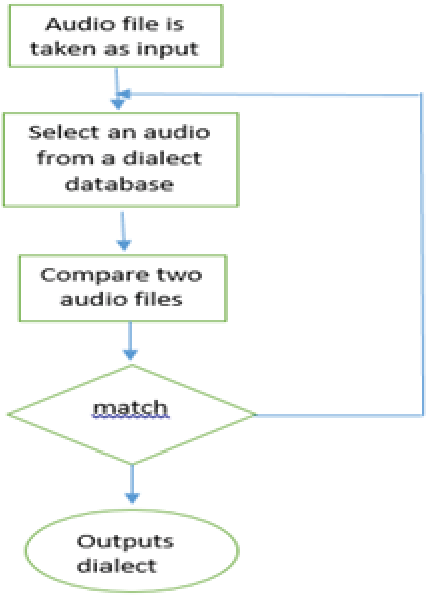

Audio file is taken as input
Select an audio from a dialect database
Compare two audio files
match
Outputs dialect

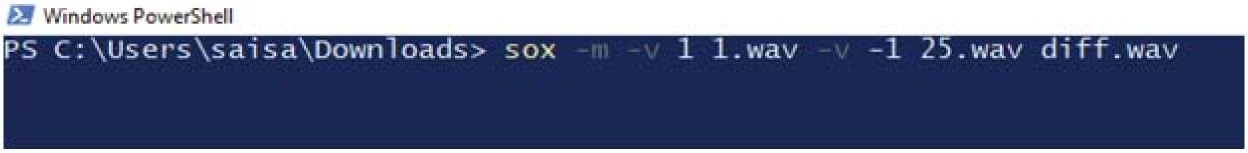
FIGURE 2. Sox procedure

Sox is a Sound Exchange package, which is a cross platform utility works on any type of operating system. Sox works with the help of command prompt. It is used to convert various formats of computer audios into other formats. Sox can also be used to add various effects on the existing files. Sox can be used for playing and recording on most of the platforms. All the functionalities of the package can be implemented by using sox keyword. Sox method is preferred when the test file is available in the same directory of the data base.

WORKING PROCEDURE

The audio file is taken as input from the user. Since we have created a database for Andhra, Telangana, Rayalaseema dialects separately we need to compare the input file with all the audio files that are present in each dialect folder to know where the input file matches. If a match is found at a particular folder then we can say that the audio belongs to that dialect. The match between two audio files can be known by seeing the image that is generated by the sox. If the output image is blank and only marked with x-axis with time and y-axis with frequency, then the two audios are said to be same. If the output generated by sox is visible spectrogram, then it is said that the two audio files that are compared are not same shown in figure 3..

i). Comparing two different audio files

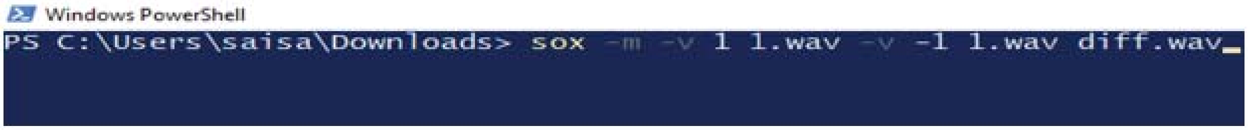


```
PS C:\Users\saisa\Downloads> sox -m -v 1 1.wav -v -1 25.wav diff.wav
```

FIGURE 3.Image showing comparing two different audio files

Let us consider the user input audio file is 1.wav, It is compared with the audio files of Andhra, Telangana and Rayalaseema dialects. In the above image presented, we are comparing two audio files named 1.wav, 25.wav.It is used to verify whether the two audio files are similar or not. In the above image m indicates mixing audio files together and v indicates volume adjustment over linear fashion. Here 1.wav is multiplied with 1 and whereas 25.wav is multiplied by 1. Implies 1.wav is represented as it is and 25.wav is inverted. The whole combination of the audio is stored in diff.wav audio file in the same directory. The multiplication factors done for the audios can be made vice versa even then it gives the same result. The audio in 1.wav is mapped with the inverted audio of 25.wav adding up at each instance gives the appropriate results and then stored in another .wav form which can be used for further processing. The below figures 4, figure 5 shows the comparison of speech samples

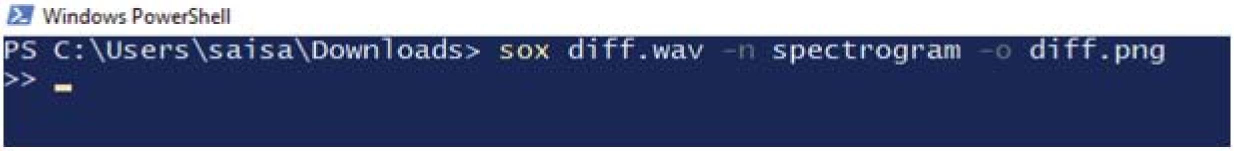
ii). Comparing two same audio files



```
PS C:\Users\saisa\Downloads> sox -m -v 1 1.wav -v -1 1.wav diff.wav_
```

FIGURE 4. Image showing comparing two same audio files

Let us consider user input audio file is 1.wav. It is compared with the audio files of Andhra, Telangana and Rayalaseema dialects. In this process we are not comparing the audio files based on the file names. In the above image presented, we are comparing two audio files named 1.wav, 1.wav.It is used to verify whether the two audio files are similar or not. In the above image –m indicates mixing audio files together and –v indicates volume adjustment over linear fashion. Here 1.wav is multiplied with 1 and whereas the second 1.wav is multiplied by -1. Implies 1.wav is represented as it is and the other 1.wav is inverted. The whole combination of the audio is stored in diff.wav audio file in the same directory. The multiplication factors done for the audios can be made vice versa even then it gives the same result. The audio in 1.wav is mapped with the inverted audio of 1.wav adding up at each instance gives the appropriate results and then stored in another .wav form which can be used for further processing.



```
PS C:\Users\saisa\Downloads> sox diff.wav -n spectrogram -o diff.png
>> _
```

FIGURE 5. Image showing how to display the output

Somehow, the procedure has been completed and it's time to represent the generated audio files by comparison. In the above image diff.wav is the audio file which contains the output generated by comparing two different audio files. -n represents the spectrogram operation should be done on complete n frames. -o represents the output of the spectrogram should be stored in the required format .i.e. image format. Finally, diff.png refers to the output image in which the spectrogram of the diff.wav is stored. The output diff.png says whether the two audio files compared are same or not. If the output image generated by sox is same then blank image with only time on x-axis and frequency on y-axis is displayed. If the output image contains visible spectrogram, then the two audio files that are compared are not same.

This procedure of comparing the audio files should be done until we found a match in a particular directory representing the dialects of the Telugu language. The searching process should be carried out in all the dialect folders.

