Array 7 (2020) 100033

Contents lists available at ScienceDirect

Array

journal homepage: www.elsevier.com/journals/array/2590-0056/open-access-journal

# An efficient integration and indexing method based on feature patterns and semantic analysis for big data

Madhu Mahesh Nashipudimath [a,b,*], Subhash K. Shinde [c], Jayshree Jain [a]

[a] Department of Computer Engineering, Pacific Academic Higher Education and Research University, Udaipur, India
[b] Department of Computer Engineering, Pillai College of Engineering, New Panvel, Navi Mumbai, India
[c] Department of Computer Engineering, Lokmanya Tilak College of Engineering, Navi Mumbai, India

ARTICLE INFO

*Keywords:*
Big data
Integration
Feature patterns
Indexing
Semantic analysis

ABSTRACT

Big Data has received much attention in the multi-domain industry. In the digital and computing world, information is generated and collected at a rate that quickly exceeds the boundaries. The traditional data integration system interconnects the limited number of resources and is built with relatively stable and generally complex and time-consuming design activities. However, the rapid growth of these large data sets creates difficulties in learning heterogeneous data structures for integration and indexing. It also creates difficulty in information retrieval for the various data analysis requirements. In this paper, a probabilistic feature Patterns (PFP) approach using feature transformation and selection method is proposed for efficient data integration and utilizing the features latent semantic analysis (F-LSA) method for indexing the unsupervised multiple heterogeneous integrated cluster data sources. The PFP approach takes the advantage of the features transformation and selection mechanism to map and cluster the data for the integration, and an analysis of the data features context relation using LSA to provide the appropriate index for fast and accurate data extraction. A huge volume of BibText dataset from different publication sources are processed to evaluated to understand the effectiveness of the proposal. The analytical study and the outcome results show the improvisation in integration and indexing of the work.

## 1. Introduction

Due to the emergence of digital data and communication networks, a huge collection of repositories has become available to the public. Many tools are available to extract information [1] from the various sources, but it is essential to extract data effectively and accurately. Some researchers came up with models and discussions where integration and indexing play important role for mining the information or querying. Integrating and indexing needs an accurate learning model to associate the data [2] and have an effective result. This task turns more challenging to develop efficient data integration and indexing for relevant information when the data is in unstructured form of distribution.

Data integration is a computationally efficient and accurate method in the field of data mining for the data categorization. The purpose of data integration is to serve trusted data from a variety of sources. It is a need to handle a large data set in terms of volume, dimensions, and complexity of data features. The core objective of data integration in real-time data from various domain sources is to generate valuable and meaningful information. The conventional data integration techniques process large volume of data utilizing the process of "ETL (extract, transform and Load)".

The source of information from many heterogeneous data source generates several challenges for integrating with the conventional data integration techniques. The heterogeneity can exist at the schema level, where different data sources often describe the same domain using different representations. It can also exist at the instance level, where different sources can represent the same real-world entity in different ways. For example, the bibliographical data of academic articles represents diverse information but contextually they are related to a subject category. Many approaches for analyzing the heterogeneity data for integration are suggested in past through schema mapping [3], record linkage through object reference and entity matching. But the challenges are still lying for integrating multiple source and unstructured data objects, due to its probability of information description and diversity in the features.

The indexing refers to the organization of data according to a specific schema or plan. The indexing provides a list of tags or names to access the integrated and structured data in fast and accurate manner. Indexing is

* Corresponding author. Department of Computer Engineering, Pacific Academic Higher Education and Research University, Udaipur, India.
*E-mail address:* madhumn@mes.ac.in (M.M. Nashipudimath).

https://doi.org/10.1016/j.array.2020.100033
Received 2 June 2019; Received in revised form 22 April 2020; Accepted 4 June 2020
Available online 10 June 2020

applied for big data to perform retrieval tasks from voluminous, complex datasets with scalable, distributed storage in cloud computing. It is impractical to perform manual exploration on such records. An efficient, high-throughput indexing technique would optimize the performance of data query operations. Therefore, efficient indexing techniques are required to effectively access big data, but the diverse in data features create many challenges to index a data object accurately.

The big data growing rate in real-time makes indexing and querying is highly challenging task. The schemes of Big Data indexing [4] is to fragment the datasets according to criteria that will be used frequently in query [5] The fragments are indexed with each containing value satisfying some query predicates. This is aimed at storing the data in a more organized manner, thereby easing information retrieval. Hence a right indexing technique is essential to effectively access big data such as, semantic indexing-based approach [2] bitmap indexes [4] etc. Indexing play an important role in supporting queries on high-dimensional big data efficiently where queries with value and dimension sub-setting conditions are commonly used by researchers to find useful information from big data.

The rightful necessity of an efficient integration and indexing method for improvising the big data query processing motivates for a better idea for information retrieval in big data. The study of the feature extraction and selection suggested an effective mean for integrating unsupervised data to a meaningful group and also support in semantic indexing-based approaches.

Few works are proposed for the integration [7] and indexing [4] to simplify the data accessing. It's difficult to conclude about integration and indexing with these discussed articles due to its inappropriate integration and indexing and heterogeneity of information in the article. In this regard, the integration of bibliographical data is considered as one of the most important tasks in the area of digital library [1].

A huge volume of bibliographical data of the research articles form a big data collection from the multiple publication sources which can be used as source for integration and indexing process.

The goal of this article is provide integration approach based on PFP and indexing based on F-LSA, as a solution to the researchers and scientists for utilizing the bibliographical data. It will help for to efficient search of literature and identify latest research trends through mining the linking of articles.

The structure of this paper is as follows: Background study is discussed in Section 2. Section 3 represents the proposed integration and indexing method, Section 4 is a discussion of the experimental evaluation and results. Conclusion of the proposed work is summarized in Section 5.

## 2. Background study

Data integration and uncertainty management are central topics in the field of data quality. Data integration [3] is one of the basic activities used to improve the quality of data which is distributed among independent data sources. The traditional data integration systems are systems interconnecting a limited number of resources, which are relatively stable in time. These have been typically built with complex and time-consuming design activities.

As discussed in Ref. [6] of 2018, data integration system needs to handle uncertainty levels on the semantic mappings between the data sources and the mediated schema. It is essential for effective construction of indexing based on the keywords for data accessing queries. It is possible by tagging features for schema elements. It helps to discover mappings through understanding the meaning of tagging features. But as per work done in 2019, it has many challenges for understanding the dependable features [8] and its associations for accurate integration [10].

The number of variables or features is often increasing to describe the information in many domains for the different form of data, such as multimedia data, web data, research articles [1] etc. Sometimes, these high-dimensional data consists of many multiple values based on its observations for a feature. In fact, all the features are often not important and distinctive because they are mostly found correlated or redundant to each other [11]. It might be very noisy for the selection [12] in some scenario. The function of these high dimensions may lead to low efficiency or poor performance in conventional learning models [13]. Therefore, it is challenging to learn the high-dimensional data features which will help to improve the accuracy and comprehensibility of results. It also has challenges to eliminate the unrelated and repeating features that involve in large volumes of data, which is a need to select from a subset of the data feature.

Integration is a computationally efficient and accurate method in the field of data mining for the data categorization. It coordinates to handle a large data set in terms of volume, dimensions, and complexity of data features. Various approaches are proposed for multi-domain [6] and uncertain data [8] for efficient and accurate integration. An uncertain data objects clustering based on the probability of similarity distribution is presented in recent paper [9] of 2019. It suggests that, the difficulties of integrating the uncertain data object is due to probability distribution which arises in many situations. The problem related distribution based feature similarity can be worked out effectively by means of the feature selection [11] and the feature extraction [14] methods. It will reduce the dimension of features finding a meaningful feature set group and selection.

Feature selection method is classified as a regulation method of feature selection for the mostly termed as "supervised" or "unsupervised" feature selection. Supervised feature selection methods uses the relationship between characteristics and feature information [13] that leads to selection of the significant and relevant features [11]. But studying large volume of data in unsupervised feature selection leads to difficulty for analysis. It happens due to deficient of feature information to refer feature selection. Hence this article focuses on the issue of unsupervised feature selection.

