Automated Solution for Resume Analysis Using Machine Learning

***Abstract*— Today, the IT industry is very competitive, and it is becoming more and harder to find a good job position in a good company even for candidates with the right skills. When a vacancy is available for even a handful of positions, hundreds of candidates are lining up for the positions. So, the experts in the company have to spend their valuable time going through all those CVs to read, score, and choose. This research aims to build a solution that automates CV analysis, and grades and identifies potential candidates with the help of Natural Language Processing, Machine Learning, Data Mining, and TF-IDF Algorithm. With this solution, the IT industry can batch-process resumes and select potential candidates without wasting time on non-potential candidates.**

Keywords*— *Data Mining, Machine Learning, Natural Language Processing, TF-IDF Algorithm

I. Introduction

The next stage that comes in a person's life after finishing education is a job. There is still, a lot of people who start working before they finish their formal education. There is a curiosity when learning their first work as a new graduate. The reasoning is that most of them have no understanding of their careers. They don't know which work is most appropriate for their subjects, their talents, their extra-curricular interests, etc. However, it is not an easy option to choose a new career, which can depend on several variables, such as wage, job description, and geographical position. The most important item to represent the candidate when applying for work is the Curriculum Vitae (CV) or Resume. Around the same time, career hunting has gotten smarter and faster in this age of technology. However, there are more than enough candidates for a single position, and it is very difficult for an employer to pick a candidate.

Traditional recruitment methods have been used by employees across the world. There are numerous conventional recruiting strategies, such as paper ads, internal recruiting, referrals, and word-of-mouth, but these hiring, and recruitment strategies are simply not enough to get the skilled applicant. The procurement process has become more intelligent and simpler at the same time in this age of technology. However, for a single position, there are more than enough candidates, and it is very complicated and time-consuming for an employer to go through each resume and pick one candidate. There are automated recruiting systems to address this issue, where applicants have to submit their CVs / Resumes to a respected company and then the system classifies the resume according to company requirements [1], [2].

In recent years, hiring in the information technology industry has seen massive growth. Companies attract thousands of young talents, and it is difficult to recognize the college every year by campus fairs. A strong fit between the candidate's credentials and the ability a corporation needs by reviewing each HR department resume of either organization. Many firms have moved to using e-recruiting tools to meet this problem. The cost, time, and commitment required to manually process, and screen candidate resumes are minimized by these channels. In order to overcome the difficulties involved with sampling, matching, and classifying applicant resumes, these programs use multiple methods. While these methods create high precision ratios in identifying applicants to fill a vacancy, they pay less attention to the run time complexities of the resulting procedure, i.e., any work offer would be given less attention [3], [4].

If there is a vacancy in a company, they will notify the company by means of paper ads or allocate this task to the work-recruiting premises, and the jobseekers will forward those ads, and apply for a job, and submit the CV. CVs / Resumes submitted by work seekers used to be analyzed and evaluated manually by the employers. In the majority of companies, this approach is already followed. However, big companies often need to deal with many CVs / Resumes each and every day, Handling such a large amount of CVs / Resumes one by one has become very problematic and time-consuming to select candidates [1], [3], [4].

According to this problem, a solution was proposed that automates CV analysis, grade, and identify potential candidates with the help of Natural Language Processing, Sentiment analysis, Machine learning, Data mining, and TF-IDF algorithm. With this solution, the IT industry can batch process CVs / Resumes and select potential candidates without wasting time on non-potential candidates. The significance of this project is to save the time and money of the companies.

II. Literature Survey

This literature survey mainly focuses on the previous research on resume classification, ranking, findings, and features of their systems.

A proposed system was developed in 2019 where a semantic analysis technique is used. Two modules are used to develop this namely Natural Language Processing Pipeline and Module of Classification. Tokenization, Stop Word Deletion, POS Marking, and Object Identification are 4 steps in the first level. Tokens for classification will be created after the completion of these measures. It is possible to classify resumes according to requirements in the second stage by applying the classification algorithm method. This method classifies resumes in various structures according to the preferences of candidates. It is also easy for HR or concerned authorities to delegate projects according to their interests to candidates [5], [6].

