International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Inductive Learning Approach in Job Recommendation

Ravita Mishra[1*], Sheetal Rathi[2]

Submitted: 10/02/2022 Accepted : 06/05/2022

Abstract: A recommender system is an information filtering system found in various applications, including social networking, e-commerce, business, academics, and research. It assists users with locating the most likely and entertaining facts from a collection of data. The job recommender system aids in the recruitment process by advising candidates and recruiters on suitable jobs and abilities. The current job recommender system provides job recommendations and the necessary abilities to assist in the search for a future profession. The machine learning algorithm is critical in guidance; however, it suffers from cold start and sparsity problems. Many researchers are unconcerned about system and data scalability. As a result, inductive learning can help overcome this problem by providing faster skill suggestions and recommendations. When dealing with a significant amount of data, or big data, the missing value ratio is frequently too high, affecting the learning model. Either discard records with missing values or replace a proper value and solve the problem to enhance model efficiency. The first strategy was ineffective since missing data can be significant in the induction process. We solve the feature selection strategy in this work, which selects relevant features and fills in lacking values. The novel feature selection and missing value handling technique are compared to the baseline algorithm on CareerBuilder's benchmark dataset. In comparison to baseline approaches, the proposed algorithm produces better outcomes. Intrusion detection, text-to-speech conversion, and job recommendation are some of the most common uses of ILA.

Keywords: *FastILA, ILA (Inductive Learning Algorithm), ILA-1, ILA-2, ILA-3, ILA-4, MVI (Missing value imputation), MCV (Most common value), PF (Penalty factor, SGBA (Scalable Graph-based Approach)*

1. Introduction

Job Suggestion: Job recommendation is a relatively young research subject in academic and corporate research applications. Various methodologies are applied to increase recommendation performance and provide better ideas for future skills. Inductive learning is critical for improving the learning performance of a model-based system [2,5]. There are numerous restrictions and performance difficulties with the current system. Only a few recommendation issues have been studied in this direction; others remain unsolved. The recruitment domain requires the development of a rapid and scalable system. To some extent, inductive learning combined with deep learning will solve this problem.

The basic goal of inductive learning algorithms is to increase their classification strength on test samples that have never been seen before. We use the classification algorithms ID3 and AQ to compare ILA performance. ID3 is a decision tree-based overturning approach that performs stepwise splitting and occasionally overfits caused by unnecessary and irrelevant requirements. It can sometimes have an impact on the classification of an unknown sample. Tree pruning is a common solution, but it does not work with probabilistic data. Uthaursamy (1991) demonstrated a method for dealing with inconclusive datasets. ID3 is similarly ineffective on a larger number of samples. The windowing solution deals with sample difficulties, but the decision tree cannot accurately classify all cases [1,9].

The AQ algorithm is a symbolic decision rule-based machine learning algorithm. It creates a combination of qualities with a value condition, and it considers specialization to exclude negative cases. The space of complexes in AQ searches corresponds to the actual data, and rules do not operate flawlessly on the training data. Similarly, CN2 uses the same heuristic technique as AQ but without relying on a single case. AQ and CN2 are rule-based induction methods that do not need flow diagrams. Salzberg 1994 has demonstrated a numeric value attribute induction system using a decision tree induction mechanism. Another RULES rule induction algorithm that can identify concealed cases has been shown by Aksoy, 1995. The RULES algorithm generates an increasing number of rules, making it difficult to handle large amounts of data.

1.1. Inductive learning

One of the most important processes in data mining is inductive learning, which focuses on identifying broad descriptions. Inductive learning is given with a set of training instances in supervised machine learning, where each sample is described by a vector of attribute values and a class label. It elicits a set of rules that applies to all scenarios. The term features in inductive learning might include the job id, firm Name, education, job title, technical skill, and class label, indicating a job recommendation or skill suggestion. In terms of application, ILA's rule narration is suitable

[1] *Thakur College of Engg. and Technology, Sant Gadge Baba Amravati University, Kandivali (E), 400101, Mumbai, India*
ORCID ID:0000-0002-4577-5411
[2] *Thakur College of Engg. and Technology, Sant Gadge Baba Amravati University, Kandivali (E), 400101, Mumbai, India.*
ORCID ID:0000-0002-1579-788X
** Corresponding Author Email: m.ravita@gmail.com*

for data exploration since it concentrates on a single rule and generates a rule with the ancestor that contains its description section. The ILA algorithm divides the world into three categories. ILA-1, ILA-2, and ILA-3 are the three algorithms, with ILA-4 being the final one. ILA develops a set of organizing rules for a set of training samples. It works iteratively, with each iteration looking for a rule covering a particular class with many training samples. Fig 1 illustrates the ILA family's various inventions from 1998 to 2021. It includes the four generations of Inductive learning algorithms and the proposed approach for the job domain. discussions of approaches are discussed below. ILA methods project in a rules-per-class way. Each category separates samples in that category from the spare classes.

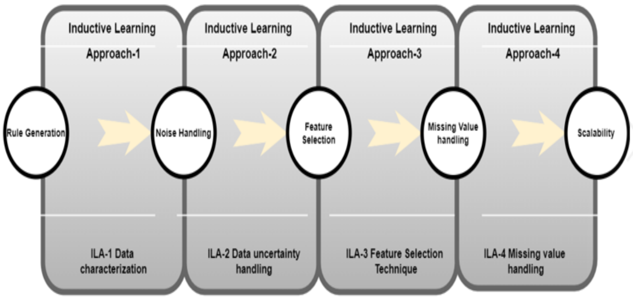


Fig. 1. ILA family (Inductive Learning Algorithm -1 for data characterization and ILA-2 for data missing value handling and ILA-3 for feature selection, and ILA-4 for handling missing value handling)

1.2. Inductive Learning Approach-1

ILA-1 is a supervised, robust algorithm that works with unique and symbolic values while categorizing representative data. Iteratively, each recurrence identifies a rule that covers a large number of training samples and generates IF-THEN rules in standard form from a collection of specimens. It evaluates all examples for a specific class, chooses the general traits, and excludes excessive or unrelated conditions. The results are simpler and more common than those obtained using the ID3 and AQ algorithms. ILA-1 uses fewer rules than ID3 and AQ and operates on a class-by-class basis. Instead of using decision tree methods, it generates a sequential list of claims, and the description feature contains the number of positive and negative samples floating around in the class. It only works with a single table or database and does not perform data analysis [2].
Rule evaluation in ILA-1: The rules are evaluated by ILA-1 using two parameters. First, there's the total number of rules and the average number of conditions. Its goal is to generate as few rules as possible to classify the instances in the training set. Second, it seeks to show whether the system can classify more samples that haven't been seen yet. ILA is a non-contradictory inductive procedure that works with non-contradictory example sets. Limitations of ILA-1: There are two major issues with ILA-1: over-fitting and a long learning time. The bias causes an overfitting problem by generating a consistent classifier on the trained set. The following points are aimed at making ILA more efficient. 1. Removal of Unnecessary Criteria: ILA removes such extra requirements from the directive, resulting in a significant reduction in the average number of conditions. 2. Unseen Examples Classification: The small junction problem, to varying degrees, will have the least impact on ILA's performance. ILA accuracy outperforms decision tree rules derived from unobserved data.
Findings: Adapt a discretization technique D-2 for noisy and incomplete examples, saving computation time and allowing ILA to deal with continuous feature values. In all of the tests, the extracted rules' generalities are achieved. ILA removes a pointless, irrelevant rule and increases accuracy from a previously unseen training set.