### 2.1. Importance of feature selection

The process of identifying the suitable and effective features is an important task. It is one of the most widely used mechanism in various multi-domain analysis [6,12] and information mining [1]. Some researchers had specifically worked on possible ways of data collection too. The article [10] published in 2018 ranks different classification methods separately for hard data sets and for easy data sets. Random forest is reported as winning method. Classifiers ranking is analyzed using different datasets with varying instances and attributes but data size remains as future effort. And hence it offers new insights and motivation to develop and upgrade classification and feature selection approach using big data.

The method of selecting a variety of features for a machine learning application have been studied and proposed by T. Nguyen et al. [15]. These algorithms can be categorized as "supervised", "semi-supervised" and "unsupervised algorithms" based on the method of using the learning feature information [13]. The supervised methods are looked in the way that they are able to choose the distinctive features [16] as latest art of 2019,because the information is encoded in the differential features. Using the available feature correlation, one can understand scanty-based methods based on form of semantic clustering [17] mentioned during 2013 and model-based clustering [18] mentioned in 2019. This study suggests that data model with small number of features for a large volume of unlabeled data will generate result. It can be easily configured to construct a common cluster data set. But it may face the problem of highly unrelated data integration due to the low feature sample problem, which is challenging for the supervised learning and data management.

Selvakumar et al. [21] had worked on the implementation of three different classification techniques. These experiments of classification were carried out with a multi-dimensional healthcare dataset. This multidimensional diabetes disease dataset contained 100 observations with 7 attributes. k-Nearest Neighbor gives higher accuracy as compared

to accuracy of Binary Logistic Regression and Multilayer Perceptron. Working with high dimension, and huge sized dataset on specific suitable platform remains as future scope.

As there is no enough information about the data features and small features, supervised algorithms may unintentionally remove many specific features or fail with select features that are not relevant. Thus the semi-supervised feature selection can be developed to take advantage of the labelled and labelled data. But semi-supervised feature selection system has to guide to search for distinctive features in absence of labels. Hence it is seen as a much more difficult problem [13]. It evaluates the characteristic of interest for their ability to retain some element properties. In many real-world applications, the lack of unlabeled data and rapid accumulation of high-dimensional resources creates the challenges for the labels identification. Hence its demands for a very promising and autonomic system for unsupervised feature selection technologies to meet the integration problem.

### 2.2. Unsupervised feature selection for integration

In unsupervised feature selection based integration, information identifies a subset of the features of holding a unique group as no label is available. These groups are in accordance with the specified standard for the more challenging grouping or clustering criterion [15]. The difficulties in the unsupervised feature selection are mostly being solved using the probability model methods [9,15]. Here the features are sorted into the group of labels utilizing the "feature selection based grouping" considering the latent features variable. Venkatesh, B, and J. Anuradha [13] proposed a visual feature and user separate class features for a generative model in 2019. The purpose is to the group the high dimensional relevant data. Similarly W. Fanet al [14] proposed a probability model in 2013 for the global integration capabilities and unsupervised feature selection.

Y. Guan et al. [22] suggested a framework of reasoning of the variations of the "unsupervised non-Gaussian" method for feature selection. Unlike previous research, this work is based on the unsupervised learning model for data transformation and feature selection, the purpose to enhance the data integration task for better results. It will maintain semantic similarities along with selection of best features for distinguishing knowledge of the original data information in heterogeneous datasets. The feature Patterns is applied to each input key-value pair to generate an arbitrary number of intermediate key-value pairs. The reducer is applied to all values associated with the same intermediate key to generate output key-value pairs.

### 2.3. Techniques

The biblometric study [7] is carried out in 2018 to analyze the integrated care literature and tests the usefulness of indexing the literature within PubMed. This idea boosted to experiment with Bibtex data on big data platform. Few Literatures [5,9] guide to know and analyze multiple techniques for indexing with pros and limitations.

In this paper [19], the modified Feature vector selection (FVS) method, easy tuning version of Support vector machine(SVM) selects a small number of data points for implementation. The FVR model is also solved analytically, as in least-squared SVM. Authors had worked with twenty six imbalanced dataset and comparison is done with several SVM based methods. Result show effectiveness of suggested method. Hence proposed method is also compared with SVM in integration performance analysis section.

Yousefpouer et al. [20]. had worked in 2017 on algorithm, where TDM is a term-document matrix that is weighted using the term frequency-inverse document frequency (TF-IDF) scheme. The most relevant features were selected from the frequency-based integration to produce the final feature subset. The experimental results for different datasets show that the proposed methods can effectively improve classification performance. Hence performance analysis of proposed method is carried with respect to Term frequency.

Recently in 2019, Ritik M, and Abhaya K·S proposed text analysis to convert understood text data into meaningful data for analysis and provide sentiment analysis. Semantic based Naïve Bayesian classification algorithm [25,35] is used for text filtering and performance accuracy is calculated. This article inspired to implement Naive Bayes and further add a component of probabilistic feature Patterns (PFP) approach and perform experiment on big data. Purpose of PFP component and experimenting with big data is to analyze the performance and its contribution in indexing and recommendation as this task is yet to be experimented with.

The big data analytics on IoT based healthcare system is developed using the Random Forest Classifier (RFC) and MapReduce process [33] as recent work of 2019. The optimal attributes are chosen by using Improved Dragonfly Algorithm (IDA) from the database for the better classification.

This method gives better results for limited number and features of attribute. In addition to this, the limitation of the proposed algorithm is computationally slow because of the big database. It is clear from the recent literature review, that the Naive Bayes has wide scope to experiment with big data but is not sufficiently experimented over all with following combinations, such as.

1 Multidimensionality and multi features
2 Unstructured and complex data
3 Bibtext and huge volume information

So in this article the above identified gaps have be experimented over Bibtext data. The performances of PFP with Naive Bayes, SVM and TF approaches and presented after analysis and measure. Further performance of proposed F-LSA for indexing is justified by comparison analysis of PFP with F-LSA and TM with LSA. It is also justified for different datasets like CiteseerX, Bibtext and Cora at concluding point.

## 3. Proposed integration and indexing method

### 3.1. Problem description

Traditional learning systems are mostly studied over supervised machine learning systems. In these systems, data objects are associated with a label and the learning systems supervised the set of data to learn the feature characteristics. It will be used for the classification as shown in Fig. 1.

This learning is well adapted for single concept, but the complexity arises when the object has multiple features. So, the improvisation of traditional supervised learning to adapt the multi-features data objects is being discussed in literature [21,23]. But most of the proposed solutions are based on the features dependence learning or associating the features

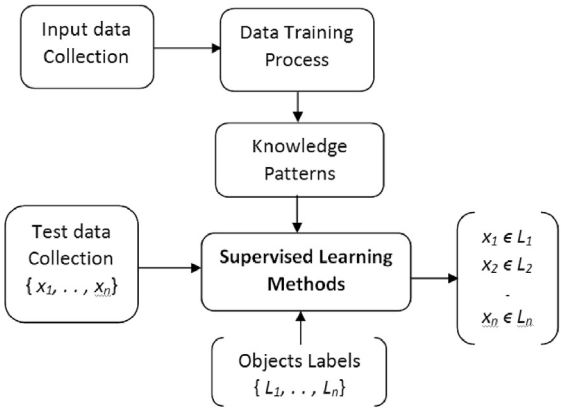


**Fig. 1.** Traditional supervised learning.

through counting their co-occurrences [24]. However, this might not be applicable for the domains where such kind of feature information is not available. In some cases, the fundamental dependency and correlations of feature are identified using association rules algorithm, but its fails to facilitate the multiple changing datasets and its features in different domains.

Target is to construct a classifier based on new feature learning system which can perform on different source of multiple features datasets as shown in Fig. 2. System provides an accurate classification for the integration and a semantic association based indexing support for the faster accessing.

The mechanism of the proposed approach works in three phases.

- Semantic data relation learning (SDRL) procedure to construct the semantic related classification patterns.
- The process of data acquisition from different sources.
- Third phase consists of two sub-tasks:
  - ✓ Data cleaning is a basic step to be performed. Feature transformation and selection method utilizing pattern knowledge follows the cleaning stage. A probabilistic feature Patterns (PFP) approach for the efficient integration is carried out through semantic classification over a Naïve Bayes classifier.
  - ✓ Feature based latent semantic analysis [26] to construct similarity index. User access module is built to present a visualization of the related BibText data which is used later for analysis mechanism.