A system was developed by Vishnunarayanan et al in 2017 where the resume screening process was conducted to qualify and disqualify the applicants according to business criteria. During the recruiting process, separate rounds are conducted to filter the eligible candidates in each round. To do the same, different filtering algorithms are used in resume screening. This primarily aims at minimizing the number of resumes in the corresponding/subsequent recruiting rounds, the meaning of making the procedure as cost-effective as possible was discussed. The study also says that this procedure can be like an acquisition, instead of cost. In automating the recruiting process, the biggest challenge encountered is that a resume does not explicitly specify how prolific the applicant can be. This may only be decided through human involvement which can be achieved by an interview. It may also be argued that there is a need for rigorous automation of the recruiting process, but there is a trade-off involved. It suggests that the resumes should be classified and screened and that the chosen applicants should be individually interviewed afterward so that the applicants are not totally elected [7], [8].

A framework that automates applicants' eligibility search and aptitude testing in a recruiting process was presented by Rout et al. in 2019. An online framework was created for the study of the aptitude or personality test and the CV of the applicant to satisfy this requirement. Based on the uploaded CVs, the device is analyzed for technical eligibility. A method of machine learning using the TF-IDF algorithm was used for this framework. A decision was provided for candidate recommendation and the performance of this method. In addition, the resulting scores help to determine the candidate's qualities by comparing the scores received in various fields. The graphical overview of any candidate's results makes it easy to assess his or her attitude and helps to better examine the CV. The framework also provides the recruiting process with a helping hand such that the candidate's CV is shortlisted and appropriate decisions. In this research, the sentiment analysis of the social media data of the candidates was not considered [9], [10], [11].

Research was conducted by Rakesh et al. in 2016 using NLP (Natural Language Processing) and ML (Machine Learning) to rank the resumes according to the given restriction. With this intelligent device, the resume was ranked in any format according to the client company's stated limitations or requirements. Basically, take the bulk of the client company's feedback resume and the client company will also have the criteria and restrictions according to this framework can rate the resume. In addition to the information given by the resume, The social accounts of candidates (such as LinkedIn, Github, etc.) can be read by this system and the most truthful information about the candidate can be provided [12].

According to a study that was conducted by Bajpai in 2020, the proposed model extracts required data from a CV / Resume and separates it on the basis of its values. However, the rating and positive weight of a CV / Resume can vary depending on the preference of various companies or employers. The whole process was segmented, and each segmented process was configured to execute its mission independently. In reality, the section that deals with Natural Language Processing operated only with the role of Natural Language Processing and equally with Computer Learning segments dealing exclusively with Machine Learning techniques. A particular approach is recommended to evaluate and analyze the data in a CV / Resume, and it was to translate data to HTML code to recognize different values. Finally, this model provides CVs / Resumes ranking based on the required details and takes prior requirements into consideration regarding employers [13], [14].

An approach for judging the resume of a career seeker was done by Chandola et al. in 2015. Sentiment Analysis was used for this approach and this model proposes to effectively shortlist by applicants based on their resumes according to the company's specifications. While the trustworthiness of a resume to shortlist, an applicant may be disputed, this is not the final procedure of the recruiting process of any organization [15], [16].

In 2023, The Classifier resume aims to make the resumes more robust for every job/university interview by using data extraction techniques focused on data from previously chosen and declined applicants which has been proposed by Metin et al. Data were collected from the resume by the device. Natural language processing (NLP) technologies were then used for data input parsing, tokenizing, stemming, and filtering. By using TF-IDF, based on the recruiter data, can be measured the score of the individual resume and suggest users' missing skills and recommend the top resume to the recruiter [17].

An approach to matching job seekers and job advertisements that combines a deductive matchmaking model based on description logic and a similarity-based ranking model was developed by Fazel-Zarandi and Fox in 2009. This approach can also be used to improve skills development Systems or Identification of structures within an organization. Domain ontology can be used to automatically annotate current information resources and execute automated reasoning to enhance the identification and retrieval of measures of competence. Another useful ontology in this regard is the organization ontology, which formalizes the organizational structure and can be used to infer capabilities and knowledge on the basis of the roles played by the agents and the communication among them [3].

The Company Recommender framework was developed in 2020 to assist recruiters with text mining and machine learning tools to identify the best job title candidate by Kelkar et al. Additional information was used according to their pattern, such as the projects in which the applicant was involved, as well as the summary of the project. This data will be collected as input

from the applicant and by analyzing the text used by him or her; the applicant has been classified into different levels of expertise. A knowledge base of various keywords that will form the basis for categorization will be designed [18].