RANDOM SEARCH TECHNIQUE

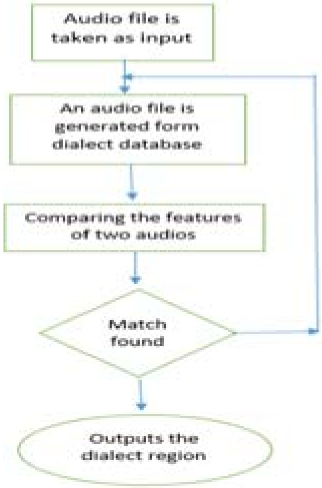

Audio file is taken as input
An audio file is generated form dialect database
Comparing the features of two audios
Match found
Outputs the dialect region

FIGURE 6. Random search procedure

From the above figure 6, the audio file is taken as input. An audio file is randomly selected from the Andhra dialects. The input audio features and randomly generated audio file features are compared if the features are similar then the input audio file is said to be of Andhra dialect. If the features of the two audios are not same then another audio file is randomly generated from another dialect dataset. The same process is continued until a match is found between the audio files. wav package in python is used to perform various operations on .wav formatted audio files. It is used for getting various audio parameters from the audio by using specific functions. By using wav package we can implement almost all the functions that are supported by the .wav audio files.wav package supports both mono sound and stereo sounds.

WORKING PROCEDURE

All words folder is the collection of audios that are present in Telangana words, Rayalaseema, Andhra final folders. The input given by the user is taken from all words folder. In each of the other three folders an audio file is taken randomly. The audio file that is given by the user is compared with the randomly selected audio files. This process takes place until there exists a matched audio file. If the user enters the audio file which is not present in the all wordsfolder then there exists an exception. If the match is found, then it prints the features of the audio by which the two audios are same and displays the output of the folder name where it is present.

Glob package: glob package is used in order to list all the file names which are present in the required directory. We can list the files by their extensions also.

Random package: random package is used to select randomly from a group.

Choice ( ):choice() is a function of random package .It returns the randomly selected audio data from the folder.

getparams() function of wav package is used for getting the characteristic features.

CRITERIA FOR COMPARING THE AUDIO FILES

Features of the audios are like number of channels in the audio, width of the audio, frame rate ,number of frames, compression type, compression name. Out of all the features compression type, compression name is same for all the audio files. The result of the getparams() is stored as a named tuple (nchannels, sampwidth, framerate, nframes, comptype, compname). The features of the input audio and randomly selected audio are compared if there is match then that is said authors are same. Finally, this searches the required audio file and says in which folder the required audio file is present. N channels represents the number of channels present in the audio i.e. stereo and mono. For mono the value of n channels is 1 and for stereo n channels value is 2.sample width indicates the width of the audio file in bytes. Framerate represents the sampling frequency at which the audio is recorded. N frames indicates the total number of audio frames created in the audio. comptype denotes compression type by default comptype is None. Compname usually it is compressed so returns the value NONE.

COMPARING DIALECT IDENTIFICATION THROUGH SOX AND RANDOM SEARCH

The dialect identification of Telugu language is performed, using sox package and Random search technique. We got an overall accuracy of 95% for both Sox and Random search. Out of 20 test samples, 19 samples are correctly classified. If we want to use the sox package, then the test sample should be placed in the each of the dialect folder and compared with all the other audio folders in the directory. Sox is also time taking procedure. So we prefer random search technique for classifying the dialects of a language.

RESULTS

Method 1: Sox Package

i).When two same audio files are compared

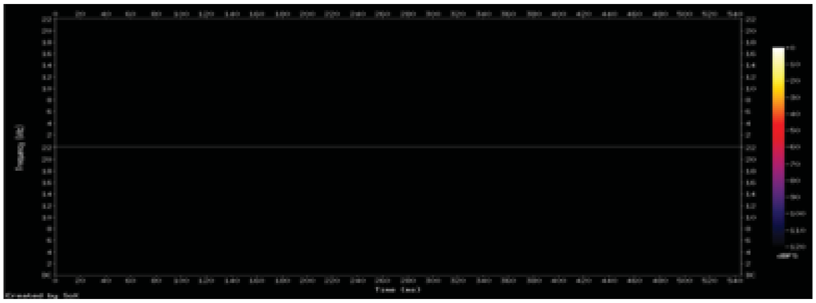


FIGURE 7. Output when two same audio files are compared

The above picture represents that the two audio files that are compared are same. Since, the 1st audio file is considered as normal and 2nd audio file is inverted, combining both the audio files at each instance of time. The summation of the frequencies at each instance results in zero. So, the output is blank. The spectrogram is represented by considering time on x-axis and frequency of the audio on y-axis.

ii). When two different audio files are compared

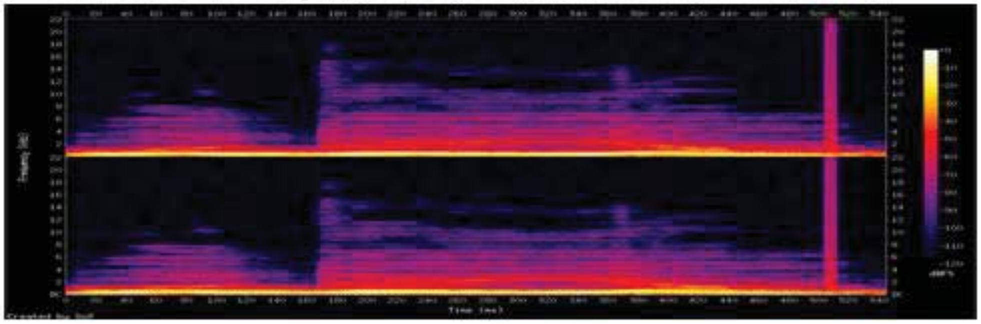
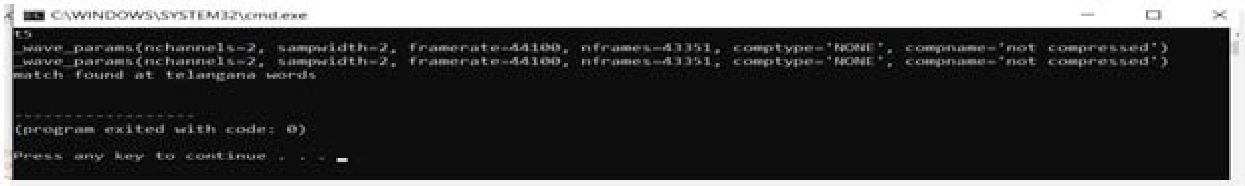


FIGURE 8. Output when two different audio files are compared

The above picture represents that the two audio files that are compared are different. Since, the audio files considered are different the output spectrogram represents the combined frequencies of the of the two audios with starting time as same but, 1st audio is considered as normal and the 2nd audio is inverted. The spectrogram is constructed by taking time on x-axis and frequency on y-axis.

