A Hybrid Approach for Resume Classification Using Machine Learning and Advanced NLP Techniques

1st Pradeep raj D
Dept. of Computer Science and Engg.
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Chennai, India
pradeepraj.dhandapani@gmail.com

2nd Rahul K
Dept. of Computer Science and Engg.
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Chennai, India
rahulbio789@gmail.com

3rd Kanimozhi N
Dept. of Computer Science and Engg.
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Chennai, India
05kanimozhi20@gmail.com

4th Shalini D
Dept. of Computer Science and Engg.
Amrita School of Computing
Amrita Vishwa Vidyapeetham
Chennai, India
shalinidheenadhayalan@gmail.com

5nd M Ranjith Kumar*
Department of Mathematics
Amrita School of Engineering
Amrita Vishwa Vidyapeetham
Chennai, India
annam.ranjith@gmail.com

***Abstract*—In the present competitive labor market, the amount of resumes organizations receive frequently needs to be increased in the traditional recruitment processes. Manual resume screening is time-consuming, resulting in inefficient candidate selection. To overcome these issues, this research provides a novel framework for resume categorization that employs machine learning (ML) approaches. Using ML techniques boosts the efficiency of resume classification by swiftly and precisely processing many resumes. The performance of the model is further increased by the inclusion of advanced NLP models like Bidirectional Encoder Representations from Transformers (BERT), which helps in capturing complex phrases and text embeddings. The trained model is evaluated using metrics that include Recall, Accuracy, Precision, and F1 score. The accuracy of 99.48% by the machine learning model proves the efficient classification of resumes. Using these techniques enhances resume classification and lowers the burden on recruiters, thus helping recruiters in their decision-making.**

***Index Terms*—Resume Classification, Machine Learning, Pre-Trained Transformers, BERT**

I. Introduction

Resumes are essential in the process of recruitment in the present world. The main way that applicants present their abilities, backgrounds, and credentials to potential employers is through resumes. But the vast number of resumes received in the current fast-paced job market may overwhelm employers, rendering traditional screening ineffectual and impossible [1]. For a recruiter, identifying worthy applicants from thousands of applications can be a difficult task, which significantly increases the workload of the recruiter [2]. There is no standard resume template which is present in the market, where each resume is in a unique format. Thus, it is a difficult task for HR to manually screen through the resumes, which are of different templates. Manually examining resumes takes a lot of time. To be able to assess a candidate's qualifications, skills, and experiences in relation to the job specifications, recruiters must carefully examine each resume. This process can take a long time, taking time away from other essential recruiting initiatives. Since there are a lot of applications to go through, it's possible to miss out on good applicants or misinterpret their qualifications. As a result, there may be missed opportunities to find top talent or incorrect estimations of the candidate's fitness for the role. These are some of the obstacles that recruiters face when their resumes are reviewed manually. Therefore, it is essential to offer a system that ensures efficient screening of resumes in the current professional environment.

Application of Machine Learning(ML) and Natural Language Processing(NLP) techniques improves the efficiency and accuracy of screening the resumes [3]. The use of machine learning techniques ensures no deserving candidate is missed out, and significantly reduces the overload of the recruiters. Datasets of resumes that are labeled are used in Machine learning algorithms to train the model to find trends and patterns that indicate successful candidates. NLP techniques, which play a crucial role in the classification process, are used in extracting meaningful and useful information such as Job roles, key skills, and history of education.

ML models can adjust and learn from the feedback, thus increasing the overall efficiency. A hybrid approach, where a pre-trained transformer model such as Bidirectional Encoder Representations from Transformers(BERT), [4], which

is known for context comprehension and sentence embedding, can be used for feature encoding combined with ML and NLP techniques to increase the performance of the models. BERT's pretrained language understanding enhances the potential of ML algorithms which allows for more accurate screening of resumes. This automated screening process reduces the burden of recruiters and gives them space to focus more on strategic responsibilities which involve conducting interviews and personal communication. This will eventually result in equally improved results for employees and applicants, as well as better-informed hiring decisions.