In this article, data features are utilized to propose a probabilistic feature Patterns (PFP) approach for the efficient integration through semantically classification over a Naïve Bayes classifier [25]. This approach implements a feature transformation and feature selection method to accurately map the data through computing the principal component analysis for multi-value data. It is used for multi-value data transformation and selection for the effective integration of datasets. An indexing [27] by features based latent semantic analysis (F-LSA) method is implemented on the integrated data. In F-LSA, the data semantic association relation with the categorized data is learned and patterns are identified. These data are used as the key sets of mapping data features. It means a representation to have a similarity index to facilitate the enhancement and precise searches for big data. The detail in discussed in next section.

**Table 1**
One-feature class table.

| One-Feature Class (OFC) | Relevant Features |
|---|---|
| Aerospace | *{ f1, f2, f3, f4, f4, f5, ... }* |
| Biomedical | *{ f1, f2, f3, f4, f4, f5, ...}* |
| Health Informatics | *{ f1, f2, f3, f4, f4, f5, ...}* |
| Cloud Computing | *{ f1, f2, f3, f4, f4, f5, ...}* |
| Data Mining | *{ f1, f2, f3, f4, f4, f5, ...}* |

### 3.2. Proposed system modules

#### 3.2.1. Semantic data relation learning (SDRL)

The supervised classifier needs a precise learning mechanism to perform an accurate classification. The mechanism of SDRL provides the learned knowledge for the design of a classifier which further classifies the data for the integration. One such "One-To-k (OT-k) Label learning method" is proposed in Refs. [23]. It describes the approach of classifying data to a single label class through multidimensional features of data. It suggests that an object having two or more feature in similar can be considered for a single label class. This approach is best suitable for data classification of multidimensional features. This methodology is inherited for SDRL process.

The learning of SDRL implements two methods. The first method extracts the required feature information from the data to the relevant and specified article class which is termed as, "One-Feature Class (OFC)" as shown in Table 1. The second method finds the most k relevant features which supports for classifying through constructing a feature pattern.

To illustrate the SDRL, let's assume a set of training data sets T, that consists of various article documents for each specific OFC represented as, T = {d1,. . ., dn\}. Assume a vector V consists of all OFC. For each OFC in V, the data sets in T are processed to construct a relevant feature vector class as R. Next by utilizing the features in R of each OFC, most appealing features are found through computing feature density (Density) among the features using Eq. (1).

$$Density = \frac{\sum_{i=1}^{n} (f_n \in D)}{|D|} \tag{1}$$

The value of Density ranges between 0 and 1. In this case, "0″ is

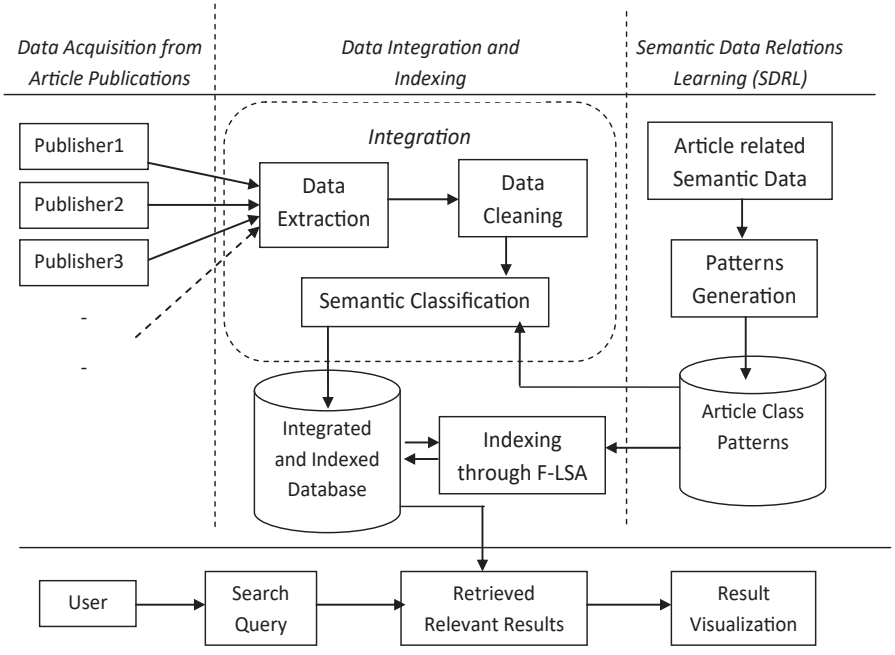


**Fig. 2.** System architecture.

considered as least relevant and "1″ as most relevant for OFC. The mechanism OFC table construction through the method of SDRL is being presented in Algorithm-1.

**Algorithm-1**
Generation of OFC relevant features

```
Input:T → Training Datasets.
V → Set of OFC
Density_Th→ Feature density threshold.
for i = 0, each class in Vloop {
  C = V[i];
//– get the training data sets of a class
  D [ ] = T [C];
  for j = 0, each data record in Dloop {
F [ ] = extractFeatures (D[j]);
j++;
  }
  s = 0;
//– construction of relevant features
  for k = 0, each features in Floop {
f = F[k];
//– Compute feature density
Density = getDensity(f);
if (Density≥Density_Th) then
R [s] = f;
s++;
end if
  }
  i++;
}
```

Most of the existing feature selection approaches with multiple features datasets suffer from information loss [4,6]. So it is essential to minimize this information loss in the SDRL. This is surmounted by data extraction through a binary association of the feature, followed by computation [11] of Information Gain(IG). It is used to find the other features which can be supportive for the data classification along with the R as presented in Fig. 3.

If the frequency of term in a set of features >1, the binary value of feature is 1 else it will be 0. The measure of IG is the ratio of the summation of feature frequency over the number of data records processed. Based on these final k-features generated, needed classification pattern combination is constructed for the article classification. It will facilitate the integration process.

### 3.2.2. Data acquisition

Data acquisition is a process to gathering data from distributed information sources. Filtering and cleaning is done and followed by storing them in scalable, big data-capable data storage. Different architectures for data acquisition have been proposed to address the different characteristics of big data. Most of data acquisition scenarios assume high-volume, high-velocity, high-variety, but low-value data. This makes adaptable and time-efficient gathering, filtering, and cleaning algorithms as a need to ensure that only the high-value fragments of the data are actually processed for the analysis.

Most of the time, demand is to obtain data from the sources which are

| D | f1 | f2 | f3 | f4 | f5 |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 1 | 0 |
| 3 | 1 | 0 | 1 | 1 | 1 |
| 4 | 1 | 1 | 0 | 0 | 0 |
| 5 | 0 | 1 | 0 | 0 | 1 |
| ∑ | 3 | 3 | 1 | 3 | 2 |
| IG | 0.6 | 0.6 | 0.2 | 0.6 | 0.4 |

Features having IG >=0.5 →

| Features |
|---|
| f1 |
| f4 |

**Fig. 3.** k-Feature Generation.

difficult to retrieve. In such cases, structured or unstructured data files delimited with comma or spaces are downloaded. It is easy to process and analyze if the files are limited but in case of high number, diverse and unstructured data makes it quite complex to process and analyze.

In this paper, data files of different technical article publishers are acquired from Information Extraction and Synthesis Laboratory (IESL). It aims to support the mining actionable knowledge from unstructured text. The gathered data containing a more than 4000 number of BibText files which have more than 6 million article records downloaded from the internet. The data records are extracted through processing each individual file and imply the data cleaning method to remove the data containing noise and filling up the missing data.

### 3.2.3. Data integration

Features are extracted to perform the feature transformation and selection for integration. The selected features are semantically associated to classify using the pattern generated through SDRL. Each data records consists of few key data fields such as author, title, keywords and abstract known as metadata of the article. The terms are extracted from the keywords and abstract field to construct a set of terms which will use for the semantic classification to classify the article class for the integration.

Semantic classification performs relevance association computation to relate the latent semantic relevancy [28]. Since, the article is technical information, the semantic-based classification [29] will compute the similarity between the set of terms information constructed to identify each article.

A “Concept of Similarity mechanism" [9] is derived to perform the association of data record. The concept of such terms and article metadata is considered to find its relevance. It associates with other elements of the article class.