Method 2

i). When input file name is present in the all words folder

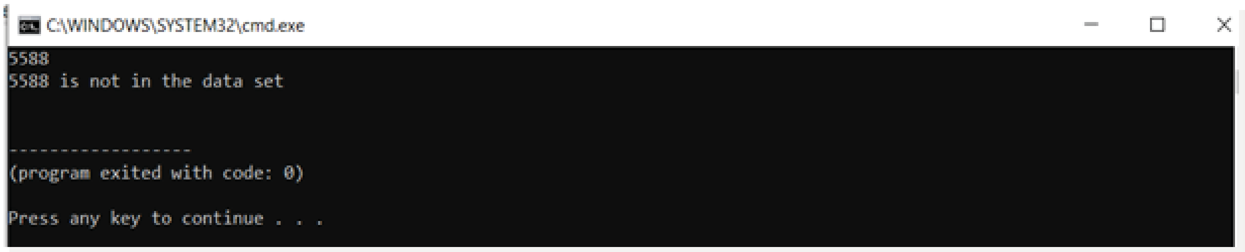


```
t5
_wave_params(nchannels=2, sampwidth=2, framerate=44100, nframes=43351, comptype='NONE', compname='not compressed')
_wave_params(nchannels=2, sampwidth=2, framerate=44100, nframes=43351, comptype='NONE', compname='not compressed')
match found at telangana words


------------------
(program exited with code: 0)

Press any key to continue . . .
```

FIGURE 9. Output when input audio is present in database

The input given by the user is t5 it searches in the all words folder for t5.wav. Since, t5.wav is present in the folder. In each of the three folders an audio is selected randomly and compared with the features of the t5.wav audio. If the tuples generated by getparams() is same then we can say the two audios are same.

ii). When input file name is not present in all words folder



```
5588
5588 is not in the data set


------------------
(program exited with code: 0)

Press any key to continue . . .
```

FIGURE 10. Output when input audio is not present in the dataset

The input given by the user is 5588, which is not available in all words folder. Since the audio is not present in the folder, it cannot search for it. Therefore, it displays 5588 is not in the data set.

CONCLUSION

In this paper, we identified the regional dialect of Telugu language using SOX and Random Search methods. These two models can be used for searching an audio file based on the feature extraction technique to identify the dialects of Telugu language. If the size of the dataset is high, then SOX will take more time when compared to Random Search. Sox method needs input parameters every time so takes some more time than random search. However, in both the models the result is accurate. We got an accuracy of 95% for both the methods. It can be further improved by taking the audio input from the user rather than audio file, which makes the searching process for audio files can be fulfilled. It can be further improved for searching a word in an audio file.

REFERENCES

1. S.ShivaprsadM.Sadanandam "Identification of regional dialects of Telugu language using text independent speech processing models" International Journal of Speech Technology Vol23 issue1 2020.
2. Mehrabani, Mahnoosh, and John H. L. Hansen. "Automatic analysis of dialect/ language sets", International Journal of Speech Technology, 2015.
3. Nagaratna B. Chittaragi, AmbareeshPrakash & Shashidhar G. Koolagudi Dialect Identification Using Spectral and Prosodic Features on Single and Ensemble Classifiers , Computer Engineering and Computer Science, 2018. DOI: 10.1007/s13369-017-2941-0
4. Saud Khan, Haider Ali, Khalil Ullah. "Pashto language dialect recognition using mel frequency cepstral coefficient and support vector machines", International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT), 2017.
5. ImeneGuellil, FaicalAzouaou. "Arabic Dialect Identification with an Unsupervised Learning (Based on a Lexicon). Application Case: ALGERIAN Dialect", IEEE Intl Conference on Computational Science and Engineering, DCABES 2016.
6. V VSreeraj, Rajeev Rajan. "Automatic dialect recognition using feature fusion", International Conference on Trends in Electronics and Informatics (ICEI), 2017.
7. Ming CHEN1 , Lujia WANG2, Cheng-zhong XU3 "A Novel Approach of System Design for Dialect Speech Interaction with NAO Robot", ICAR 2017
8. Laszlo Czap, Lu Zhao. "Phonetic aspects of Chinese Shaanxi Xi'an dialect", IEEE International Conference on Cognitive Info communications (CogInfoCom),2017.
9. Saud Khan, Haider Ali, Khalil Ullah. "Pashto language dialect recognition using mel frequency cepstral coefficient and support vector machines", International Conference on Innovations in Electrical Engineering and Computational Technologies(ICIEECT), 2017.
10. Mehrabani, Mahnoosh, and John H. L. Hansen. "Automatic analysis of dialect/language sets", International Journal of Speech Technology, 2015.
11. Tanvira Ismail and L. Joyprakash Singh "Dialect Identification of Assamese Language using Spectral Features", Indian Journal of Science and Technology, May 2017
12. Nagaratna B. Chittaragi*, Shashidhar G. Koolagudi "Acoustic Features based Word Level Dialect Classification using SVM and Ensemble Methods", 10th International Conference on Contemporary Computing (IC3), 10-12 August 2017, Noida.
13. Sarma, M et.al Dialect Identification from Assamese speech using prosodic features and a neuro fuzzy classifier, International Conference on Signal Processing and Integrated Networks (SPIN), pp. 127–132, 2016.
14. S. Magesh Kumar, V. AuxiliaOsvin Nancy, A Balasundaram, SeenaNaikkorra, D Kothandaraman and E Sudarshan, Innovative Task Scheduling Algorithm in Cloud Computing, IOP Conference Series: Materials Science and Engineering, 2020, 981(2), 022023
15. Kothandaraman D, A Balasundaram, SeenaNaikKorra, E Sudarshan, B Vijaykumar, "Enhancing dull images using discrete wavelet families and fuzzy", IOP Conference Series: Materials Science and Engineering, vol.981, issue.2,2020, pp.02220.
16. Balasundaram A, S Ashokkumar, D Kothandaraman, E Sudarshan, A Harshaverdhan, "Computer vision based fatigue detection using facial parameters", IOP Conference Series: Materials Science and Engineering, vol. 981, issue 2, 2020, pp. 022005.