In 2020, An automated Machine Learning model was developed for the resume short-listing process by Roy, Chowdhary, and Bhatia. The problem for developing this system was separating the proper applicants. Practically, CVs are not normal, and every resume has a different structure and format. The next one maps the CV to the job description to understand whether the applicant can do the job for which is employed. This model takes as input the characteristics extracted from the candidate's resume and finds their categories, further relies on the classified resume mapped on the required job description, and recommends the most fitting candidate profile to HR [19], [20].

III. Materials and Methodology

A resume analysis system has been developed here to easily find the correct candidate from a lot of resumes. In this research, a keyword-based methodology was used to analyze the resumes. Input is the resumes as PDF and the next step is keyword extracting and summarizing. The next clarifier the resumes using machine learning and finally the output is the analyzed resumes. This process is highly accurate because this method extracts the keywords gets the necessary contents matches the inputs and gives the correct solutions. Our proposed solution's main targets can be seen in the below diagram.

Fig. 1. Implementation Steps

The detailed design of our system can be seen in the below diagram.

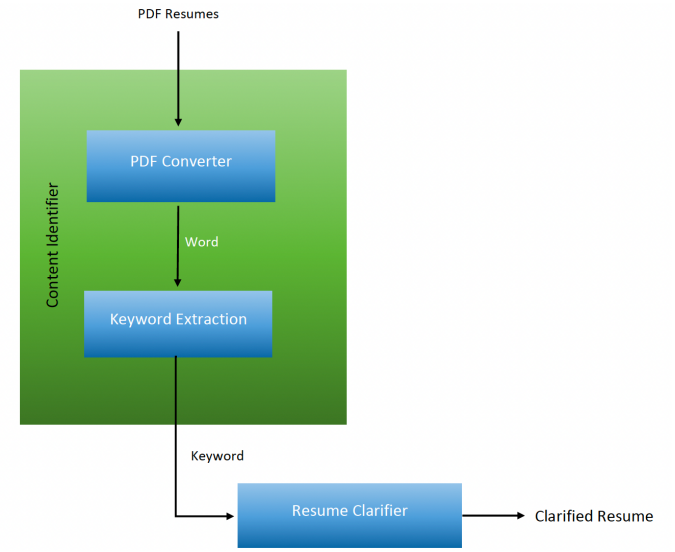

Fig. 2. Architectural diagram of the system

A. Content Identifier

There are two subtopics in the content identifier. They are PDF Converter and Keyword Extracting.

1) PDF Converter

First, the pdf converter was used to convert a pdf format CV to a Word file. Because it is very helpful to identify the content of the whole pdf file. The image processing part was used the get data from the pdf file. However, it was not very effective because a lot of resumes are created in different formats. So that reason it is very hard to find the content of the resumes. Finally, the method was chosen to convert the pdf file to a text file. The content of the resume can be identified easily after converting the pdf resume to a text file and it was very helpful to our next stage of keyword extracting.

2) Keyword Extracting

Firstly, The Word document was read the keyword extracting. The stop words and full stop and comma, etc. were removed. After the keywords were mapped to the resume clarification. For the Keyword extraction, the already identified keywords and topics were used to extract. In the keyword extracting there are three sub-topics.

First is Reading the converted Word document. Because before the prepossessing the Word document should be read. In this part, there are PDFDocument.python and PDFDocumentWordList.python files. PDFDocument.python can be seen below.

```
import PDFConverter as pd

def get_Document_Data_Set():
    text_D = pd.main()

    return text_D

def main():
    text_D=get_Document_Data_Set()
    return text_D
```

Fig. 3. PDFDocument.python file

The next part is the Preprocessing part of the text document. There are two files in the preprocessing part. They are Stopword.python and PreProssesor.python. The stopword.python file is depicted in Fig.4.

```python
from nltk.stem import PorterStemmer
from nltk.corpus import stopwords
import string
from nltk.tokenize import sent_tokenize, word_tokenize

def create_punctuation_list():
    punctuation_list = []

    for c in string.punctuation:
        punctuation_list.append(c)
    punctuation_list.append("''")
    punctuation_list.append('``')
    return punctuation_list


def create_stop_word_list():
    stop_word_list = []

    text_stopword_Name = []
    file_stopword_Name = open("StopWord.txt", mode='r')
    text_stopword_Name = file_stopword_Name.read()
    textWord_Tok_stopword_Name = word_tokenize(text_stopword_Name)
    for word in textWord_Tok_stopword_Name:
        stop_word_list.append(word)
    return stop_word_list
```