Let’s consider a data record, ĐR which consists of set of terms belong to an article class as, ĐR = {t1, t2,. . . ., tn }. To find the relation association, the relation between the terms and the features of the article class is calculated. It gives frequency of term, calculation as f(x) for the dependency of this term to other terms using the equation-2.

$$f(x) = \frac{Term_t}{Z}\ ,\ Term \in\ Article_p \tag{2}$$

where, T is the term related to the terms of article class $Article_p$, and Z is the total number of extracted terms from the data records. On completion of freq(x) identification, the similarity relevancy probability as P(t|a), is computed of each term present in each record as against each article class as A, using the equation-3.

$$similarity(r : Article_p) = Prob\left[\ (term\ \cap\ Article_p)\ /\ |Article_p|\right] \tag{3}$$

Let’s assume data record, d has a phrase like “Big data and Data mining”. Using equation (2), the probability "Semantic Term Similarity" is computed in related to article class, n as shown in equation-4,

$$\text{similarity}(r : Article_p)_n = \text{Prob}\left[\ (\text{term}(\text{Big},\text{Data},\text{Mining}) \cap Article_p / |Article_p|)\ \right] \tag{4}$$

Few techniques [21] are discussed for classification. But The functionality of a Naive Bayes Classifiers [25] is completely independent from the class attribute values variations, which generally being termed as Condition-Independence. This approach is constructed based on the assumption for the simplification and generalization of class information in relation to the Naïve Bayes probability assumptions. The constructed class information symbolizes the characteristic of the class attribute relation which supports the classifier in accurate classification. It was experienced that "Bayes approach" is useful in an assured circumstances and it is extremely reliant on the hypothesis of the target classification information for the proficient outcomes. Because of high dependency, a little divergence in the supposition hypothesis constructs a set of inaccuracy in recognition. In the proposed semantic classification method,

Naive Bayes is used to compute the probability of relevance in compare to the designed OTF article class.

The probability of relevance of terms t with article class (AC) is defined as, pr(t(1→n) ∩ ac(1→k)). Based on this pr(ac1-k) obtained for each AC, the highest probable relevance AC is selected as, pr(C) to classify the record. The methodology of the modified Naïve Bayes to compute the Bayes probability similarity as β in relevance to AC terms for efficiently classifying the data sets is presented in Algorithm-2.

**Algorithm-2**
Semantic Classification Method

Input:
*T* → Set of technical terms of a data record *d*.
*AC_ Set[ x]*→ Set of article class.
*AC_Terms[ x]* → Set of terms from *AC_Set[ x]*
$\beta = 0$;//– probability similarity
C = null;//– Initial record Class
**For** $k = 0$; $k <$ *size(AC_Set [k]);* each value in *AC_Set[k]*
*AC_Terms[ x] = AC_Set[k];*
//– Compute the Bayes probability similarity
α = prob(AC_Terms [x] ∩ T);
**if** $\alpha > \beta$ **then**
$C = AC_k$;
β = α;
End If
End for
*d*, classified as, *C*.

This classified class C of the data record is used for integration. This method is evaluated extensively for analysis over a BibText data collection of different publishers and stored in a structure manner for the indexing.

### 3.2.4. Data indexing

The indexing of structured data using feature latent semantic analysis (LSA) has been widely used in many fields of natural language processing [29] as per work done in 2018, where co-occurrence features can be acquired by the transfer relations between the records. The co-occurrence information of the terms can be captured when Singular Value Decomposition (SVD) is decomposed as proposed by Refs. [30] in the year 2019.

The example shown in Table 2 and Table 3 are for data record and its feature document matrix respectively. The term weight represents word frequency of the term and the matrix is transform to a singular value reducing to a two-dimensional space.

A sample dataset consists of the titles of 9 technical articles. Terms occurring in more than one title are extracted. There are two classes of records as, "Computer Interaction (I)" and "Data Mining (M)". This dataset can be described by means of a term by document matrix where each cell entry indicates the frequency with which a term occurs in a record.

**Table 2**
The data records titles.

| Record id | Article Class | Article Title |
|---|---|---|
| S11 | I | Recognition of Head Gestures Using Hidden Markov Models |
| S12 | I | Graphical User Interfaces and Integration |
| S13 | I | Direct Manipulation for Comprehensible, Predictable and Controllable User Interfaces |
| S14 | I | A Brief Direct History of Human Computer Interaction Technology |
| S15 | I | Integration of Speech and Gesture for Multimodel Human-Computer Interaction |
| S21 | M | An Overview of Document Mining Technology and Software |
| S22 | M | Mining Sequential Patterns: Generalizations And Performance Improvements |
| S2d3 | M | Web Based Parallel/Distributed Medical Data Mining Using Software Agents |
| S24 | M | Evaluation of Sampling for Data Mining of Association Rules |
| S25 | M | Mining Association Rules with Item patterns Constraints |

The value of similarity between the terms will be 1 in Table 3 for occurrence of terms information.

The similarity of 0 is between some disintegrated terms, that means there is no or little relations. By the changes of the similarity value, one can think "user" and "interface", "interface" and "human", "user" and "human" co-occur. In LSI [31], the "user" and "human" projected to the same dimension space. Comparison of the feature matrix between the similarity matrix transformed to SVD is shown in Table 4.

The degree of similarity in features reflects the correlation between the terms. The weight value not only reflects the correlation in the features but also embodies the co-occurrence information between the features in SVD space. The similarity degree in the documents mainly depends on the number of the co-occurrence of features. In generating latent semantic space, the latent relations will be excavated because of the transitivity between the features.

Tables 4 and 5 represent the similar data records indexed to their relevance group. This will be useful for accessing the data record much faster by reducing the time complexity.

### 3.3. Data access and visualization based on PFP and F-LSA

User data access model is designed to evaluate the proposed integration and indexing mechanism. It consists of a user interface to submit a query for mining and visualization of Bibliographic data of the technical article. The search query retrieves the most relevance articles as results and its information as shown in Fig. 4 and Fig. 5.

The aim of Fig. 4 is to show the search result by proposed method. Fig. 5 shows the first page search result along with total number of pages to follow. It also shows total record count of matching through search method supported by total time taken for search operation. User can click on page centric view link provided in front of each search result.

A visualization analysis of data records in relation to the author is presented by an article centric view as shown in Fig. 5. This is detail of page link from previous page (as shown in Fig. 4). It provides details of particular search result, i.e. author names, related article title and keywords. It also gives details and reference of related article with respect to keywords.

## 4. Experiment evaluation

The experimental work was carried out over Hadoop Framework for storage the big data. Processing and analysis was performed using Java technologies. The data access visualization is evaluated through a web interface provided by Apache Tomcat Web server. The SDRL mechanism is implemented to generate the required knowledge pattern for integration. Later, the obtained data sources are cleaned and exported to Hadoop for storage.

The Java program performs the integration through semantic classification of data record with the help of knowledge pattern. After completion of integration, the indexing method F-LSA is executed to rank the integrated data group. To evaluate the improvisation in the indexing, a Data Access Visualization interface is built, where multiple queries are submitted to measure the accuracy of the outcome.

### 4.1. Data source

Data are acquired from the data files of diverse technical article publishers from Information Extraction and Synthesis Laboratory (IESL) [32]. It consists of more than 4000 number of BibText files which have more than 6 million article records.

The parameters used to retrieve data for integration is keywords of article. These keywords are further tokenized as terms of term document.