1.3. Inductive Learning Approach-2

There were loud, inadequate, active, undesirable, continuous, and missing values in real-world data. To get from data to intelligence, all steps are crucial. If a dataset's description is less than perfect, it's not enough to induce rules. The basic goal of inductive learning algorithms is to find general definitions of a topic from a group of training examples. ILA-2 is an improved version of the inductive learning algorithm-1, with two new assessment metrics for dealing with data uncertainty and bias. ILA-2 is more accurate than ILA-1, and it quickly detects unseen instances. Its learning rate and classifier size have increased. ILA is an iterative algorithm that works on a rules-per-class basis. Rather than a decision tree, it generates a list of rules in sequential order. The extracted criteria limit false-positive illusion before selecting a default class. The rule set goes through a post-pruning procedure, with the first rule being deleted and their location being double-checked. The penalty factor controls the algorithm's performance.
In ILA-2, the class dimension and precision are the most important rule evaluation parameters. The sum of the conditions of the rules in the group is the class dimension. Precision is defined as the assessed certainty on test data, and it is used to assess the certainty of a prediction on unknown data. The Penalty Factor is comparable to sensitivity in that it estimates the negative impact of false-negative instances on description form. Considering the two closeouts and the centre of the PF series, the parameter Penalty factor (PF) is set to 1 and 5. The algorithm works similarly to the basic ILA-1 in the upscale mode. For penalty factor 1, the algorithm creates a compact category for the lower back analysis sets.
FastILA speeds up ILA-2 by using only one description to generate a new rule, cutting the processing time in half from 17 seconds to 10 seconds. It has two loops: an outer loop that takes time O (c) and works for each class feature value and an interior loop that takes time O (c) and works for each class feature value. The inner circle continues to work until it has covered all of the samples in the current group [6].
Findings: Feature subset selection (FSS) methods are incorporated into the pre-processing processes and help to increase performance. It removes the redundant attribute combinations and decreases the system's processing time.

1.4. Inductive Learning Approach-3

For big and noisy datasets, ILA-1 is inefficient, and three factors usually characterize it: 1) many rules generated, 2) rules are simple, 3) generated rules induction power, i.e. the accuracy of properly recognizing the new sample, and finally, 4) generated rules speed (efficiency of the algorithm). ILA-3 was created and fitted with a new feature selection algorithm to deal with such a situation. It splits the dataset into secondary tables, with one sub-table for each class value. ILA-3 then continues its usual processes of generating rules for all combinations except those that exist in ExcludedCombinationsList, i.e. except the irrelevant combinations, by calling a new algorithm called CombExclude that includes several combinations and excludes irrelevant features and stores them in ExcludedCombinationsList. It performs well on huge datasets with a large number of possible combinations. Because it manipulates the full dataset without eliminating any combinations, it is substantially more efficient than the original

ILA [10].

Feature subset selection: It detects and removes as much unrelated and unnecessary data as possible, making the learning process run faster and more efficiently and reducing the data's dimensionality. The feature subset selection strategy is more compact to reflect the target concept. It enhances future classification accuracy and lowers irrelevant features, removes irrelevant, redundant, or noisy data, and has immediate impacts for applications such as intensifying a knowledge base algorithm and increasing extract performance to predict accuracy and consistency.

Model for feature selection: There are three ways to do it: The filter model first chooses features in the pre-processing step. This strategy has a major flaw in that it ignores the impact of the selected pieces on the algorithm's performance. On the other hand, the wrapper model employs a forward search [best-first or Hill-climbing] algorithm, while the hybrid model employs both features. Feature selection techniques are usually either stand-alone or tuned with an inductive algorithm. As a subprogram, stand-alone algorithms are referred to as induction algorithms. They can be used as a pre-processing step in any inductive learning algorithm before the learning algorithm is executed. They can be used as a pre-processing step in any inductive learning algorithm before the learning algorithm is executed. It greatly decreases the dataset, resulting in a major improvement in the method's competency. The fundamental disadvantage of stand-alone is that it ignores the underlying properties of the learning algorithm. The customized feature algorithms are typically embedded into an inductive learning algorithm and take advantage of the learning algorithm's functionality. It looks for better features compatible with the learning algorithm to increase learning performance.

Taxonomy for feature selection: Dd is indicated by a duplicate example; all duplicated samples should be removed, leaving only one. Both instances of the conflicting samples indicated by Cc should be eliminated from the dataset in this situation. Mc stands for missing classes; in this situation, when an attribute is removed from a dataset, it also removes some class values from the dataset. The combination that removes from a dataset is very irrelevant. Performance, the sum of resulting rules, and the mean of conditions in the generated rules have all improved; nonetheless, the induction power of the resulting rules on the original dataset will remain at 100%. As a result, a combination is irrelevant if the ratio Cc = 0 is removed. The weakly unrelated combination is removed from a dataset, and performance, the sum of resulted rules, and the mean of conditions in the resulted rules are all improved; as a result, the induction power of the resulted rules on the original dataset is less than 100%. If the ratio Cc / Cd > 0 after it is deleted, we award a weakly irrelevant in ILA-3. CcCdRatio will denote this ratio. It is relevant if the appropriate combination is not highly or weakly irrelevant and is not eliminated. If Cd = 0 or causes, a variety is suitable; otherwise, a missed class value case is appropriate (MissDecisionClassValue). CombExclude eliminates all unnecessary combinations of characteristics from the dataset and solves the issues that other inductive learning algorithms, such as ID3 and AQ, have with multiple rules generated, rule simplicity, rule correctness, and rule generation time.