Advances in Intelligent Systems and Computing 1353

Marcin Paprzycki · Sabu M. Thampi · Sushmita Mitra · Ljiljana Trajkovic · El-Sayed M. El-Alfy *Editors*

Intelligent Systems, Technologies and Applications

Proceedings of Sixth ISTA 2020, India

Springer

Marcin Paprzycki · Sabu M. Thampi ·
Sushmita Mitra · Ljiljana Trajkovic ·
El-Sayed M. El-Alfy
Editors

Intelligent Systems, Technologies and Applications

Proceedings of Sixth ISTA 2020, India

Springer

Contents

MapReduce-Driven Rough Set Fuzzy Classification Rule Generation for Big Data Processing

Hanumanthu Bhukya and M. Sadanandam

Abstract The term “big data” has become one of the essential research discussions nowadays. Due to a large amount of data availability and data processing nowadays, the topic of data science and big data becomes of prominent interest in the present research. Big data applications can be accomplished through the MapReduce programming model because these are mostly concerning scalability. The MapReduce models are intended to categorize data into various groups that are processed in parallel and whose outcome gathered to offer a single solution. There are numerous incremental models introduced by various authors to analyze and extract data from vast data sources. But, the large amount of data and diversity of the data sources there is a necessity for instant intelligent response pretense a severe problem to the current learning algorithms. Various classification models modified to this new framework, and this paper proposes a rough set fuzzy classification rule generation algorithm (RS-FCRG) to present attractive results with a MapReduce model for big data. This algorithm achieves an interpretable pattern that can handle massive data. It provides a significant accuracy with better execution time because the algorithm applies the MapReduce programming model in the Hadoop platform; it is one of the most beneficial frameworks to dispense with significant collections of data. The experiment takes on the UCI census (KDD) info dataset. The experimental results show that the proposed algorithm gets high accuracy with 95% when compared with the chi-FRBCS-Bigdata-Max algorithm.

Keywords Big data · Apache spark · Fuzzy rule-based classification system · MapReduce · Rule generation process

H. Bhukya (✉) · M. Sadanandam
Department of Computer & Science Engineering, KUCE&T, Kakatiya University, Warangal, Telangana, India

M. Sadanandam
e-mail: msadanandam@kakatiya.ac.in

M. Paprzycki et al. (eds.), *Intelligent Systems, Technologies and Applications*,
Advances in Intelligent Systems and Computing 1353,
https://doi.org/10.1007/978-981-16-0730-1_6

1 Introduction

One of the unusual big data in recent years has been known across the information technology (IT) industry as big data. The term “big data” means the research and processing of large record repositories, which traditional statistical analysis and management structures incapable of distributing [1]. This method can be recognized in different situations such as web pages, mobile devices, social networks, sensor networks, and multimedia data. With the comfort of obtaining additional information, monitoring, and information extraction procedures should be employed, and correct additional information should be collected [2]. However, traditional techniques and designs commonly used to extract information that cannot process datasets of this length [3]. Since the acquired traditional learning systems need structures that must be modified by following the advice of current guidelines that can deal with massive information, even effectively maintaining its impending capacity.

The fuzzy rule-based classification (FRBC) [4] is one of the processes to tackle big data. It is a fantastic tool available for widespread pattern recognition and mining. They can provide excellent accurate results while presenting an explainable design to the end-user by using some semantic labels. One of the challenges that make it difficult to extract valuable information from vital records is the uncertainty associated with the scope and validity inherent in big data. FRBCS genuinely deals with trade, uncertainty, or confusion, which makes it an exciting technology for handling great information, where it can reap your fundamental doubts. In the case of matters with extensive records, a variety of conditions and features are provided regularly. FRBCs reduce their performance in these places as the search area is significantly increased. This complex of growth begins the learning method from technology with problems of scalability or complexity that can end up with non-interpretable standards [5]. To this end situation, many methods have tried to increase complex systems parallel with powerful applications; however, they can focus on reducing processing time while maintaining accuracy. They cannot handle large amounts of records.

Fuzzy rule-based systems were used distinctly to control problems in large datasets [6]. One of the vital features based on the fuzzy rule-based method is their awareness because the fuzzy rule is linguistically explainable. More recently, existing systems have been applied mainly to strongly disorganize the fuzzy rules classification problems [7]. The primary way to plan FRBCS is to automate blur patterns of numerical information. Therefore, the rule-based machine to regularly arrange difficulties was a problematic part.

In many investigations, complex fuzzy sets were generated and modified using default input data for a rule-based system mainly to increase the accuracy of FRBCS classification. As explained in [8], it is possible to observe the change of association functions for predecessor fuzzy sets by weight designations, since obtaining organic properties can further degrade FRBCS interpretation potential.

The different modules of the fuzzy rule-based system (FRBS) are the rule base, fuzzification module, inference module, database, and the de-fuzzification module. Fuzzy rules are included in the rule base. The elements of the fuzzy inference system

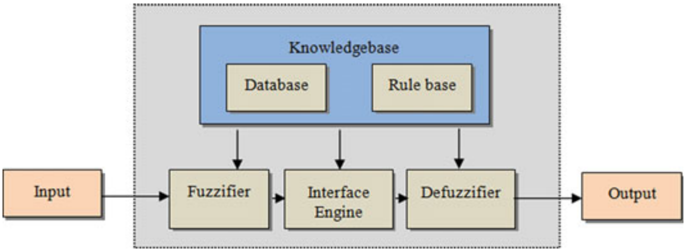


Fig. 1 Fuzzy inference system

are shown in Fig. 1 Sivanandam et al. [9]. The rule base includes fuzzy rules. The database comprises of the association functions used in the fuzzy inference system. The inference module presents the inferencing based on rules. Fuzzifier transforms crisp inputs into linguistic values. Defuzzifier converts the fuzzy score into crisp output.

In this paper, we suggested a rough set fuzzy classification rule generation algorithm (RS-FCRG) using MapReduce. It is capable of extending an interpretable version while maintaining competitive predictive accuracy within the large set of information, which has been shown as the Chi-FRBCS-BigData algorithm. This technique is wholly based on Chi et al. [10], the classical RS-FCRG method, which has been modified to deal with significant events following MapReduce technology, which varies in the “Reduce” performance and is compared to investigate how they handle extensive records.

2 Related Work

Various works discussed how a fuzzy classification rule generation could be created from very large datasets.

A set of fuzzy rules was achieved by the use of some design for the classification system. A fuzzy rule-based system and genetic algorithm were the two fundamental tools for the fuzzy rule method. Garrido et al. [11] were proposed an FRBS development model to manage difficulties in fuzzy rule generation. In that method, the number of rules of the FRBS and the description of the linguistic labels are predestined. There were two phases did in the model construction. First, the rule base was constructed, and then the linguistic labels were optimized, which preserve the interpretability of the rules [6]. Second, the heuristic systems for rule weight analysis were analyzed, which showed how each fuzzy rule’s rule weight could be explicitly recognized in FRBCS. Those methods presented well in multi-class pattern classification difficulties with a lot of classes. The partition by fuzzy three-cornered sets was a homogenous fuzzy divider for an unlimited number of training patterns with the period. The association degrees for fuzzy rules are calculated, and then rule weight is determined [12].

One of the currently developed algorithms was Chi-FRBCS-BigData, and a linguistic fuzzy rule-based classification system was recommended to deal with big data. The suggested method was used in the MapReduce framework, and it has been designed in two separate versions Chi-FRBCS-BigData-Ave and chi-FRBCS-BigData-Max. Chi-FRBCS-BigData-Max research for the rules with the same precursor and determine the rule with the highest rule weight. Chi-FRBCSBigData-Ave the rule weight the rules with the same consequents are collected, and the average is calculated. The highest average rule weight is chosen [10]. Feature selection uses the ensemble to find the most accurate features [11]. There are two main classification techniques, supervised and unsupervised. The supervised classification methods are decision tree and support vector machine. The data mining method to form a classification model is CRISP-DM. A decision tree is a tool used here to generate classification rules. Some steps build the classification model [12]. First, the planning is done to get an idea, and then the data is collected and understand by some questionnaires, and the information is prepared to build a classification model using decision tree the algorithm which gives the appropriate result.