**Table 3**
Transformed feature matrix based on article terms.

| Tid | Terms | S11 | S12 | S13 | S14 | S15 | S21 | S22 | S23 | S24 | S25 | *Total* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| doc1 | Interfaces | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **2** |
| doc 2 | User | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **2** |
| doc 3 | Direct | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | **2** |
| doc t4 | Interaction | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | **2** |
| doc 5 | Computer | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | **2** |
| doc t6 | Human | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | **2** |
| doc t7 | Integration | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | **2** |
| doc 8 | Mining | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | **5** |
| doc 9 | Patterns | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | **2** |
| doc 10 | Data | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | **2** |
| doc 11 | Software | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | **2** |
| doc 12 | Association | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | **2** |
| ***Total*** | | **4** | **3** | **0** | **4** | **4** | **1** | **2** | **3** | **3** | **3** | |

**Table 4**
Feature similarity matrix truncated to two-dimension.

| | doc *1* | doc *2* | doc *3* | doc *4* | doc *5* | doc *6* | doc *7* | doc *8* | doc *9* | doc *10* | doc *11* | doc *12* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| doc1 | 0.587 | 0.581 | 0.574 | 0.581 | 0.598 | 0.598 | 0.571 | −0.547 | −0.489 | 0.225 | 0.347 | −0.403 |
| doc 2 | 0.581 | 0.594 | 0.588 | 0.912 | 1.521 | 0.487 | 0.487 | −1.214 | 0.984 | 0.287 | −0.189 | 0.142 |
| doc 3 | 0.574 | 0.588 | 0.997 | 2.104 | 3.124 | 1.124 | 1.057 | −0.657 | 0.199 | 0.471 | −0.147 | 0.440 |
| doc t4 | 0.581 | 0.912 | 2.104 | 2.011 | 3.014 | 1.351 | 1.192 | 0.214 | −0.324 | −0.214 | −0.147 | 0.247 |
| doc 5 | 0.598 | 1.521 | 3.124 | 3.014 | 0.547 | 0.547 | 0.981 | 0.124 | −0.314 | 0.145 | 0.112 | 0.263 |
| doc t6 | 0.598 | 0.487 | 1.124 | 1.351 | 0.547 | 2.124 | 0.214 | 0.941 | 0.814 | −0.214 | −0.614 | −0.547 |
| doc t7 | 0.571 | 0.487 | 1.057 | 1.192 | 0.981 | 0.214 | 0.987 | −1.021 | 0.140 | −1.021 | 0.149 | 0.127 |
| doc 8 | −0.547 | −1.214 | −0.657 | 0.214 | 0.124 | 0.941 | −1.021 | 3.147 | 1.158 | 0.981 | 1.024 | 0.924 |
| doc 9 | −0.489 | 0.984 | 0.199 | −0.324 | −0.314 | 0.814 | 0.140 | 1.158 | 1.201 | 1.095 | 0.547 | 0.458 |
| doc 10 | 0.225 | 0.287 | 0.471 | −0.214 | 0.145 | −0.214 | −1.021 | 0.981 | 1.095 | 2.021 | 3.001 | 2.014 |
| doc 11 | 0.347 | −0.189 | −0.147 | −0.147 | 0.112 | −0.614 | 0.149 | 1.024 | 0.547 | 3.001 | 1.260 | 0.902 |
| doc 12 | −0.403 | 0.142 | 0.440 | 0.247 | 0.263 | −0.547 | 0.127 | 0.924 | 0.458 | 2.014 | 0.902 | 1.225 |

**Table 5**
Similarity matrix between the data records.

| | *S11* | *S12* | *S13* | *S14* | *S15* | *S21* | *S2d2* | *S23* | *S24* | *S25* |
|---|---|---|---|---|---|---|---|---|---|---|
| *S11* | 0.527 | 1.243 | 1.024 | 1.254 | 0.547 | −0.065 | −0.127 | −0.147 | −0.124 | −0.012 |
| *S12* | 1.243 | 3.247 | 2.254 | 3.147 | 2.047 | 0.241 | 0.250 | −0.021 | 0.451 | 0.114 |
| *S13* | 1.024 | 2.254 | 2.414 | 2.995 | 1.324 | −0.132 | −0.214 | −0.357 | −0.121 | −0.221 |
| *S14* | 1.254 | 3.147 | 2.995 | 2.265 | 3.248 | −0.214 | 0.144 | 0.140 | −0.124 | −0.148 |
| *S15* | 0.547 | 2.047 | 1.324 | 3.248 | 1.554 | 0.012 | −0.124 | −0.147 | 0.521 | −0.612 |
| *d1* | −0.065 | 0.241 | −0.132 | −0.214 | 0.012 | 3.624 | 1.547 | 0.784 | 0.665 | 0.687 |
| *d2* | −0.127 | 0.250 | −0.214 | 0.144 | −0.124 | 1.547 | 0.625 | 1.241 | 1.741 | 1.514 |
| *d3* | −0.147 | −0.021 | −0.357 | 0.140 | −0.147 | 0.784 | 1.241 | 1.847 | 2.054 | 2.154 |
| *d4* | −0.124 | 0.451 | −0.121 | −0.124 | 0.521 | 0.665 | 1.741 | 2.054 | 2.147 | 1.884 |
| *d5* | −0.012 | 0.114 | −0.221 | −0.148 | −0.612 | 0.687 | 1.514 | 2.154 | 1.884 | 2.149 |

The obtained data files are transformed to specified tagged form for preprocessing and imply the data cleaning method to remove the data containing noise and filling up the missing data.

Authors assume that each article has keywords after abstract which will be extracted as features along with features from abstract. Title of article is preprocessed to extract keywords from it, if list of keywords are not given after abstract in article. Number of terms and documents are initialized based on selected dataset to find/extract features using proposed method to find article. The system configuration used to perform experiment is I5 machine with 8 GB of RAM.

### 4.2. Evaluation measures

It is a subjective work similar to article [33] of 2019 and difficult to evaluate the quality of integration. The high degree of similarity between the cluster and within the cluster is achieved by low affinity integration objective function in design.

This can be seen as an internal standard for the quality of the integrated cluster. It is observed in the literature [22] also, however, it is not necessary to translate the good effect in the application for a good score on the internal reference.

Two criteria for evaluation of the results like Purity and Normalized Mutual Information (NMI) are used as discussed in Ref. [34].

These methods are used for the cluster that is allocated. The Purpose is to find/evaluate whether the object of integrated result is same as original class information. This is checked for each object to check its matching with original class.

Each data clusters is represented as, C = {C1, …,CJ } for the partition of the dataset constructed through the clustering algorithm, and partitioned, P = {P1, PI } the partition inferred by the unique classification. J and I are correspondingly the numbers of clusters denoted as |C| and the number of cluster classes denoted as |P|. The N denotes the total number of data objects in datasets.

**Purity Metric:** The evaluation of a measure of purity is a simple and transparent. Each individual cluster is allocated to the class determined by dividing the number of including the number of objects in the precision of the specified object and this exact allocation, and most frequently in the cluster data sets to calculate the purity [34]. The number of clusters is easy to achieve a high purity as shown in equation (5). Therefore, trade-off quality cannot be used for the purity of the number of clusters.