Table 1. Comparison of ILA Algorithm

Training set	**ILA-1**	**ILA-2**	**ILA-3**	**ILA-4**
Objective	It generates canonical form rules and eliminates all unused and irrelevant conditions.	Have two loops: the outer loop for class attribute value and the inner loop is repeated until the generated rules cover all examples of the current class.	It was developed and tailored with a new feature selection algorithm. ILA-3 works and excludes the most irrelevant features from the dataset.	Handle dataset with missing values. Applied dataset with varying completeness
Dataset	Weather, shape dataset	Weather, shape dataset	Marketing and weather dataset	Weather data with missing values, Breast cancer and Marketing
Algorithm comp	ID3, AQ	ID3, Similarity model, ILA-1	ID3, AQ, ILA-1, ILA-2	LR, NB and RF
Rules	1. Removal of irrelevant Conditions. 2. Categorisation of hidden samples.	1. In every iteration, it generates more than one rule.	Exclude all irrelevant combinations of attributes from the dataset.	New approaches like Most Common Value (MCV) and the Most Common Value Restricted to a Concept (MCVRC) were used.
Advantage	1. It is mainly helpful in data exploration. 2. It removes all unwanted and irrelevant conditions from the data.	1. Rules suitable for data exploration; 2. It targets a single rule at a time. 3. Various positive and negative samples are found in each description.	1.ILA-3 overcomes ILA and other algorithms for all factors. 2.ILA-3 enhances the efficiency of ILA significantly. 3. CombExclude had been discussed and tailored with ILA to produce the new inductive algorithm.	1. Effectiveness is good 2. Accuracy moderate.
Drawback	1. Overfitting 2. Long learning time 3. inefficient for large datasets	1. inefficient for large datasets.		Time-consuming and less efficient
Future Work	1. Allow to handle noisy and incomplete samples. 2. Adapt a discretisation algorithm D-2 by offering savings in CPU time.	1. Adapting different feature subset selection (FSS) approaches.	Work on distributed environment.	The distributed environment has addressed the efficiency issue.
Accuracy	Better than ID3 and AQ	Better than ILA-1	Improve to another version	Better than the other three algorithms (Moderate)

Findings: For all criteria, ILA-3 outperforms ILA and other methods. It improves the effect is obvious from the data that ILA-3 considerably improves the efficiency of ILA.

1.5. Inductive Learning Approach-4

This algorithm's main goal is to provide a set of classification criteria. It examines specified training data without discarding any values. The model is iterative, with each repeat exploring new rules to maximize the number of training examples. Training data samples are labelled, and generated rules are acknowledged. Such rules are rejected in the following cycle. ILA specifically specifies a rule-per-class approach, in which rule induction is used to infer unrelated samples in the present category from specimens in other classes. The missing value in the dataset causes numerous issues; it reduces the system's efficiency—data preparation and analysis challenges, resulting in bias inquiry, including over-fitting and under-fitting—and leads to a bias investigation. The Delete Strategy is a method for dealing with missing values that uses simple approaches. It asks for data samples that are missing certain features. The second strategy, unvarying treatment, asserts that when the precise answer for all frameworks and the last one case-by-case procedures are used, framework-specific fatalistic value, forecast value, and disseminate value strategies are applied.

ILA-4, designed to cope with missing values, produces the best results since it avoids most of the difficulties other approaches encounter. ILA-4's main goal is to determine how missing information will be handled and eliminated from the database. It considers the significant existing class without jeopardizing the ILA's inner workings, and it maximizes the MCV approach's potential for missing value replacement. The three strategies outlined above work well; the ILA model receives a pre-processed dataset with all possible missing value combinations that have already been substituted. The missing value will be addressed during ILA-4's induction phase [2]. ILA-4 adds new characteristics, such as storing more datasets with missing samples for usage with ILA; these features are not present in a normal approach. The time it takes to assemble ILA-4 is a minor expense that is critical in the design of data pre-processing.

2. Literature Survey

Wohlrab and Fürnkranz (2009) compiled a list of methods for dealing with hidden values in datasets. In the traditional separate-and-conquer rule learning algorithm, this model is used. In addition, the author explores and differentiates general methods using eight strategies: a) Ignored Value, Any Value, Special Value, and Common Value Delete strategy: work on hidden values example ignores, b) Ignored Value, Any Value, Special Value, and Common Value Ignored Value, Any Value, Special Value, and Common Value Ignored Value, Any Value, Special Value, It treats hidden values uniformly across all samples, and c) It treats hidden values differently depending on the sample, such as despairing value, predicted value, and distributed value approach.

Raja and Thangavel (2020) used unsupervised machine learning to impute missing values. Using a combination of soft computing and clustering approaches, a rough K-means centroid-based methodology offers a novel solution for detecting missing value inconsistencies. They compared their methods to those that employed the following models as a foundation: rough K-means, K-means parameter, etc. The author obtains favourable results compared to the UCI benchmark datasets [7].

Rashid and Gupt (2021) demonstrated a model to state machine learning's possible assigned value. It compares MVI approaches to statistics-based methods for their usual time costs. The authors also look at the k-nearest neighbour, support vector machines, naive Bayes, and mode/median as MVI techniques, comparing their efficiency to specified baseline MVI algorithms. They contain the outcome as well as possible extension strategies [8].

MVI approaches that use geographical data to close sensors have been presented by Liu (2021). Because data is transferred in huge numbers, it's necessary to plan for severe data loss from several sensors due to a single occurrence. Imputations of data from nearby or correlated detectors are impossible. The author suggests an MVI approach for univariate time-series data and an iterative framework using multiple segmentation for substantial gaps to address present difficulties. The results of the experiments demonstrate significant gains, particularly in terms of RMSE (root-mean-square-error) metrics.

Do (2018) has established a way for evaluating 30 methods and verifying each approach's ability to meet the following criteria: It does two things: it reconstructs biochemical pathways for association networks, and it increases analytical capacity while keeping control of established metabolic quantitative trait loci. According to the analysis, the k-nearest neighbour method fared well in ineffectiveness and executional costs. This approach is a possible solution when many imputations are required, and larger computing expenses are required.

Saleh (2009) proposed a model without transforming data into a single table; RILA's relational database inductive learning technique can be used for data analysis. In a distributed context, the algorithm fails. The main goal of RILA was to develop rule selection approaches based on the ILA-2 algorithm, which was dubbed early selection. In RILA, rule selection occurs after the hypothesis search procedure is completed. Because RILA has a lot of flaws, DRILA (Distributed Relational Inductive Learning) improves on it by learning from interconnected tables (centralized database). Existing ILA algorithms operate on a single table, which causes issues in a variety of real-world applications. DRILA is a distributed data access algorithm that is an extension of the ILA family algorithm. The distributed relational database identifies distributed rules, includes a collection of websites, and includes several logical, integrated databases disseminated over a network for maintenance purposes—the goal is to analyze items stored in tables and dispersed over many locations. The claims created can anticipate the value of an unknown object property. DRILA leverages data from various locations (sites) and any conceivably related structure at each location where foreign keys join databases. A new hypothesis search algorithm was applied to enhance efficiency instead of recomputing or revising the rule. Because Drill does not reproduce distributed relational rules for processing purposes, it provides scalability and usability [5].

Vachik et al. (2018) developed a homogeneous graph-based strategy for employee recruitment. This method employs Nodes as job posts, and edges are used to compute similarity measures. Weight is defined as the number of co-interactions between the job and the user session in the behavioural data. The content of job posts is learned using a deep learning-based embedding model, and the content of job posts is learned using deep learning embedding. They employ heuristics to aggregate the ratings and then use the PageRank algorithm to make suggestions based on the user's history of job interactions [16,22].