One of the most common forms of big data handling today is MapReduce, which is a new version of distributed programming that organizes calculation in two main processes: the map feature that is obtained to separate the individual dataset and address each sub-problem independently, the element boundary that collects and collects the effects of the map function.

Many researchers have started to express well-known machine learning strategies in distributed computing models that include MapReduce to overcome these challenges. This methodology quickly became very popular due to the development of open-source frameworks like Apache Hadoop2 and Apache Spark3. In recent years, several FRBCs have been introduced based on Hadoop or Spark. Although there has been a significant development, the most extreme contributions do not reach today's outcomes in terms of accuracy and interpretability. Some of them implement many near-improvements or study tactics to get an approximate global response.

2.1 Big Data and the MapReduce Programming Model

The significant period of big data is associated with exponential prosperity in generating records that the region has taken in recent years. It also generated a great distraction because of the capabilities within the development of processing data and knowledge extraction. Incredible statistics is an excellent time for combining all the large and complex data so that it will be difficult to process or check the use of traditional software tools or data processing applications. Initially, this idea was defined as a 3 V model, precisely volume, velocity, and variety.

Volume: This characteristic refers to the huge amounts of data that need to be processed to obtain helpful information.

Velocity: These items indicate that statistical treatment packages should be able to reap effects at an economic time.

Variety: This feature suggests that data can be presented in a pair of codecs: structured and unstructured, such as textual content, digital records, or multimedia, among others. Big data problems occur in various areas and sectors, such as commercial, economic and commercial sports, public administrations, national protection, or investigations.

2.2 Apache Hadoop and Apache Spark

In recent years, distributed computing has become very familiar with machine learning systems that acquire network in the open-source frameworks such as Apache Spark8 and Apache Hadoop. These structures offer a transparent exchange machine that allows the consumer to understand the most honest information processing. Hadoop Media includes a distributed reporting device that relies on the Google file system known as the Hadoop Distributed File System (HDFS) and MapReduce model implementation. The MapReduce model extension is presented as a spark.

Spark is advanced in resilient distributed dataSets (RDD), describing divided records and data transformation. The overall performance of a user-described algorithm with a level arrangement consists of several changes divided into jobs.

This information slip (Fig. 2) allows the consumer to run an unlimited number of MapReduce functions on the same important program and supports a much broader type of algorithms and strategies from Hadoop.

The implementation of user-defined algorithms involves a series of stages consisting of some adjustments that can be divided into tasks. The level consists of the most efficient variations that do not require any shuffling/splitting operations (e.g., map and filter operations).

This scrolling of records (Fig. 2) allows the user to run an unlimited range of MapReduce tasks on the same main program, helping with a much broader type of algorithms and strategies than Hadoop.

The main disadvantage of the above system is that it contains all the attributes to be completed so that the duration of the general rule is large, indicating the complexity and accuracy that are performed that it does not reach. To this end, the option is

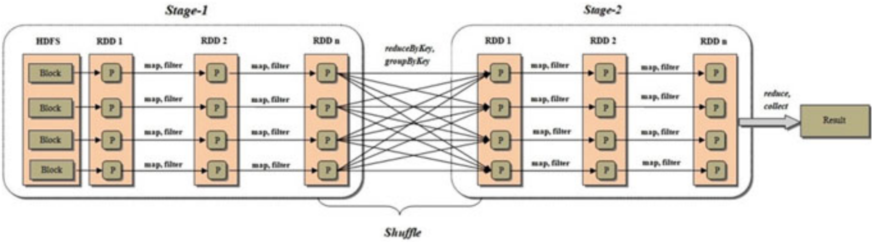


Fig. 2 Spark's data flow (P = Partition)

combined with RS-FCRG on our proposed device. The similar work with only rough sets was carried out in the following papers which could be extended with fuzzy logic for better performance [13, 14].

3 Proposed Rough Set Fuzzy Classification Rule Generation Algorithm for Big Data

Fuzzy classification rules are more expressive and allow setting the conditions of the most natural described. It is also more acceptable, as it allows symbolic information to be formulated in a natural way using linguistic terms.

Fuzzy sets represent these terms included by fuzzy rules. The members of the universe of discourse belong to a fuzzy set of certain stages of the membership, which can be described by the associated membership feature. So, fuzzy sets are usually used to address the constraints of accurate (excessive) representations by providing support for vagueness, uncertainty, ambiguity in human knowledge.

In this section, we will propose two variants of a linguistic FRBCS that control big data. First, we offer some comments associated with FRBCSs and the fuzzy learning algorithm that has been utilized in this work. Then, we will describe how this method is adapted for big data using a MapReduce system transformed to create two alternatives that will present several classification results.

3.1 Definition of Fuzzy Rule-Based Classifier

For fuzzy classification rule generation, different methods have been introduced. Let us understand that we have n training patterns $X_p = \left(x_{p1}, \ldots, x_{pn}\right)$, $p = 1, 2, \ldots, m$ from N various classes where X_p is an n-dimensional vector of characteristics in which x_{pi} is the ith attribute value of the pth training pattern ($i = 1, 2, \ldots, n$). For our N - class, n-dimensional classification difficulty, we use fuzzy "if–then" rules of the form below:

$$\text{Rule} R_q : \text{if } x_1 \text{ is } A_{q1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{qn} \text{ then class } C_q \text{ with } CF_q \quad (1)$$

where R_q is the label of the qth fuzzy if–then rule, $X = (x_1, \ldots, x_n)$ is *an* n-dimensional vector of a pattern, A_{qi} provides a precursor fuzzy set, C_q is a class label. CF_q is the weight allocated to the qth rule.

To measure the unity degree of every training pattern X_p with the precursor element of the rule $A_q = A_{q1}, \ldots, A_{qn}$ here, using the product executive as follows.

$$\mu_{Aq}\left(x_p\right) = \mu_{Aq1}\left(x_{p1}\right).\mu_{Aq2}\left(x_{p2}\right) \ldots \mu_{Aqn}\left(x_{pn}\right), \; p = 1, 2, \ldots, m \quad (2)$$

where $\mu_{Aq1}(x_{p1})$ is the unity degree of x_{pn} with fuzzy association function A_{qn}. To define the consistent class of the qth rule C_q, we estimate the confidence grade of the association rule.

3.2 The Working Model of Proposed Fuzzy Rule-Based Classification Systems

Figure 3 shows the overall block diagram of the proposed system. Training data is given as the input data to the Mapper phase. Feature selection and RS-FCRG algorithms are implemented in the Mapper phase. The output of the Mapper will be given to the reducer which fuses the fuzzy rules from the Mapper. Now the developed classifier is tested by giving test data as input. The efficiency of the classifier is computed employing accuracy.