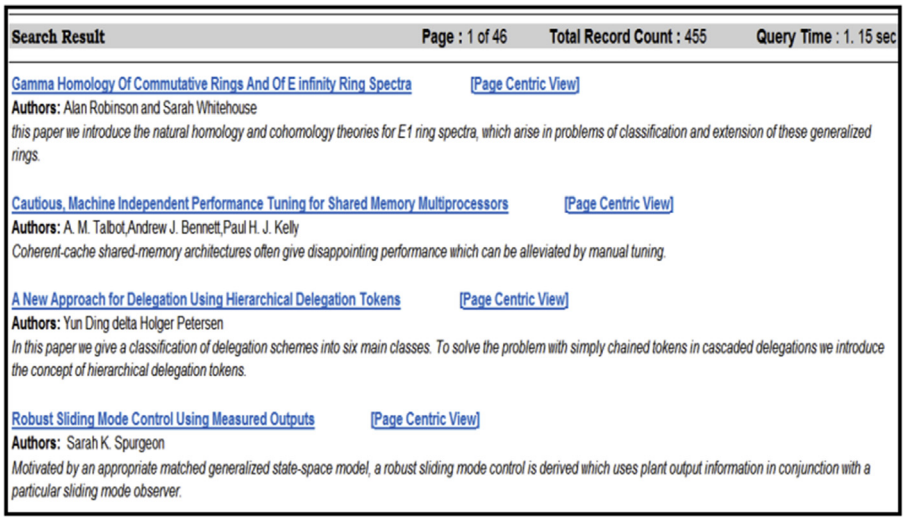


**Fig. 4.** Data access results.

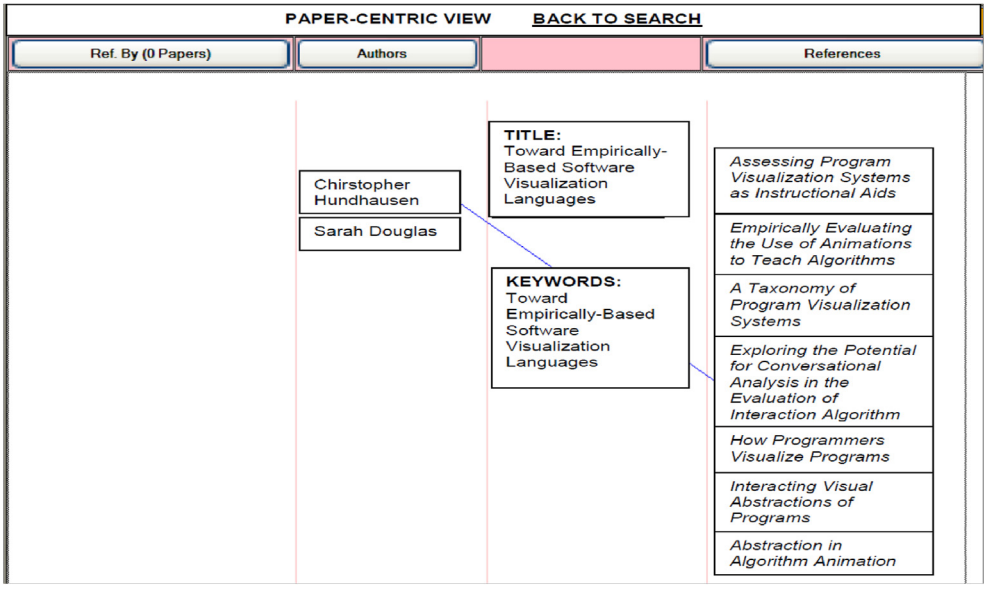


**Fig. 5.** Data access visualization.

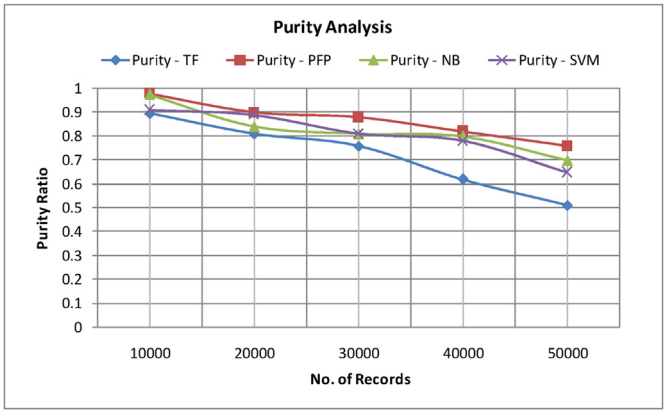


**Fig. 6.** Integrated data purity analysis.

$$Purity(C,P)=\frac{1}{N}\sum_{j}\max\left|C_j\cap P_i\right| \tag{5}$$

**NMI Metric:** The measure of NMI metric offers a number of independent information in the cluster. The measure takes a maximum value [15], when this integration to partition completely consistent with the original partition. The average mutual information of NMI is computed between a pair of clusters and the individual class as shown in equation-6.

$$NMI(C,P)=\frac{\sum_{i=1}^{I}\sum_{j=1}^{J}\left|C_j\cap P_i\right|\log\frac{N\left|C_j\cap P_i\right|}{\left|C_j\cap P_i\right|}}{\sqrt{\sum_{j=1}^{J}\left|C_j\right|\log\frac{\left|P_i\right|}{N}\sum_{i=1}^{i}\left|P_i\right|\log\frac{\left|P_i\right|}{N}}} \tag{6}$$

Many previous studies have only been used to analyze the purity metric and evaluate the clustering algorithm performance. However, when a larger number of clusters are available for the integration, it is easy to achieve purity measure. Particularly if each object has its own data clusters purity measured as 1. In addition, many of the partitions have the same purity as they are associated with each other. For example,

a number of object data in each cluster is different to objects that make up the cluster. So, measure NMI metrics is measured to have the overall effectiveness and how obtained clusters results are equivalent to the original classes.

In order to evaluate the usefulness of the proposed indexing method, the indexed results "precision (IPR), recall (IRR) and accuracy (IAR)" are measured. The result as the based on the standard equations as shown below

$$IP_R = \frac{|No.of\ True\ result\ |}{|Total\ No.of\ Associated\ Result|}$$

$$IR_R = \frac{|No.of\ True\ result\ |}{|Total\ No.of\ Related + Associated\ Result|}$$

$$IA_R = \frac{|Total\ No.of\ Associated\ Result\ |}{|Total\ No.of\ Search\ Result|}$$

The outcome of these measures is discussed in the following section. It is based on the quantitative results to determine the indexed results measurement towards ranking.

### 4.3. Results

In this section, the experiment result analysis of the proposed PFP based integration is discussed. F-LSA based indexing method is analyzed through comparison of semantically associating approaches based on Term Frequency (TF), Naive Bayes (NBP), and Support Vector Machines (SVM) to measure the performance improvisation.

An experiment result analysis is provided for the integration and indexing through comparing the outcomes in their normal and with the proposed method. In the Normal analysis form, data integration is made simply based on it terms related to the class. In case of proposed integration the integrated data clusters is analyzed against the class articles through measuring it purity and NMI.

#### 4.3.1. Integration performance analysis

The analysis of the integration performance in terms of purity measure and NMI measure is presented in Figs. 6 and 7 respectively. The proposed PFP shows an average of 8% higher purity compared to the existing semantic based associating approaches over the records.

In comparison with NB based association, PFP shows a nearby purity but TF and SVM shows linear loss of purity with increasing the number of records. The enhancement in the purity is due to the accuracy in learning of the feature association between the data records to its article class through the probabilistic feature Patterns and semantic similarity.

In case of NMI analysis the mutual information is measured, which is shared between the data record and the learned article class terms in proposed PFP and the existing semantic based associating approaches for the integration. Result shows the decreasing in NMI variation in case of TF, Support Vector Machine and Naïve Bayes due it has high number of independent terms with increase number of data records. But the proposed PFP probability of relevance of terms with article class increases with the number of mutual information and supports in achieving better NMI in comparison. The SDRL approach makes to relate the anonymity records more precise in order to achieve better purity and NMI in comparison.

#### 4.3.2. Indexing performance analysis

The indexing of data helps to retrieve information much faster and accurate. To have an analysis of the indexing approach "Precision", "Recall" and "Accuracy" of the indexed data is measured through an user accessing model. Evaluation of proposed F-LSA based indexing method with TF and LSI based indexing for the integrated data according their article classes is carried out. The outcome of the indexing analysis results are shown in Figs. 8–10 respectively.

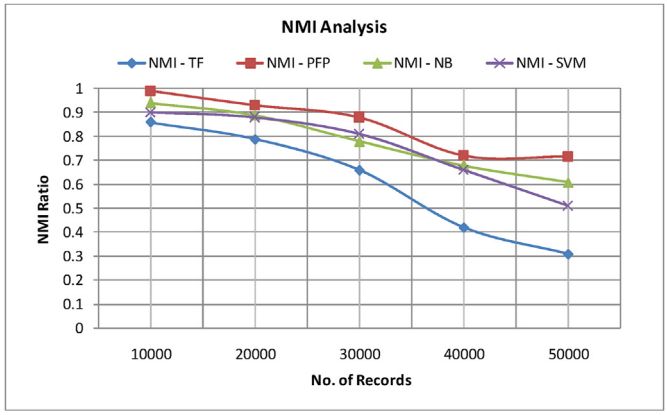


**Fig. 7.** Integrated data NMI analysis.

Figs. 8 and 9 show the performance of indexing accuracy in terms of precision and recall in comparison. The relevance of result retrieved against each query are measured to compute the precision and recall. Figures show that more number of data records in the article classes leads to reduction the precision and increase of recall. The reduction of the precision is due to the increase in the number of irrelevant associated results in the retrieve results. This might be the cause of the purity reduction in the integrated data and lowering in similarity index among the data record which results in poor indexing. But an enhancement of an average of 14% precision with TF based indexing and 3% with LSI is being achieved with utilizing the proposed F-LSA based index approach. Fig. 10 shows the accuracy of the retrieved query results. The accuracy of retrieved results completely depends on the semantic similarity measure between the query and indexed results. The irrelevancy in indexing and lower in semantic similarity index among the records lowers down the accuracy of the retrieved results. The computed result shows a reduction in accuracy with increasing number of data records, but an average of 15% improvisation in accuracy is achieved in compared to TF based indexing and 6% in compare to LSI over the integrated data.