Table 2. Literature Survey Summary (ILA Approach)

	Author Name, Year	Contribution, the dataset used	Advantage	Gaps
1	[14],1998 Susan Dumais, 1998	**ILA**: 1. Decision Trees, Naïve Bayes, Bayes Nets, etc.2. Collate the efficacy of five, unlike algorithms. 3. Most accurate and fastest algorithm-SVM. **Dataset**: Reuters-21578 (12,902 classified into 118 categories)	1. All of the five classifiers are very fast. 2. Less than 2 msec take to determine a new document category.	1. Few algorithms were considered. 2. Extend the text representation models documents, as well as knowledge-based features. 3. More time is spent in the initial text phase than categorisation.
2	[3] Mehmet R. Tolun, 1998	**ILA-1**: 1. It generates IF-THEN rules in canonical form. 2. AQ and ID3. 3. Stepwise forward technique helps to search instance space. 4. ILA is designed for handling discrete and symbolic attribute **Dataset: weather Dataset**	1. More general and robust. 2. Rules are suitable for data exploration. 3. Single rule active at a time 4. overcome the attribute selection problem	1. Noisy and incomplete examples. 2. Not work with continuous attribute values.
3	[6] Oludag M, 1999	**ILA-2**: 1. Novel estimation metric handles variability in the data. 2. ILA-2 handles continuous feature discretization by using the entropy-based algorithm. 3. Deals with uncertain data **Dataset**: object classification	1. Penalty factor control the Performance. 2. Future prediction accuracy estimation on unseen data is handled by the Holdout method. 3. Processing time will reduce by faster pass criteria.	1. Feature subset selection (FSS) approaches give better performance. 2. Search space requirements and processing time of the system will reduce.
4	[10], Saleh M. Abu-Soud,2018	**ILA-3**: 1. New Feature Selection Algorithm. 2. ILA-3. CombExclude excludes all irrelevant combinations of attributes. 3. Not involved in the induction process itself. **Dataset**: letter recognition dataset (20000)	1. Overcomes the problem of ID3 and AQ **2.** the efficiency of ILA had been greatly enhanced by the new version	1. Not handle missing value.
5	[25] Luca G., IEEE,2019	1. Within heterogeneous data, it identifies important features and prediction outcomes. 2. Graph embedding algorithms can obtain excellent results **Dataset**: Human Protein Reference Database.	Method: Graphsage 1. 95 % F1 score achieved in Supervised. 2. Unsupervised methods gave reduced prediction accuracy.	1. Improve the performance by leveraging ensemble techniques. 2. Different characteristics of presentation are used.

3. Proposed methodology

3.1. Schematics of the proposed graph-based recommendation system: The graph-based recommendation system will collect six steps.

The input data sources (data collection) for data collection, pre-processing and feature engineering for cleaning and arranging attributes, algorithm selection (Page rank, Modified Page rank, and Random surfer) for page rank and similarity computation, recommendation parameter and parameter tuning for quantitative evaluation and validation purpose, baseline approach for result validation purpose. Fig 2 depicts the whole process of job recommendation with different graph-based and inductive learning algorithms.

1. Input Data Collection: Data was collected from the recruitment company website CareerBuilder.com (Job listing dataset, Job dataset) and Naulari.com. All three datasets are in different sizes and formats; the CareerBuilder dataset has other files (job, resume, user, history) and has more than 2 GB's instances. Naukari.com dataset contains only one file having 14 attributes approximates 1 GB's samples. Datasets are given to the next step, exploratory data analysis [27,28].

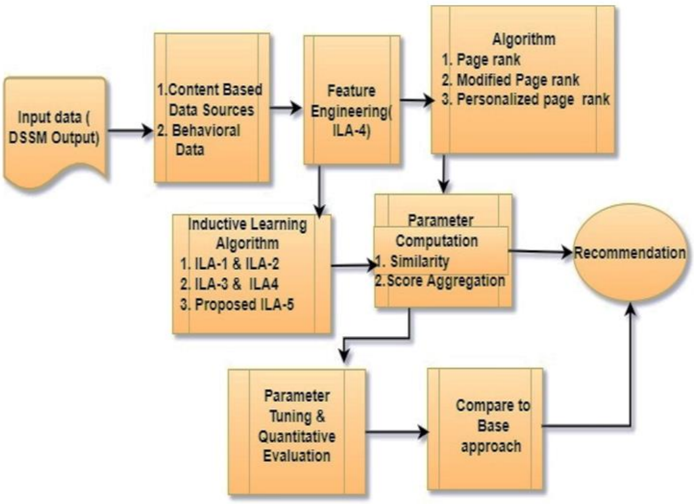


Fig.2. Graph-based Recommendation using Proposed ILA

2. Exploratory Data Analysis (EDA): Both **datasets** require cleaning and removing null values and outliers using statistical techniques. Extracting essential features from the Naukri Dataset using a feature selection algorithm. CareerBuilder dataset contains different files; extract crucial elements from that files and merge both files. The resultant files are further processed by using a new feature selection algorithm. For performance improvement purposes, we have used the new feature selection algorithm termed ILA-3.

3. Feature engineering and attribute selection: : Feature engineering is necessary for analysis purposes. ILA-3 algorithm selects the best features and ignores irrelevant features. One hot encoding applies to the chosen features for further processing. The extracted data are Input to the proposed algorithm for semantic evaluation purposes. One hot encoding applies in the chosen features for three datasets. The extracted one-hot encoded data are selected as Input to the algorithm for further semantic evaluation.

4. Enhanced **Deep Semantic Structure Modelling (E-DSSM):** The machine learning algorithm cannot work with large datasets; a simple machine learning algorithm will not perform ranking and matching accurately because of data bias problems. So, there is a need to design a deep learning algorithm to extract hidden information meaning and handle large datasets. DSSM algorithm generates matching and ranking effectively. 2-arm DSSM also gives the best result in offline data [18].

5. Graph-Based Recommendation **(GBR):** Graph-based recommendation system allows linkage exploration methods from the graph model to handle the limitations of CF-based approaches such as data sparsity problems and aim to improve the quality of the recommendations. In this case, the ranking algorithm (page rank) ranks the pages based on their acceptance and importance of pages. Similarly modified Page Rank Algorithm addresses the Cold start and data sparsity problem [17].

6. Graph-based Recommendation (ILR): In these steps, we apply the inductive learning-4 algorithm to find unseen data in the recommendation pool and handle missing and noisy data [12]. The inductive learning recommendation generates the rule based on the rank list generated by E-DSSM and SGBA model. After generating a specific rule for each feature improves the recommendation speed and system accuracy.

7. Rule Generation: Rule generation is an important task of the inductive learning approach, the basic ILA-1 approach generates the rules, but we are using here to generate rules based on the specific feature we have selected and also generating the rule based on the removal of contradict and duplicate feature and swap the missing value with actual value. The generated rule at the stage of E-DSSM will help give a proper score and matched and unmatched candidate list. The rule generated in the GBA steps will help classify whether the candidate is recommended for the job or not.