The accuracy level is improved by the use of feature selection and bagging. Feature selection is the process of selecting a subset of related characteristics for control in model development. Here classifier divides the training dataset into equal n datasets and this individual dataset is given for feature selection. Feature selection selects the attribute which has a high relationship with the class attribute. By this method, the length of the rules is minimized and the accuracy level is improved.

Also in this paper, we suggest a learning algorithm fashioned of three steps are described below: (1) Preprocessing and Partitioning (2) Rule generation process, (3) Evolutionary rule selection.

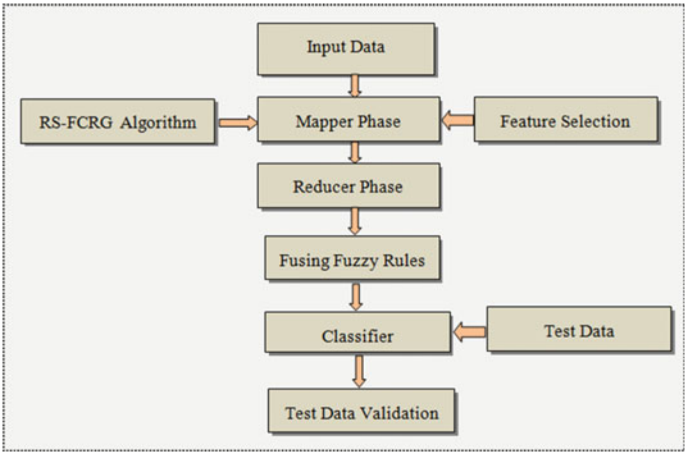


Fig. 3 Block diagram

3.2.1 Preprocessing and Partitioning

This section aims to explain fuzzy sets that meet the training data's actual appearance while keeping the number of fuzzy sets by a variable constant like low, medium, and high. This step is divided into again two parts.

Preprocessing:

The primary assignment of the training data is reorganized into a normal appearance. This transformation involves the possibility of an aggregate transform Theorem, described in Theorem 1. This theorem suggests that any dataset can be converted into a new dataset where all the variables equal a normal appearance, despite the initial distribution.

Theorem 1 *if X is a continuous random variable with cumulative distribution function (CDF).*
$F_X(x)$ *& if* $Y = F_X(x)$, *then Y is a uniform random variable on the interval* [0,1].

Proof Suppose $Y_g = (X)$ is a function of$X$ where$g$ is differentiable & strictly improving. Thus, its inverse g^{-1} uniquely exists. The CDF of Y can be derived using

$$F_Y(y) = \text{Prob}(Y \le y) = \text{Prob}\big(X \le g^{-1}(y)\big) = F_X\big(g^{-1}(y)\big) \tag{3}$$

And its density is given by

$$\begin{aligned} f_Y(y) &= \frac{d}{dy}F_Y(y) = \frac{d}{dy}F_X\big(g^{-1}(y)\big) \\ &= f_X\big(g^{-1}(y)\big).\frac{d}{dy}g^{-1}(y) \end{aligned} \tag{4}$$

This method is described as the CDF manner and provides the distribution of Y to be determined as follows

$$\begin{aligned} F_Y(y) &= \text{Prob}(Y \le y) = \text{Prob}\big(X \le F_X^{-1}(y)\big) \\ &= F_X\big(F_X^{-1}(y)\big) = y \end{aligned} \tag{5}$$

Partitioning

This step described fuzzy sets. The fuzzy sets are designed using three-sided association functions and automatically transformed behind the interval [0,1] in the recently modified space. It is justifying that the representation of each fuzzy set in the new space can be achieved by employing the contrary, improving distribution function or quantile function. In this condition, for every point describing the three-sided membership function, we would linearly combine the same value between the two nearest quintiles by determining the inverse of the linear function used to estimate the CDF.

3.2.2 Rule Generation Process

Later the fuzzy sets have been developed and preprocessed the training data, and the rule base is formed by employing a modern rule generation algorithm planned for big data. The rule generation algorithm process having two subsequent stages is motivated by the notions introduced in CHI-BD and a priori algorithms that stages are most promising item set search and Construction of fuzzy rules.

Most promising item set search

In this stage, we recognize each fuzzy set and nominal as items. When some of these elements are collectively developed in a distributed transaction, they create a set of elements (item set). To obtain the different potential combinations of potential elements, a three-step method is employed and explained below.

Discretization of the examples

Discretization of the examples: The complete sets of items presented in the training set are determined using the rule generation process of the CHI-BD algorithm. This method consists of deforming all samples by measuring the association measure of similarity of each value with all of the equal variables' fuzzy sets. It means every value followed by fuzzy sets reaches the most degree of membership. In the case of nominal costs, no estimate is transferred. The following example shows a detailed explanation, where the sequel determines which variable the disorganized fuzzy sets corresponds to.

Example 2 Given the following discretized example:

$$\text{Low}_1, \text{High}_2, \text{Medium}_3,$$

All the possible item sets are:

$$\begin{aligned}
&\{\text{Low}_1\}, \left\{\text{High}_2\right\}, \{\text{Medium}_3\},\\
&\left\{\text{Low}_1, \text{High}_2\right\}, \{\text{Low}_1, \text{Medium}_3\},\\
&\{\text{High}_2, \text{Medium}_3\},\\
&\left\{\text{Low}_1, \text{High}_2, \text{Medium}_3\right\}
\end{aligned}$$

Frequent itemsets search:

The support for each set of items is counted, and only those that are between the minimum supports are kept. In the beginning Apriori algorithm, the assistance of an itemset is when the itemset rises in the training set. Here, the support of an itemset *I* redefined as.

$$\mathrm{supp}_{\mathrm{crisp}}(I) = \frac{\mathrm{count}(I)}{N} \tag{6}$$

where $\mathrm{count}(I)$ is the original support used and the number of training samples is *N*, it will be called as $\mathrm{supp}_{\mathrm{crisp}}(I)$ to separate the crisp support utilized in this step.

Most confident item sets selection:

Another filtering process is provided between the frequent itemsets, based on the confidence determination of the itemsets. In this step, the resolution of an itemset is described as

$$\mathrm{conf}_{\mathrm{crisp}}(I) = \frac{\underset{m=1,\ldots,M}{\max}(\mathrm{countClass}(I, y_m))}{\mathrm{count}(I)} \tag{7}$$

where $\mathrm{countClass}(I, y_m)$ is number of sample counts belonging to the class y_m. In which item set *I* offered it will be called $\mathrm{conf}_{\mathrm{crisp}}(I)$ to separate this confidence from that employed in the fuzzy rules generation.

Construction of fuzzy rules

Based on the various assuring frequent itemsets obtained in the earlier step, a fuzzy rule base is designed as follows.