#### 4.3.3. Integration and indexing performance analysis

The analysis efficiency and ability of the proposed PFP based integration through semantically classification over a Naïve Bayes classifier with F-LSA based indexing (PFP with F-LSA) over the heterogeneous sources is carried out. Evaluation is carried out with the three digital bibliography datasets: "CiteSeerX", "BibText" and "Cora" with the Normal form of integration with LSA based indexing (Normal with LSA). Efficiency is tested by submitting few queries to the designed used data access interface such as, "Software engineering", "Artificial Intelligence" and "Data Security" on the three datasets. Over 100 top retrieved results are analyzed for the query to find the number of true result, number of associated results and total number of related results from the number of search results.

Figs. 11 and 12 show the precision and recall performance for the proposed PFP with F-LSA and Normal with LSA. The obtained result of the proposed PFP with F-LSA shows an average of 13.2% of improvisation in the precision and 10.5% of low recall in compare to the Normal with LSA. The improvisation achieved due to the retrieval of more number of related results, which suggest the improvisation of the integration. The total number of true results suggests the accuracy of the indexing. In case of accuracy measure, an average of 11.8% of improvisation is observed due to high number of associated result retrieval as shown in Fig. 13.

The experiment outcome of the precision, recall and accuracy against the various bibliographic datasets shows an enhancement compared to TF with LSA. The improvisation of the result is due to efficient integration and indexing of data records semantically. The methods of F-LSA facilities to retrieve data, to identify the most similarity data records indexed

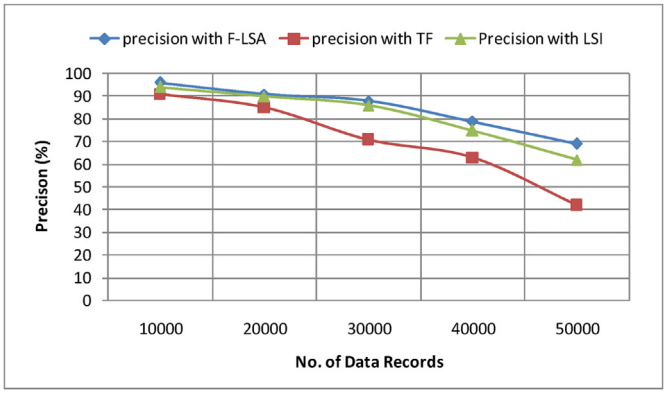


**Fig. 8.** Precision result analysis.

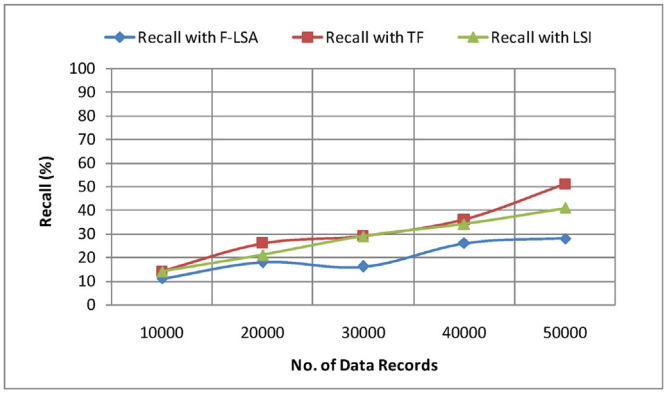


**Fig. 9.** Recall result analysis.

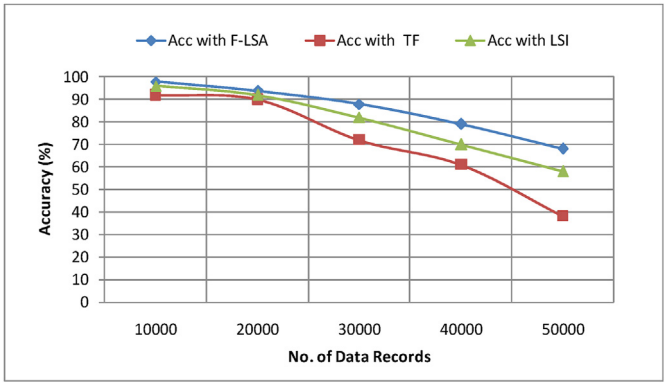


**Fig. 10.** Accuracy result analysis.

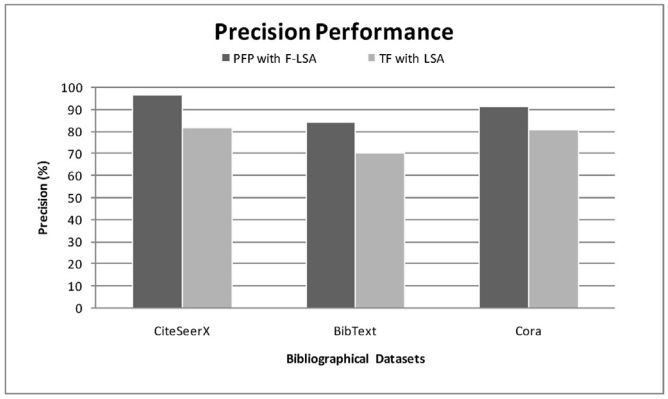


**Fig. 11.** Precision performance analysis.

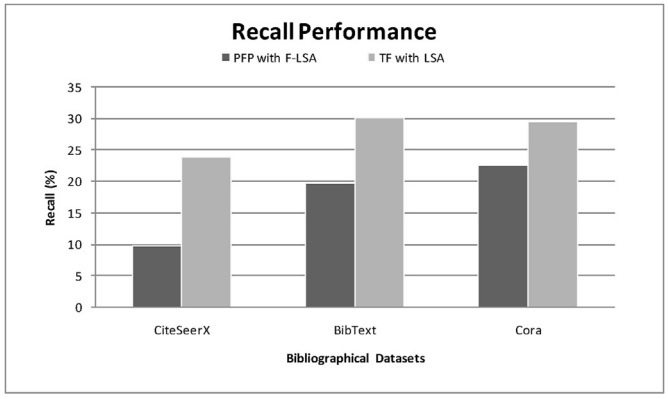


**Fig. 12.** Recall performance analysis.

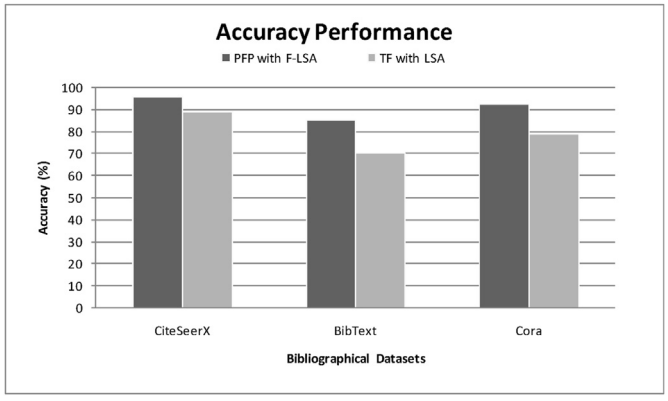


**Fig. 13.** Accuracy performance analysis.

to their relevance group formed through integration.

Purpose of indexing the records is for faster access and quick retrieval. Accuracy depends on indexing as it depends on feature representation. Good/unique features results in better accuracy. Good features reveal data clearly and any method depends how quality features are extracted for indexing.

## 5. Conclusion

Integration and indexing are two principal functions needed to organize and retrieve data faster in today's big data environment. Hence Implementation is in two folds. Initial line of work is a PFP approach for the efficient integration through semantic classification over a Naïve Bayes classifier. Further an indexing by F-LSA method is implemented on the integrated data.

This paper reported a systematic performance evaluation of an efficient integration of Bibtext data sets. The proposed PFP approach proved as right choice compared to SVM and TF especially for following three possibilities such as multidimensionality and multi features, unstructured and complex data, Bibtext and huge volume information.

The selection of features and semantic analyzing is able to enhance the integration in big data. A process of semantic data relation learning (SDRL) method is discussed to learn k-features related to collection of data. These features are analyzed to predict a one-feature class of a data. Further effective grouping is done for Integration. The process of indexing utilizes the latent semantic analysis (LSA) method to understand the degree of similarity between features. The correlation between the terms of data records ranks appropriately for indexing.