Fig. 4. Stopword.python file

The final step of the Keyword extraction is the Content identifier. In this part, the keywords of the input Word document were printed. The ContentIdentifier.python file is depicted in in Fig.5.

```python
import PDFDocumentWordList as Dw
import PreProcessor as Prp


textSent_Tok_Que = Dw.main()

key_words = Prp.ignoring_unwanted_words(textSent_Tok_Que)


def printTheKeyWord():
    for key in key_words:
        print(key)


printTheKeyWord()
```

Fig. 5. ContentIdentifier.python file

B. *TF-IDF Algorithm*

The TF-IDF algorithm was used for the keyword extraction. TF-IDF means "Term Frequency—Inverse Document Frequency." This is a technique for quantifying a word in documents. The weight of each word was usually calculated, which means the value of the word in the document and the corpus. This method is a commonly used technique for information retrieval and text mining. The computer can only recognize any data in the form of a numerical value. So, for this purpose, all the text was vectorized so that the computer could understand the text better.

TF-IDF = Term Frequency (TF) * Inverse Document Frequency (IDF)

1) *Term Frequency*

Term Frequency means the frequency of a word in a text. This mostly depends on the length of the document and the generality of the term, for example, a very common word such as "a" may appear several times in the document. But if take two texts, one with 100 words and the other with 10,000 words. There is a high possibility that a common word such as "a" will be more present in the 10,000-word text. But we cannot assume that the longer document is more valuable than the shorter one. For this very purpose perform a normalization of the frequency value and divide the frequency with the total number of words in the text. To vectorize the documents, first check the count of each term. In the worst case, if the phrase does not appear in the document, the particular TF value will be 0 and in the other extreme case, if all the terms in the document are the same, it will be 1. The final value of the normalized TF value will be in the range of [0 to 1]. 0, 1 inclusive of this.

t - term(word)
d – document(set of words)
N – count of corpus
corpus – the total document set
TF (t,d) = count of t in d / number of words in d

2) *Document Frequency*

This measures the value of the text in the whole set of corpus, This is similar to TF. The only difference is that the TF is the frequency counter for the term t in document d where the DF is the number of occurrences of the term t in document N. DF is the number of documents in which the word appears.

df(t) = occurrence of t in documents

3) *Inverse Document Frequency*

The inverse of the frequency of the document which measures the informativeness of the term t. When the measure IDF, it will be very low for the most common terms like stop words (because stop words like "is" are present in almost all documents, and N/df will offer a very low value to that word).

idf(t) = N/df

When a word that is not in the vocabulary occurs, this time the df will be 0. So that time adding 1 to the denominator.

idf(t) = log(N/(df + 1))

Finally, the TF and IDF values were taken and those Values were multiplied and the TF-IDF score was calculated.

tf-idf(t, d) = tf(t, d) * log(N/(df + 1))

C. *Resume Clarifier*

For the resume clarifier, a model was created for training our dataset. Data preprocessing is one of the main steps in every problem in data analysis, as it ensures that the model is correct because it depends largely on data quality. Data collection, selection, and integration, as well as data cleaning (filling in

missing data, minimizing noise in the data, and removing conflicting data), data transformation (normalizing data, discretizing / aggregating data, and creating new attributes), and data reduction (reducing the number of variables, reducing the number of cases, and balancing skewed data) entailed by it. The preprocessed data (well-formed data) can then be fed into a machine-learning algorithm after these steps are completed correctly. Two machine learning algorithms were used for the training of our dataset. They are the Support Vector Machine algorithm and the Decision Tree algorithm. Train our dataset from the above algorithms has been given high accuracy from the SVM algorithm.

D. Data and Data Description

For the data collection, data were collected from LinkedIn profiles, and scraping tools were used for that process. That data was downloaded and it was arranged into our dataset. As well as the dataset from Kaggle was downloaded. The all data were not arranged. They were separated by columns and the dataset was prepared. The Description, Job Title, Skills, and Experience Years were included in that dataset, and the level was added to their profiles. Next stage, the dataset was preprocessed. This was the main step in the data analysis. Because that was most important to train our dataset. From that, good accuracy can be obtained from this data. The data preprocessing parts consisted of data integration (collecting data, selecting data, integrating data), data cleaning (filling in missing data, eliminating inconsistent data), data transformation (normalizing data, discretizing / aggregating data), data reduction (reducing the number of variables, reducing the number of cases and balancing skewed data). After those preprocessing steps, our machine learning algorithms were added to the train.