8. Result Validation and Hyperparameter Tuning: For result validation purposes, three hyperparameters are selected for tuning purposes. We are validating the result on baseline approach MF (Matrix factorization), GBA (graph-based approach), Skill BERT, and comparing implicit and explicit effects signals of on proposed algorithm. To validate the results of the Proposed ILA-5 approach in jong data, we select various measures like precision, recall, accuracy and PMI score.

9. Metric selection and Performance Evaluation: Different measures like Precision and Recall are basic metrics for performance evaluation and accuracy validation to check the model's efficiency and accuracy. Pointwise Mutual Information (PMI) and Maximum Likelihood Estimation (MLE) evaluate the best score; cosine similarity metrics check for similarity computation.

10. Job Recommendation: The research output is a scalable recommender system algorithm that handles large datasets using GPU or high-performance computing and gives the best recommendation and career suggestions for job recruitment applications.

3.2. Proposed ILA Algorithm

This algorithm selects the appropriate features and discards the irrelevant element based on the criteria mentioned in the E-DSSM algorithm. It also classifies results in two classes: recommended or not recommended. The algorithm discusses in section 3.1 selects the combination of features.

Algorithm 1: Feature selection

Input: Dataset that contains N attributes, one decision class
Output: The set that contains the excluded irrelevant combinations that include j attributes

```
1  Assign ExcludedCombinationsList(j) = Φ, Dtemp= D, I=1
2  #Comb = n! / (j! *(n-j)!)
3  Do While (i<=#Comb) and (CcCdRatio ≥ PredefinedRatio)
      3.1 MaxCcCdRatio = -∞, MissDecisionClassValue = false
       repeat
      3.2 Generate a new combination Nc from Dtemp with j attributes
      3.3 Remove Nc from Dtemp along with its data
      3.4 Cd = number of duplicates in Dtemp
      3.5 If (Cd = 0) or (MissDecisionClassValue = true) then go to step
   5.3.11.
      3.6 Eliminate duplicates from Dtemp
      3.7 Cc = number of contradicts in Dtemp, eliminate contradicts
   from Dtemp
      3.8 If ((MissDecisionClassValue = true) then go to step 5.3.11.
      3.9 CcCdRatio = 100-((Cc / Cd) *100 with upper limit = 100)
      3.10 If (CcCdRatio > MaxCcCdRatio) and (CcCdRatio>=
   PredefinedRatio) then
        (MaxCcCdRatio= 2.10 CcCdRatio) and (MaxComb= Nc)
      3.11 Restore Nc into Dtemp along with its data
      3.12 Increase i by 1
4  Until (i>#Comb)
5    If MaxComb<>Φ then
       5.1. Append combination (MaxComb) into
   ExcludedCombinationsList(j)
       5.2. Remove combination (MaxComb) from Dtemp permanently
   along with its data
       5.3. Eliminate duplicates from Dtemp
       5.4. Eliminate contradicts from Dtemp
       5.5. let MaxComb = Φ
       5.6. n=n-j
       5.7. if n ≤ 0, then go to step 6
       5.8. go to step 2
6    end if
7    end while
8  End
```

This algorithm is a subset of the main feature selection algorithm; it defines the PredefinedRatio and partitions it into two tables.

Algorithm 2: PredefinedRatio Value

Input: PredefinedRatio Value
Output: R

```
1  Input the PredefinedRatio value (≥0 and ≤100)
2  Partition the table containing m examples into n sub-tables—one table
   for each possible value of the class attribute.
3  Initialize feature combination count j as j = 1.
4  if ExcludedCombinationsList(j) ≠ Φ then
   Call CombExclude (out: dataset T, PredefinedRatio, j, in:
   ExcludedCombinationsList(j))
5  For every fusion of j features not in the list
   ExcludedCombinationsList(j), enumerate the occurrences of that
   feature count that shos under the same combination of features in
   unidentified rows of the secondary table under consideration but at the
```

same time that should not occur under the same variety of characteristics of other sub-tables. Repeat the first combination with the utmost number of experiences as max-combination.

6 Take condition max-combination and j by one and go to Step 4.

7 flag all rows of the secondary table for examination, the values of max-combination are considered classified.

8 Append rule to R whose left-hand side is feature names of max-combination with their values, and it is separated by AND operator(s). Its right-hand side contains the decision feature value associated with the secondary table.

9 Take condition all rows are flagged as classified, then swap to process another secondary table and go to Step 3. Otherwise, go to Step 5. If no secondary tables are available, exit with the set of rules obtained so far.

Interpretation: As the dataset contains many missing and irrelevant values, to handle missing value or variation of missing value, we proposed an inductive learning algorithm that handless missing value improves the system's performance. The modified version of the algorithm is discussed below.

Algorithm 3: Modified ILA-4 (CareerBuilder's Sample dataset)

1. Assign MaxCombination
2. Partition the whole dataset (table) into two subtables (n1 and n2). One table for each class (recommended and not recommended).
3. Set the counter for a selected attribute (j=1)
4. We create a unique combination of the distinct attribute (j) for each sub-table.
5. Unique combinations are replaced with an expected value.
6. The same table counts the number of experiences under the same fusion of features in an unmarked row.
7. Check MaxCombination is null or not, and then the counter creates a unique combination of distinct j.
8. Label the sub-table row for consideration and add a rule to the ruleset.
9. Check more rules remain unclassified, then process the sub table.

a. Mark all rows are classified, and unmarked row is not in another sub table.

b. Substitute unmarked row missing value starting from the more specific combination.

c. Extract the associated row and mark it as classified;

d. Ignore it if the same rules are extracted earlier.

e. If it is a new rule, append it into the extracted ruleset.

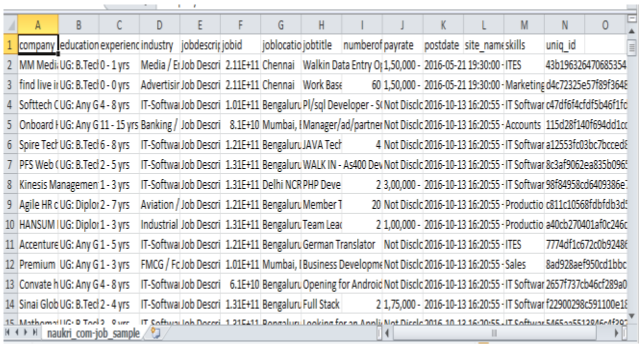
10. If no secondary table remains, exit with the set of rules obtained so far.
11. End

4. Experimental Setup

Inductive learning-3 and Inductive learning-4 have been applied and tested on several datasets from 10 to 16 attribute significance. All experiments were conducted on an IBM compatible machine With 64-bit Windows 10 Professional OS, a 6.2 GHz Intel Core i5-4460 processor, and 16 GB RAM.