The integration method associates the features semantically to classify the data record using learned patterns. Further a feature LSA among the integrated class data is implemented to construct the indexing of the integrated data structure. The user access model is presented to evaluate the effectiveness of the integration and indexing based on bibliographical data which includes various technical article published by well known publishers in different domains. Further Performance analysis of two different combinations i,e "PFP with F-LSA" and "TF with F-LSA" is presented. The effectiveness and efficiency of PFP with F-LSA is proved for different data set like Citeseer and Cora apart from Bibtext. There is improvisation in terms of precision and accuracy during indexed data retrieval. Dynamic construction of index during query processing can be carried out in future to minimize the complexity of index column storage.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## CRediT authorship contribution statement

**Madhu Mahesh Nashipudimath:** Conceptualization, Methodology, Investigation, Writing - original draft, Software. **Subhash K. Shinde:** Data curation, Software, Visualization, Supervision. **Jayshree Jain:** Validation, Writing - review & editing, Supervision.

## Acknowledgment

Madhu Mahesh Nashipudimath would like to thank Principal and Management of Pillai College of Engineering, New Panvel, Navi Mumbai, India for their continue support and cooperation during this research work. Special gratitude to Dr. Satishkumar Varma for his constructive criticism and suggestions.

This research work received no specific grant from any funding agency in the public, commercial, or not-for profit sectors.

## References

[1] Salloum SA, Al-Emran M, Monem AA, Shaalan K. Using text mining techniques for extracting information from research articles. Intell Nat Lang Process Trend Appl. 2018;740:373–97.
[2] Zhanga P, Essaidc A, Merkb CZ, Cavalluccia D. Case-based reasoning for knowledge capitalization in inventive design using latent semantic analysis. In: Elsevier international conf. on know. based and intelligent information and eng. systems; 2017. p. 6–8.
[3] Jain Divya, Singh Vijendra. Feature selection and classification systems for chronic disease prediction: a review. Egypt Inf J 2018;19(3):179–89.
[4] Nashipudimath MM, Shinde SK. Indexing in big data. In: Computing, communication and signal processing. Springer; 2019. p. 133–42.
[5] Gani A, Siddiqa A, Shamshirband1 S, Hanum F. A survey on indexing techniques for big data: taxonomy and performance evaluation. Springer-Verlag Knowledge Info. System; 2015.
[6] Lei Cong, Zhu Xiaofeng. Unsupervised feature selection via local structure learning and sparse learning. Multimed Tool Appl 2018;77(22):29605–22.
[7] Lewis Suzanne, Damarell Raechel A, Tieman Jennifer J, Camilla Trenerry. Finding the integrated care evidence base in PubMed and beyond: a bibliometric study of the challenges. Int J Int Care 2018;18(3).
[8] Roh Y, Heo G, Whang SE. A survey on data collection for machine learning: a big data integration perspective. IEEE Trans Knowl Data Eng 2019.
[9] Zhou Peng, Chen Jiangyong, Fan Mingyu, Du Liang, Shen Yi-Dong, Li Xuejun. Unsupervised feature selection for balanced clustering. Knowl Based Syst 2019.
[10] Luan, Cuiju, and Guozhu Dong. "Experimental identification of hard data sets for classification and feature selection methods with insights on method selection." Data Knowl Eng 118 : 41-512018.
[11] Pratiwi Asriyanti Indah. On the feature selection and classification based on information gain for document sentiment analysis. Appl Comput Intell Soft Comput 2018.
[12] Hancer Emrah, Xue Bing, Zhang Mengjie. Differential evolution for filter feature selection based on information theory and feature ranking. Knowl Base Syst 2018; 140:103–11.
[13] Venkatesh B, Anuradha J. A hybrid feature selection approach for handling a high-dimensional data. In: Innovations in computer science and engineering. Singapore: Springer; 2019. p. 365–73.
[14] Fan W, Bouguila N, Ziou D. Unsupervised hybrid feature extraction selection for high-dimensional non-Gaussian data clustering with variation inference. IEEE Trans Knowl Data Eng Jul. 2013;25(No. 7):1670–85.
[15] Thi Nguyen T-H, Huynh V-N. A k-means-like algorithm for clustering categorical data using an information theoretic-based dissimilarity measure. Springer International Publishing Switzerland; 2016. p. 115–30. https://doi.org/10.1007/978-3-319-30024-5.
[16] Adnan K, Akbar R. An analytical study of information extraction from unstructured and multidimensional big data. J Big Data 2019;6(1):91.
[17] Yang J, Li X. MapReduce based method for big data semantic clustering. In: IEEE international conference in systems, man, and cybernetics (SMC); 2013. p. 2814–9.
[18] Celeux Gilles, Maugis-Rabusseau Cathy, Sedki Mohammed. Variable selection in model-based clustering and discriminant analysis with a regularization approach. Adv Data Anal Classif 2019;13(1):259–78.
[19] Liu, Zio E. Integration of feature vector selection and support vector machine for classification of imbalanced data. Appl Soft Comput J 2018.
[20] Yousefpour Alireza, Ibrahim Roliana, Hamed Haza Nuzly Abdel. Ordinal-based and frequency-based integration of feature selection methods for sentiment analysis. Expert Syst Appl. 2017;75:80–93.
[21] Selvakumar S, Senthamarai Kannan K, GothaiNachiyar S. Prediction of diabetes diagnosis using classification based data mining techniques. Int J Stat Syst 2017; 12(2).
[22] Guan Y, Jordan MI, Dy JG. A unified probabilistic model for global and local unsupervised feature selection. In: Proc. 28th Int. Conf. Mach. Learn.; 2011. p. 1073–80.
[23] Kumar Reddy LK, Phani Kumar S. A OT-k label learning classification based on association rules for multi-label datasets. J Theor Appl Inf Technol 2017;95(19).
[24] Zhang Qingchen, Yang Laurence T, Castiglione Arcangelo, Chen Zhikui, Peng Li. Secure weighted possibilistic c-means algorithm on cloud for clustering big data. Inf Sci 2019;479:515–25.
[25] Mallik Ritik, Abhaya Kumar Sahoo. A novel approach to spam filtering using semantic based naive bayesian classifier in text analytics. In: Emerging technologies in data mining and information security. Singapore: Springer; 2019. p. 301–9.
[26] Adamu1 FB, Habbal1 A, Hassan1 S, Cottrell RL, White B, Abdullahi1 I. A survey on indexing techniques for big data: taxonomy and performance evaluation. In: 4th international conference on internet applications, protocol and services; 2016. https://doi.org/10.13140/RG.2.1.1844.0721.
[27] Liu Xiaojun, Wang Mengmeng, Fu Hanliang. Visualized analysis of knowledge development in green building based on bibliographic data mining. J Supercomput 2018:1–17.
[28] Anandarajan Murugan, Hill Chelsey, Nolan Thomas. Semantic space representation and latent semantic analysis. In: Practical text analytics. Cham: Springer; 2019. p. 77–91.
[29] Merchant Kaiz, Pande Yash. Nlp based latent semantic analysis for legal text summarization. In: 2018 international conference on advances in computing, communications and informatics (ICACCI). IEEE; 2018. p. 1803–7.
[30] Li Zhen, Li Weiguang, Zhao Xuezhi. Feature frequency extraction based on singular value decomposition and its application on rotor faults diagnosis. J Vib Contr 2019; 25(6):1246–62.
[31] Song W, Park SC. Genetic algorithm for text clustering based on latent semantic indexing. Elsevier Comput Math Appl 2009;57:1901–7.
[32] Bibtext data is available on https://sites.google.com/a/iesl.cs.umass.edu/home/data/bibtex.
[33] Lakshmanaprabu SK, Shankar K, Ilayaraja M, Nasir Abdul Wahid, Vijayakumar V, Chilamkurti Naveen. Random forest for big data classification in the internet of things using optimal features. International journal of machine learning and cybernetics 2019;10(10):2609–18.
[34] Lakshmanaprabu SK, Shankar K, Ilayaraja M, Nasir Abdul Wahid, Vijayakumar V, Chilamkurti Naveen. Random forest for big data classification in the internet of things using optimal features. International journal of machine learning and cybernetics 2019;10(10):2609–18.
[35] Bettoumi Safa, Jlassi Chiraz, Arous Najet. Collaborative multi-view K-means clustering. Soft Computing 2019;23(3):937–45.