IV. Results and Discussion

The dataset was trained using two machine-learning algorithms for the experimental results. They are the Support Vector Machine algorithm and Decision Tree algorithm. The model was trained by changing the dataset and as well as changing the algorithm. From those algorithms, each algorithm was tested using 1000 datasets and 10000 datasets. From those datasets, various accuracies have been given for each algorithm. Testing with the SVM algorithm, the accuracy of 37% and 72% from 1000 and 10000 datasets have been given. As well as the Decision tree algorithm the accuracy of 21% and 64% from 1000 and 1000 datasets have been given. The Expected results of accuracy from our training dataset can be seen below table.

TABLE I. Accuracy of Dataset

Algorithm Used	Volume of Dataset	
	1000	10000
Support Vector Machine	37%	72%
Decision Tree	21%	64%

The results of accuracy when training the model changing the algorithm and dataset volume can be seen below in Fig.6.

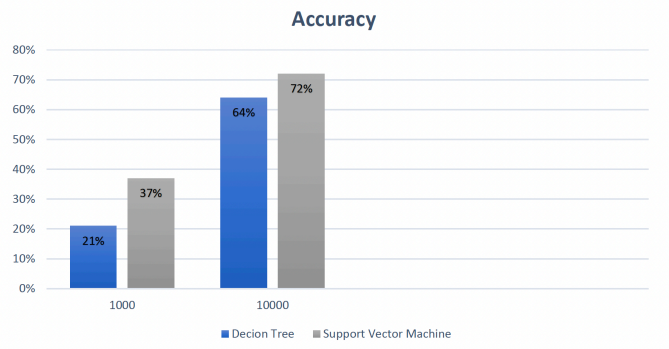

Fig. 6. Accuracy of the dataset

From that, there is a high volume of the dataset in which more accuracy was given, and in the low data set the low accuracy was given. In the above table, the Support Vector Machine algorithm has worked well when training our dataset and more accuracy has been given than the Decision Tree algorithm. So, the Support Vector Machine was used for our work.

In this CV analyzing system, some algorithms and machine learning algorithms to implement the system have been used. For the training, in our machine learning model, the 10000 of a dataset was used and 72% accuracy was given. But when the dataset capacity becomes high, and accuracy of this system is increased. As well as the different functionalities can be used to increase accuracy and testing for our model training in the Support Vector Machine algorithm. To do that the keyword extraction process of our system can be more developed using the TF-IDF algorithm functionalities and Natural Language Processing. So that different methods can be used to develop more accuracy in the keyword extraction process. After clarifying resumes, the output was given as the level of the resume. The Low Value, Medium Value, and High Value were given as output. Low value means he has low technical knowledge and low experience. Medium value means Moderate technical skills and moderate working experience. High value means he has a high knowledge of technical skills and more years of experience. That is the main output of our resume clarifier.

V. Conclusion

This Resume analyzing system is developed by keyword extracting and machine learning algorithms. In this research, the main parts are content identifier and resume clarifier. In the content, the Identifier has two sub-parts. They are PDF Extraction and Keyword Extraction. The PDF resume is converted to a Word document by the PDF Extraction algorithm. Next is the Keyword Extraction part. In this part, the converted Word document is gotten and the output of keywords is gotten. In this step, the Natural Language Processing and TF-IDF are used for keyword Extraction. The output of our

Keyword Extraction is the keyword of the resume. In this research, the next main part is the Resume clarifier. In this step, the dataset is trained using the Support Vector Machine and Decision Tree algorithms. It has been given more accuracy in the Support Vector Machine algorithm. So, SVM is used for the Model Training. The next step is analyzing the resumes. For that, the output of content identifier keywords is input to our model and the resumes are analyzed for the given inputs. This research has been completed successfully to build a solution that automates CV analysis, and grades and identifies potential candidates with the help of Natural Language Processing, machine learning, data mining, and TF-IDF algorithm. The significance of this project is to save the time and money of the companies.