4.1. Dataset description

Three datasets we are using are the CareerBuilder job listing dataset having (a size of more than 2 GB, 39123829 instances, 18 attributes) and the CareerBuilder sample dataset having (size 47 MB, 25800 cases, 16 features), and the Naukri sample dataset having (53 MB, 22,300 cases, 14 attributes). We used the Tensorflow for numerical computation and support for Machine Learning and Deep Learning architectures. TensorFlow used. Figure 4.1 represents the sample dataset of CareerBuilder's job listing.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	company	education	experienc	industry	jobdescrip	jobid	joblocatio	jobtitle	numberof	payrate	postdate	site_name	skills	uniq_id	
2	MM Medi	UG: B.Tec	0 - 1 yrs	Media / E	Job Descri	2.11E+11	Chennai	Walkin Data Entry O		1,50,000 -	2016-05-21 19:30:00 -		ITES	43b196326470685354	
3	find live i	UG: B.Tec	0 - 0 yrs	Advertisir	Job Descri	2.11E+11	Chennai	Work Base	60	1,50,000 -	2016-05-21 19:30:00 -		Marketing	d4c72325e57f89f3648	
4	Softtech C	UG: Any G	4 - 8 yrs	IT-Softwa	Job Descri	1.01E+11	Bengaluru	Pl/sql Developer - S		Not Disclo	2016-10-13 16:20:55 -		IT Softwar	c47df6f4cfdf5b46f1fc	
5	Onboard I	UG: Any G	11 - 15 yrs	Banking /	Job Descri	8.1E+10	Mumbai, I	Manager/ad/partner		Not Disclo	2016-10-13 16:20:55 -		Accounts	115d28f140f694dd1cc	
6	Spire Tech	UG: B.Tec	6 - 8 yrs	IT-Softwa	Job Descri	1.21E+11	Bengaluru	JAVA Tech	4	Not Disclo	2016-10-13 16:20:55 -		IT Softwar	a12553fc03bc7bcced8	
7	PFS Web (	UG: B.Tec	2 - 5 yrs	IT-Softwa	Job Descri	1.31E+11	Bengaluru	WALK IN - As400 Dev		Not Disclo	2016-10-13 16:20:55 -		IT Softwar	8c3af9062ea835b0965	
8	Kinesis Managemen		1 - 3 yrs	IT-Softwa	Job Descri	1.31E+11	Delhi NCR	PHP Deve	2	3,00,000 -	2016-10-13 16:20:55 -		IT Softwar	98f84958cd6409386e	
9	Agile HR c	UG: Diplo	2 - 7 yrs	Aviation /	Job Descri	1.21E+11	Bengaluru	Member T	20	Not Disclo	2016-10-13 16:20:55 -		Productio	c811c10568fdbfdb3d5	
10	HANSUM I	UG: Diplo	1 - 3 yrs	Industrial	Job Descri	1.31E+11	Bengaluru	Team Lea	2	1,00,000 -	2016-10-13 16:20:55 -		Productio	a40cb270401af0c246c	
11	Accenture	UG: Any G	1 - 5 yrs	IT-Softwa	Job Descri	1.21E+11	Bengaluru	German Translator		Not Disclo	2016-10-13 16:20:55 -		ITES	7774df1c672c0b9248	
12	Premium	UG: Any G	1 - 3 yrs	FMCG / Fc	Job Descri	1.01E+11	Mumbai, I	Business Developme		Not Disclo	2016-10-13 16:20:55 -		Sales	8ad928aef950cd1bbc	
13	Convate h	UG: Any G	4 - 8 yrs	IT-Softwa	Job Descri	6.1E+10	Bengaluru	Opening for Androic		Not Disclo	2016-10-13 16:20:55 -		IT Softwar	2657f737cb46cf289a0	
14	Sinai Glob	UG: B.Tec	2 - 4 yrs	IT-Softwa	Job Descri	1.31E+11	Bengaluru	Full Stack	2	1,75,000 -	2016-10-13 16:20:55 -		IT Softwar	f22900298c591100e1	

naukri_com-job_sample

Fig. 3 CareerBuilder's Sample dataset

After applying the modified ILA-3 algorithm on my sample dataset, the resultant dataset and its characteristics are discussed below in table 4. Table 4 includes dataset feature descriptions and their encoding.

Table 3. Characteristics of CareerBuilder Job Recommendation Dataset

Sr No.	Characteristics of Data	CareerBuilder Job recommendation dataset
1	No. of attributes	16+1
2	No. of Examples	20000
3	Average value per attribute	15
4	Number of Class Value	26
5	Average Distribution of examples among class values	770

In ILA, rules are generated based on the combination of attributes if three attributes, then their combination is for a single feature (3), and the variety of two elements are (3). The combination of three parts is (1). So, a total of 7 combinations of 3 attributes are generated.

Eq. (1) computes a combination of attributes in the dataset

$$\text{n attributes} = 2^{n}\text{-}1 \quad (1)$$

The number of features gets high, and their combinations increase dramatically. For each extra feature number of combinations doubled. If we work on a large dataset with many attributes, ILA generates many varieties and increases the periods.

Table 4. Encoded Dataset (Two datasets careerbuilder1 and CareerBuilder Sample encoded and their results used)

CareerBuilder 1		CareerBuilder Sample	
Content Feature	Encoding	Content Feature	Encoding
Job Title	20	Job Title	18
Job Description	20	Job description	20
State	55	Education	10
	20	State	12
Requirement Skill	55	City	10
		Country	06
		Skill	55

4.2. Experimental Results

Here, we describe the execution of all the approaches and then analyze the outcome of disparate elements of the models.

Performance of all methods: Hypothesize all models mentioned above into a single category termed GBA with ILA. A standard recommendation model ignores the word order speed and has many problems with semantic understanding of words and ranking of nodes. Proposed graph models use a simple approach to implement and are easy to use at a large-scale data and parallelize the system easily. Table 4.3 depicts the performance of the model.

Table 5. Performance of models (Measuring the performance of proposed model along with existing model)

Model	Precision	Recall	Runtime	Accuracy
doc2vec	91.8	85.3	89	89.2
MF Graph based approach	80.0	78.4	680	72.8
Graph based approach (Input)	92.6	73.2	72	89.0
Graph based approach _DSSM	93.1	91.8	100	90.0
Graph based approach _C_DSSM	92.1	91.8	98	91.5
Graph based approach_2_arm_ DSSM	93.76	91.6	97	92
SJBA_ILR	94.6	92.1	99	94

The base model Matrix factorization (MF) gives high relevance but an inferior recommendation. It produces a low MSE score but has a specific limitation that affects the recommendation conduct. The system leads to a cold start and scalability that affect the model's accuracy. A new system is called the SGBA technique (the latest job does not wait for representation, Matcher learns vector embedding). SGBA system also resolves another problem in matrix factorization termed distribution skew; it tests the system against recommendations (poorly represented are fewer).
Another base approach, Graph-based recommendation (GBR), collates with our proposed system. The GBA model will handle the cold start problem but fails to address the scalability embedding problem due to manual parameter selection. Our approach addresses this problem using a Modified rank algorithm with inductive learning concepts. The execution of the graph-based model with the proposed inductive learning approach shows satisfactory results.