Conversion from item sets to candidate rules:

Each item set is transformed one or more of the candidate's rules. To this end, for a set of transmitted elements, the algorithm maintains the idea of the models in which the item set of objects appears and gets its class labels. Later, a new candidate rule was created for each of these classes.

Example 2 The following itemsets that have given the past filtering stage

$$\{\mathrm{High}_2\}, \{\mathrm{Low}_1, \mathrm{High}_2\}, \{\mathrm{Low}_1, \mathrm{High}_2, \mathrm{Medium}_3\}$$

And given the examples that have generated those itemsets:

$$\text{Low}_1, \text{High}_2, \text{Medium}_3 \rightarrow C_1$$
$$\text{Low}_1, \text{High}_2, \text{Medium}_3 \rightarrow C_2$$

we can extract the classes the item sets belongs to (C_1 and C_2) and generate the corresponding candidate rules:

IF A_2 is High THEN C_1
IF A_2 is High THEN C_2
IF A_1 is low and A_2 is High THEN C_1
IF A_1 is low and A_2 is High THEN C_2
IF A_1 is low and A_2 is High and A_3 is Medium THEN C_1
IF A_1 is low and A_2 is High and A_3 is Medium THEN C_2

Pruning and Filtering.

The rules are beginning separated based on their support and confidence. These are calculated using the following equations.

$$\text{supp}_{\text{fuzzy}}(R) = \frac{\text{matchClass + matchNotClass}}{N} \tag{8}$$

$$\text{conf}_{\text{fuzzy}}(R) = \frac{\text{matchClass}}{\text{matchClass} + \text{matchNotClass}} \tag{9}$$

Recognizing the N, matchNotClass and matchClass determined in above equations sequentially. As illustrated in the extraction of item sets, N is weighted by the cost of every class. The filtering method consists of removing those rules having support or confidence lower than a specific threshold.

3.2.3 Evolutionary Rule Selection

While the rule base has been achieved, a rule selection approach is used to make a compact and particular design. To this end, we perform the CHC evolutionary algorithm (EA) due to its strength to dispense with mixed search fields and the excellent results achieved by this EA algorithm in state-of-the-art FRBCSs like FARC-HD or IVTURS. Unlike the different approaches that offer the use of CHC

```
Algorithm:
Function: generate_rule_base(T, R)
IInput: A pre processed training set T R containing N labeled samples $x_i$
Output: A rule base RB.

Begin
    1. Search for the most promising itemsets
    1.1 Discretization of the samples
    $T\,R_d \leftarrow TR.Map\ (x_i \leftarrow discretize\ (x_i))$
    $Itemsets \leftarrow T\,R_d.Map(x_i^d \leftarrow extract\_itemsets(x_i^d))$
    1.2 Search for frequent itemsets
    $SuppConf \leftarrow Itemsets$
    $reduceByKey(itemset \leftarrow support\_and\_confidence(itemset)$
    $Itemsets_{Freq} \leftarrow SuppConf.fileter(is\_frequent(itemset))$
    1.3 Selection of the most confident itemsets
    $Itemsets_{Conf} \leftarrow Itemsets_{Freq}.filter(is\_confident(itemset)$
    $Itemsets_{Prom} \leftarrow distributed\_pruning(Itemsets_{Conf})$
    2. Construction of fuzzy rules
    2.1 Conversion from itemsets to candidate rules
    $Rules_{cand} \leftarrow Itemsets_{prom}.map(itemset \leftarrow rule(itemset))$
    $Rules_{broad} \leftarrow broadcast(Rules_{cand}.collect())$
    2.2 Computation of rule weights and conflict resolution
    $matchings \leftarrow T\,R_d.map(x_i \leftarrow matching(Rules_{broad}))$
    $SuppConfWght \leftarrow matchings.reduceByKey(rule \leftarrow$
        $support\_confidence\_weight(rule))$
    $Rules_{no\_conflicts} \leftarrow SuppConfw\ ght.map(rule \leftarrow$
        $resolve\_conflicts(rule))$
    2.3 Filtering and pruning
    $Rules_{freq} \leftarrow Rules_{no\_conflicts}$
    $filter(rule \leftarrow is\_frequent(rule))$
    $Rules_{conf} \leftarrow Rules_{freq}.filter(rule \leftarrow is\_confident(rule))$
    $Rules \leftarrow distributed\_pruning(Rules_{conf})$
    RETURN $build\_rule\_base(Rules)$
```

The experimental study conducted shows that our new method can deal with big data problems that are acquiring the same results precisely because the original set

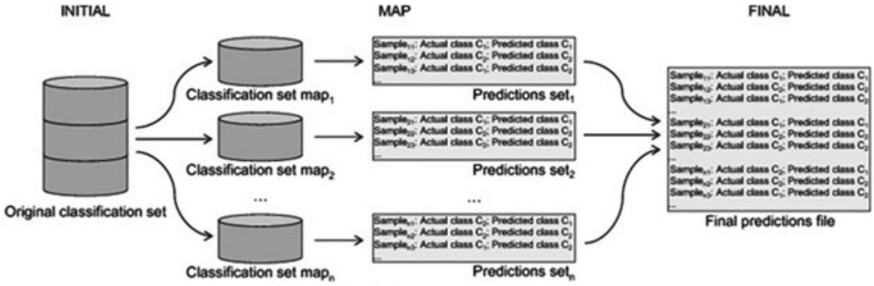

INITIAL
MAP
FINAL
Original classification set
Classification set map$_1$
Classification set map$_2$
...
Classification set map$_n$
Sample$_{11}$: Actual class C$_1$; Predicted class C$_1$
Sample$_{12}$: Actual class C$_2$; Predicted class C$_2$
Sample$_{13}$: Actual class C$_1$; Predicted class C$_2$
Predictions set$_1$
Sample$_{21}$: Actual class C$_1$; Predicted class C$_1$
Sample$_{22}$: Actual class C$_2$; Predicted class C$_2$
Sample$_{23}$: Actual class C$_2$; Predicted class C$_2$
Predictions set$_2$
...
Sample$_{n1}$: Actual class C$_2$; Predicted class C$_1$
Sample$_{n2}$: Actual class C$_2$; Predicted class C$_2$
Sample$_{n3}$: Actual class C$_1$; Predicted class C$_2$
Predictions set$_n$
Sample$_{11}$: Actual class C$_1$; Predicted class C$_1$
Sample$_{12}$: Actual class C$_2$; Predicted class C$_2$
Sample$_{13}$: Actual class C$_1$; Predicted class C$_2$
Sample$_{21}$: Actual class C$_1$; Predicted class C$_1$
Sample$_{22}$: Actual class C$_2$; Predicted class C$_2$
Sample$_{23}$: Actual class C$_2$; Predicted class C$_2$
Sample$_{n1}$: Actual class C$_2$; Predicted class C$_1$
Sample$_{n2}$: Actual class C$_2$; Predicted class C$_2$
Sample$_{n3}$: Actual class C$_1$; Predicted class C$_2$
Final predictions file

Fig. 4 Map classification sets prediction

of Chi algorithm (if they can be achieved in big data problems) regardless of the set of nodes used for execution, not like CHIBDLocal. Also, the acquired run times show that CHIBDGlobal is significantly faster than CHIBDLocal when it comes to big datasets (Fig. 4).