REFERENCES

[1] J. W. N. K. J. M. S. M. and M. P. , "Resume Classification using Machine Learning," *International Journal for Research in Applied Science & Engineering Technology (IJRASET),* vol. 7, no. XI, pp. 423-425, 2019.

[2] S. T. Gopalakrishna and V. V. , "Automated Tool for Resume Classification Using Sementic Analysis," *IJAIA,* 2019.

[3] M. F.-Z. and M. S. F. , "Semantic Matchmaking for Job Recruitment: An Ontology-Based Hybrid Approach," 2009.

[4] M. G. M. P. and C. B. , "Adaptability in an agent-based virtual organisation," *International Journal of Agent-Oriented Software Engineering,* vol. 134, pp. 111-120, 2009.

[5] S. T. Gopalakrishna and V. V. , "AUTOMATED TOOL FOR RESUME CLASSIFICATION USING SEMENTIC ANALYSIS," *International Journal of Artificial Intelligence and Applications (IJAIA),* vol. 10, no. 1, pp. 11-23, 2019.

[6] Y. Pawar and S. H. Gawande, "A Comparative Study on Different Types of Approaches to Text Categorization," *International Journal of Machine Learning and Computing,* vol. 02, no. 04, 2012.

[7] R. Vishnunarayanan, P. Shreekrishna , A. Krishnan., S. Palanivel and A. Umamakeswari, "RESUME CLASSIFICATION USING ANALYTIC HIERARCHY PROCESS AND KEYWORD EXTRACTION," *International Journal of Pure and Applied Mathematics,* vol. 115 , no. 07, pp. 43-48, 2017.

[8] C. A. Bayraktar and M. Ozbek, "Cost-Benefit Analysis of the Hiring Process," in *Asia-Pacific Power and Energy Engineering Conference*, Wuhan, 2011.

[9] S. Bagade, J. Rout and P. Yede, "Personality Evaluation and CV Analysis using Machine Learning Algorithm," *International Journal of Computer Sciences and Engineering,* vol. 07, no. 05, 2019.

[10] F. Ahmed , M. Anannya and T. Rahman , "Automated CV Processing along with Psychometric Analysis in Job Recruiting Process," 2015.

[11] A. Palve, R. Sonawane and A. Potgantwar, "Sentiment Analysis of Twitter Streaming Data for Recommendation using, Apache Spark," *International Journal of Scientific Research in Network Security and Communication,* vol. 05, no. 03, pp. 99-103, 2017.

[12] J. A. A. Zubeda , M. A. A. Shaheen, G. R. N. Godavari and S. S. Z. M. S. Naseem , "Resume Ranking using NLP and Machine Learning," Mumbai, 2016.

[13] S. Bajpai, S. Mamgai and M. Sainger, "Analyzing Resume using Natural LanguageProcessing Machine Learning and Django," *International Journal for Research in Applied Science & Engineering Technology (IJRASET),* vol. 08 , no. V , pp. 2037-2039, 2020.

[14] S. Sanyal, S. Hazra, S. Adhikary and N. Ghosh, "Resume Parser with Natural Language Processing," *International Journal of Engineering Science and Computing,* vol. 07, no. 02, pp. 4484-4489, 2017.

[15] D. Chandola, A. Garg, A. Maurya and A. Kushwaha, "ONLINE RESUME PARSING SYSTEM USING TEXT ANALYTICS," *Journal of Multi Disciplinary Engineering Technologies (JMDET),* vol. 09, no. 01, 2015.

[16] U. Marjit, K. Sharma and U. Biswas, "Discovering Resume Information Using Linked Data," *International Journal of Web & Semantic Technology,* vol. 03, no. 02, 2012.

[17] E. Aydın and M. Turan, "An AI-Based Shortlisting Model for Sustainability of Human Resource Management," vol. 15, 2023.

[18] B. Kelkar, R. Shedbale, D. Khade, P. Pol and A. Damame, "Resume Analyzer Using Text Processing," vol. 11, no. 05, pp. 353-365, 2020.

[19] P. K. Roya, S. S. Chowdhary and R. Bhatia, "A Machine Learning approach for automation of Resume Recommendation system," *International Conference on Computational Intelligence and Data Science,* vol. 167, p. 2318–2327, 2020.

[20] A. Otaibi and S. Ykhlef, "A survey of job recommender systems," *International Journal of Physical Sciences,* vol. 07, p. 5127–5142, 2012.