5. Results and Discussion

Results of ILA-3 Feature selection on datasets 1 and 2 (careerbuilder1, CareerBuilder Sample): After applying the feature selection algorithm on dataset1 and dataset2, only relevant features were selected for further processing, and irrelevant elements were removed from the dataset. Table 5 includes a new set of features.

Table 6. Experimental datasets performance measurement with varying missing value rates

	CareerBuilder Dataset				CareerBuilder Job listing Dataset			
Missing values	0%	10 %	30%	50%	0%	10%	30%	50%
Accuracy	100	98	93	89	99	98	94	86

After applying ILA-4 new results are: After using ILA-4 in the CareerBuilder dataset (considering missing values are 0,10,30,50%). As we observe, if the missing value percentage increases, the number of rules also increases, and execution time increases. To improve the performance of ILA-4 and reduce the execution time proposed ILA algorithm gives better results. The variation in results in the different datasets was observed; in dataset2, few rules are generated compared to datsset1.

Table 7. Modified ILA-4 precision against primary experimental datasets with diverse hidden value rates

	CareerBuilder dataset				CareerBuilder Job listing dataset			
% of missing values	0%	10 %	30%	50%	0%	10%	30%	50%
No. of rules	30	30	36	42	28	28	34	41
Average no. of Condition	3,283.14		3.25	3.45	3.20	3.11	3.20	3.46
Execution time(s)	1.721.8		2.2	2.8	1.60	1.7	2.3	2.8

The precision of the ILA model decreases as the ratio of missing values grows. There is a slight variation in both dataset accuracy, and it is resolved by using the distributed feature.
Different techniques handle the percentage of missing values; the three delete strategies (MCV, most common value strategy, delete approach) are used to remove the missing value and compare results with ILA-4.

Evaluation Metrics: For execution estimation, randomly pick 10-20% of job data and estimate the efficacy of top-k recommendation and their similarity ranking; we adopt two estimation measures: precisionj@k and $recall@k$. Consider default value of k is (10-30), and it reports intermediate results for all users in the experimental set.

Table 8. The ILA4 vs. basic algorithm Accuracy (LR-linear regression, NB-Naïve bays, RF-Random Forest)

Algorithm	**ILA-4**	**LR**	**NB**	**RF**
Accuracy (%)	73.1	65.4	71.7	68.5%
Execution Time (s)	0.03	0.01	0.04	0.01
TP rate	0.73	0.65	0.71	0.68
FP rate	92.6	73.2	0.45	0.52
Precision	93.1	91.8	0.70	0.65
Recall	92.1	91.8	0.71	0.68
F-measure	0.729	0.63	0.70	0.66

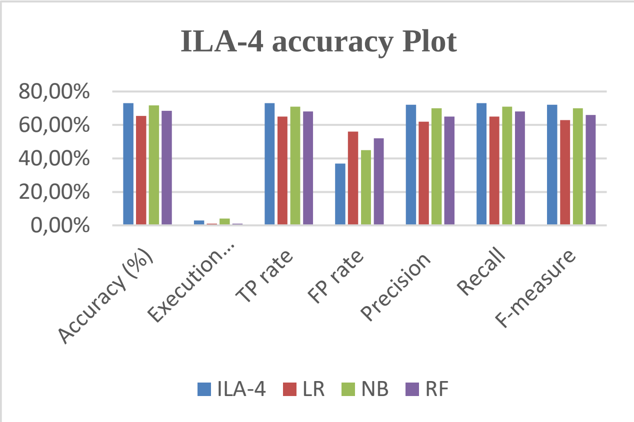


Fig 5. ILA4 vs basic algorithm Accuracy

Here we computed the accuracy of the ILA-4 algorithm and compared it to baseline algorithm LR, NB and RF. ILA-4 gives improved accuracy, reduces execution time, and improves precision and recall values. It is observed that the efficacy of correctly categorized instances of modified ILA-4 is on par with other inductive learning algorithms, even if there is a slightly higher execution time.

Table 9. Precision Recall and accuracy (Measuring the performance of proposed model with baseline approach)

Metrics	GBA	Joint BPR Margin	&SkillBERT	SJRS	SJRS-ILR
Precision @5	60%	88%	81%	89.5%	91.2
Recall@10	54%	75%	78%	87%	89.9%
Accuracy	89%	91.24%	85%	92.0%	93.9%

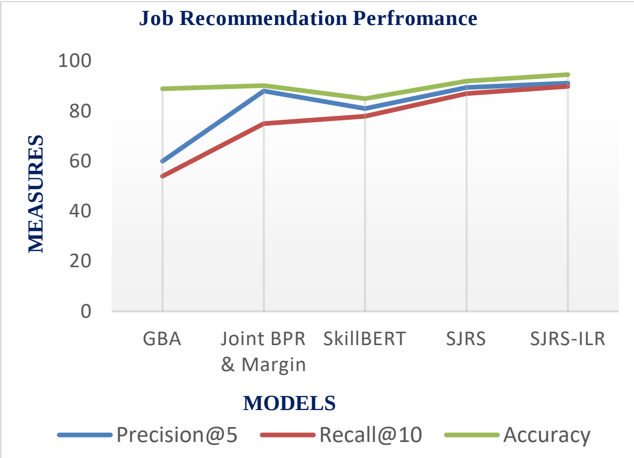


Fig 6. Job Recommendation Performance Comparison with baseline approach (GBA and Joint BPR & Margin and Skill BERT)

Results shown in the table below conclude that SJRS and SJRS-ILR improved the performance of the classification model over GBA and Joint BPR & margin. The use of XGBoost and Adams optimizer with SJRS based features gives 93.9% and 92.0% for the SJRS model.

6. Conclusion

The job recommendation system faces many problems like scalability and performance. To improve the system scalability, we need to design a scalable algorithm to handle massive datasets and varied datasets. We proposed a novel approach to replace distinct hidden values within datasets and machine learning techniques. Here we adapted the ILA inductive learning algorithm in our previous results that demonstrate the inductive effect of the algorithm. We adopt this approach by adding new features and creating new features and their scope to handle more datasets with missing instances. Adding this prominent feature to our proposed methodology will improve the system performance. The model's preliminary tests and comparative effectiveness will give better accuracy. It utilizes general approaches for exchanging the hidden values, including the MCV, the MCVRC, and the Delete strategy. Practical analysis shows excellent and favorable results for the proposed methodology regarding the generated rules' number and complexity. After applying modified ILA, an insignificant cost is inherent in creating the new process during inductive learning.

The proposed model's performance is measured using appropriate parameters in different datasets and satisfactory results. My future work is to use the distributed environment to improve the model performance.