4 Experimental Study

This research intends to investigate the quality of the RS-FCRG Bigdata algorithm in the big data situation. For this, we will analyze the problems from the UCI dataset repository. To test the performance of the suggested approach, we have analyzed the results achieved by the RS-FCRG approach so that we can compare their behavior concerning the chosen big data difficulties.

4.1 Dataset

To evaluate the performance of the proposed method, we used the UCI Census (KDD) info dataset. The RS-FCRG-Bigdata approach is tested on UCI Census (KDD) info dataset. The data contains 41 demographic variables related to employment. The weight of the example indicates the number of people in the community represented by each record due to stratified sampling. The original table contains 199,523 rows and 42 columns. An additional column edu_year has been added to assist in the study (Table 1).

Table 1 Dataset characteristics represent

Dataset	Attribute features	No. of records	No. of attributes	No. of rows	No. of columns
Multivariate	Categorical, integer	299,285	40	199,523	42

4.2 Metrics Used

The outcome of the measures depends on the results obtained True Positive (TP), True Negative (TN), False Positive (FN), and True Negative (TN).

True Positive.

True Positive (TP): The transaction cases which are not fraud and the system model has predicted as not fraud.

True Negative (TN): The transaction cases which are fraud and the system model has predicted as a fraud.

False Positive (FN): The transaction cases which are fraud and the system model has predicted as not fraud.

True Negative (TN): The transaction cases which are not fraud and the system model has predicted as a fraud.

We have used many metrics because the set of the dataset used in this paper as markedly unbalanced. The use of the exact measurement will now be inaccurate to assess the accuracy of the method. Here we are using the below equation to calculate the accuracy of the proposed approach.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \times 100 \tag{10}$$

4.3 Results and Discussions

To demonstrate how the proposed method is capable to overcome the complexity of the design by reducing the number of final rules, we display in Table 2 the number of rules generated by each mapping method and the number of final rules, when the separate RBs generated by each map are combined. To do this, we have chosen UCI Census (KDD) info dataset with 199,523 columns, 42 rows, and 40 attributes.

Table 2 Average number of rule generated for various maps

Dataset	8 Maps	16 Maps	32 Maps	64 Maps	128 Maps
Census	34,278.0	34,341.1	34,376.5	34,392.5	34,397.3

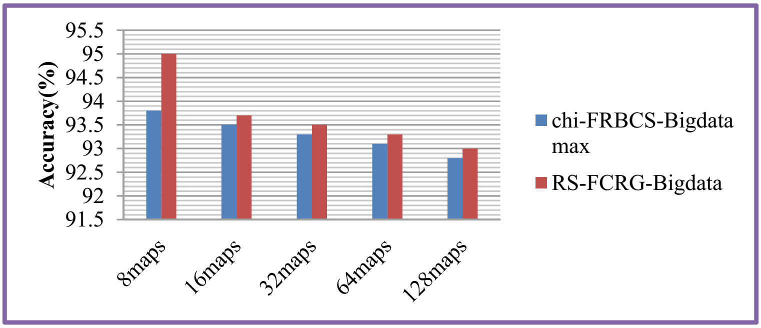


Fig. 5 Accuracy comparison between previous algorithms

In Table 2, we show the number of rules generated averagely for the proposed RS-FCRG-BigData algorithms using 8, 16, 32, 64, and 128 maps over the chosen dataset.

In Fig. 5, we represent the average results for the RSFC-RG-BigData variants, and the census datasets examined. It determines the progression of the accuracy measure when the number of maps is different. The proposed approach gets better accuracy with 95% when compared with the previous algorithm.

5 Conclusion

This paper proposed a semantic fuzzy rule-based classification algorithm for big data problems called rough set fuzzy classification rule generation algorithm (RS-FCRG). This algorithm achieved an interpretable representation that can manage big datasets, presenting good accuracy, and fast acknowledgment times. For this, the proposed approach utilized the MapReduce programming model on the Hadoop platform, and it is one of the most successful clarifications to deal with big data nowadays efficiently. This way our model distributes the calculation using the map function and then collects the results through the reduced function the experimental study took on Census-Income (KDD) dataset. The dataset is having with 199,523 rows and 42 columns, and the comparison took the chi-FRBCS-Bigdata-Max algorithm. The proposed approach got high accuracy with 95% than the previous approach.

References

1. P. Zikopoulos, C. Eaton, D. DeRoos, T. Deutsch, G. Lapis, *Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data* (McGraw-Hill, New York, 2011)
2. S. Madden, From databases to big data. IEEE Internet Comput. **16**(3), 4–6 (2012)

3. A. Sathi, *Big Data Analytics: Disruptive Technologies for Changing the Game* (MC Press, 2012)
4. H. Ishibuchi, T. Nakashima, M. Nii, *Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining* (Springer, Berlin, 2004)
5. T.P. Hong, Y.C. Lee, M.T. Wu, An effective parallel approach for genetic-fuzzy data mining. Expert Syst. Appl. **41**(2), 655–662 (2014)
6. O. Cordón, F. Herrera, A. Peregrín, Applicability of the fuzzy operators in the design of fuzzy logic controllers. Fuzzy Sets Syst. pp. 15–41 (1997)
7. Z. Chi, H. Yan, T. Pham, *Fuzzy Algorithms with Applications to Image Processing and Pattern Recognition* (World Scientific, New York, 1996).
8. H. Ishibuchi, T. Yamamoto, Rule weight specification in fuzzy rule-based classification systems. IEEE Trans. Fuzzy Syst. pp.428–435 (2005)
9. S.N. Sivanandam, S. Sumathi, S.N. Deepa, *Introduction to Fuzzy Logic Using Matlab* (Springer, Berlin, 2007)
10. Z. Chi, H. Yan, T. Pham, *Fuzzy Algorithms with Applications to Image Processing and Pattern Recognition* (World Scientific, Singapore, 1996)
11. J.F. Garrido, I.R. Ramos, A methodology for constructing fuzzy rule-based classification systems. Mathware Soft Comput. **7**, 432–475 (2010)
12. T. Yamamoto, H. Ishibuchi, Rule weight specification in fuzzy rule-based classification systems. IEEE Trans. Fuzzy Syst. pp. 428–435 (2005)
13. H. Bhukya, M. Sadanandam, Rough sets base associative classification rules extraction from big data. Int. J. Innov. Technol. Explor. Eng. **9**(1) (2019) ISSN: 2278-3075
14. H. Bhukya, M. Sadanandam, Rough sets base incremental associative classification rules generation on MapReduce framework. Int. J. Adv. Sci. Technol. **29**(05), 13218–13227 (2020)