Acknowledgments

The authors are grateful for the Thakur College of Engineering and Technology's support. This research did not receive specific grants from the public, commercial, or not-for-profit funding agencies.

References

[1] J.R. Quinlan. Learning efficient classification procedures and their application to chess end games, in: R.S. Michalski, J.G. Carbonell and T.M. Mitchell, eds., Machine Learning: An Artificial Intelligence Approach (Morgan Kaufmann, San Mateo, CA, 1983).

[2] Ammar Elhassan, Saleh M. Abu-Soud, Firas Alghanim, Walid Salameh, "ILA4: Overcoming missing values in machine learning datasets – An inductive learning approach", Journal of King Saud University – Computer and Information Sciences xxx (XXXX) xxx

[3] Mehmet R. Tolun, Saleh M. Abu-Soud, "ILA: an inductive learning algorithm for rule extraction", Expert Systems with Applications Volume 14, Issue 3, April 1998, Pages 361-370, https://doi.org/10.1016/S0957-4174(97)00089-4.

[4] Ravita Mishra, Sheetal Rathi, Enhanced DSSM (Deep Semantic Structure Modelling) Technique for Job Recommendation, Journal of King Saud University - Computer and Information Sciences, 2021, ISSN 1319-1578, https://doi.org/10.1016/j.jksuci.2021.07.018.

[5] Abu-Soud S. and Al Ibrahim A., DRILA: A Distributed Relational Inductive Learning Algorithm, WSEAS Transactions on Computers, Issue 6, Volume 8, June 2009, ISSN: 1109-2750.

[6] Oludag M., Tolun M., Sever H., and Abu-Soud S., "ILA-2: An Inductive Learning Algorithm for Knowledge Discovery", Cybernetics and Systems: An International Journal, vol. 30, no. 7, Oct.-Nov. 1999.

[7] Raja, P.S., Thangavel, K. "Missing value imputation using unsupervised machine learning techniques", Soft Comput. 24, 4361–4392. 2020, https://doi.org/ 10.1007/s00500-019-04199-6.

[8] Rashid W., Gupt, M.K., "A Perspective of Missing Value Imputation Approaches". In: Gao, X. Z., Tiwari, S., Trivedi, M., Mishra, K. (eds) Advances in Computational Intelligence and Communication Technology. Advances in Intelligent Systems and Computing, vol 1086. Springer, Singapore.2021, 10.1007/ 978-981-15-1275-9_25.

[9] Xingdog WU, "Inductive Learning: Algorithms and Frontiers" Department of Artificial Intelligence, University of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN, UK, Artificial intelligence Review 7, 93.—

[10] Saleh M. Abu-Soud and Sufyan Almajali, "ILA-3: An Inductive Learning Algorithm with a New Feature Selection Approach ", WSEAS Transactions on Systems and Control · January 2018.

[11] Le Wu, Yonghui Yang, Lei Chen, Defu Lian, Richang Hong, Meng Wang, "Learning to Transfer Graph Embeddings for Inductive Graph-based Recommendation ", SIGIR '20, July 25–30, 2020, Virtual Event, China, https://doi.org/10.1145/3397271.3401145.

[12] Ravita Mishra, Dr Sheetal Rathi, "Efficient and Scalable Job Recommender System Using Collaborative Filtering", Paprzycki M., Gunjan V. (eds) ICDSMLA 2019. Lecture Notes in Electrical Engineering, vol 601. Springer, Singapore https://doi.org/10.1007/978-981-15-1420-3_91.

[13] George V. Lashkia, Laurence Anthony, "An inductive learning method for medical diagnosis", Pattern Recognition Letters 24 (2003) 273–282, Received 30 October 2001; received in revised form 23 April 2002.

[14] Susan Dumais, John Platt, David Heckerman, Mehran Sahami, "Inductive Learning Algorithms and Representations for Text

Categorization", Proceedings of the seventh international conference on information and knowledge management. ACM.1998.

[15] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2018. Deep Learning based Recommender System: A Survey and New Perspectives. ACM Comput. Surv. 1, 1, Article 1 (July 2018), 35 pages. DOI: 0000001.0000001.

[16] Vachik S. Dave, Baichuan Zhang, Mohammad AI Hasan, Khalifeh Aljadda and Mohammad Korayem, "A combined representation learning approach for better job and skill recommendation", CIKM '18 ACM ISBN 978-1-4503-60149-2/18/10. DOI: 10.1145/3269206.3272023., ACM-2018.

[17] Pavlos Kefalas, Panagiotis Symeonidis, and Yannis Manolopoulos, "A Graph-Based Taxonomy of Recommendation Algorithms and Systems in LBSNs", IEEE transaction on knowledge and data engineering, Vol. 28, NO. 3, March 2016.

[18] Ece C. Mutlu, Toktam A. Oghaz, Amirarsalan Rajabi, Ivan Garibay, "Review of Graph Feature Learning and Feature extraction Technique for Link Prediction", 1901.03425v4 [cs.SI] 27 July 2020.

[19] Amber Nigam, Shikha Tyagi, Kuldeep Tyagi, Arpan Saxena, "SkillBERT: "Skilling "the BERT to classify skills!", ICLR 2021 Conference.

[20] Mahdi Jalili, Sajad Ahmadian, Maliheh Izadi, Parham Moradi, Mostafa Saleh, "Evaluating Collaborative Filtering Recommender Algorithms: A Survey" DOI: 10.1109/ACCESS.2018.2883742, Vol-6, IEEE Access, Nov 2018.

[21] Adrien Mogenet, Tuan-Anh Nguyen Pham, Masahiro Kazama, Jialin Kong. 2019. Predicting Online Performance of Job Recommender Systems with Offline Evaluation. In Thirteenth ACM Conference on Recommender Systems (RecSys' 19), September 16–20, 2019, Copenhagen, Denmark. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3298689.3347032.

[22] Charu C aggrawal, "Recommender system Textbook", ISBN 978-3-319-29657- 9 ISBN 978-3-319-29659-3, DOI 10.1007/978-3-319-29659-3, Springer International Publishing Switzerland 2016.

[23] Vedant Bhatia, P Rawat, A Kumar, RR Shah, "End-to-End Resume Parsing and Finding Candidates for a Job Description using BERT", arXiv preprint arXiv:1910.03089, Computer Science, Information Retrieval, 2018.

[24] Luca G. Cellamare_, Michele A. Bertoldi_, Alberto Parravicini†, Marco D. Santambrogio," Exploring transductive and inductive methods for vertex embedding in biological networks", 978-1-7281-3815-2/19/$31.00 ©2019 IEEE.

[25] A. Gugnani, V. K. Reddy Kasireddy and K. Ponnalagu, "Generating Unified Candidate Skill Graph for Career Path Recommendation," 2018 IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, Singapore, 2018, pp. 328-333, DOI: 10.1109/ICDMW.2018.00054.

[26] www.careerbuilder.com, last access, 2022.

[27] www.naukri.com, last access, 2022.