The sections that follow are as follows. A thorough literature review regarding present developments in the automation process of resume classification is given in section 2. The results of the experiments conducted with various machine learning models are explained in Section 3, whereas Section 4 summarizes the research project and outlines its future directions.

II. Literature Review

Riya Pal Shahrukh and co-authors [5] proposed the research on resume classification by using different Machine Learning Algorithms. TF-IDF Vectorization is used in the paper which helps in finding the important word in a set of words. The Algorithms used include Naive Bayes, Random Forest(RF), and Support Vector Machine(SVM). The effectiveness of these algorithms is explored based on classifying resumes accurately. The performance of these algorithms is evaluated based on metrics, which include precision, accuracy, recall, and F1 score to determine their efficiency in classification. The accuracy of the NB classifier, SVM, and Random Forest are 45%, 60%, and 70%, respectively. The end result of the confusion matrix shows that Random Forest has the best scores among others. Utilization of these techniques leads the authors to enhance the resume screening process. This research contributes to aiding the recruitment process for resume analysis using Machine Learning.

Suhas Tangadle Gopalakrishna et al. [6] use a tool that employs ensemble learning, which is based on voting classifiers in classifying candidate profiles into suitable domains based on their interests and expertise. This research uses two modules, the Natural Language Processing Pipeline (NLPP) and the Classification module. In NLPP, the resume is fed in which the unnecessary information is eliminated leading to data in token form. Then, the classification module analyzes the tokens to classify them into suitable domains using ensemble learning-based voting classifiers. It utilizes topic modeling techniques to introduce new domains and calls Stack-Overflow REST APIs. The efficiency of prediction is made and the machine learning domain has the majority vote share. Overall, the tool aims to enhance productivity in software companies by automating routine tasks through AI and machine learning.

Pradeep Kumar Roye et al. [7] proposed the research works to automate the resume recommendation process for job seekers by using a machine learning-based model. The challenges of screening a large number of resumes and matching them to job descriptions has been addressed in this paper. The model contains two main components, Classification and Recommendation. In the classification phase, the model utilizes ML models such as Logistic Regression, Linear SVM, and Random Forest to categorize resumes into different domains. The evaluation to find the performance of the classifiers is done by 10-fold cross-validation, which gives the highest accuracy in Linear SVM classifiers. The recommendation phase employs two approaches: A content-based recommendation system and k-nearest Neighbours(KNN). The Content-based recommendation matches the resume's job description and computing the data utilized using cosine similarity. Then, k-nearest neighbors are used to match the nearest resumes to the job descriptions. The proposed approach shows promise for improving the effectiveness of the recruitment process, with potential for further enhancement using deep learning models and industry-specific customization.

Shabna Nasser et al. [8] delve into the approach for classifying resumes into different categories. After preprocessing the data files, the pre-trained Glove-100 dimensional word vector is used to represent the word feature vectors. The embedded words are fed to the CNN deep learning network for classification. The CNN model is comprised of 8 layers with an input layer, two 1D-convolutional layers, an embedding layer, one max-pooling layer, a dropout layer, and a dense layer. It uses a hierarchy of classification levels and evaluates the performance using precision, f-score, and recall. The proposed system is designed to classify resumes into technical and non-technical domains, further categorizing them into specific job classes. The system consists of four major modules: extracting plain text, preprocessing, feature representation, and classification. Two datasets, resumes, and job descriptions, are used for training the models. The results show promising performance with training accuracies ranging from 90% to 99% and testing accuracies ranging from 87.9% to 96%. Future research directions include expanding the sample datasets, adding more data classes, and using custom word embedding models with more deep neural networks.

Luiza Sayfullina et al. [9] classified 523 resumes 98 essays and 90,000 job descriptions using Indeed Job Search API written by children about their dream job into 27 categories. Three methods were employed- fast text classifier and convolutional neural networks for sentence classification. The fast text model gave 71.99% accuracy for the Job Description, 33.40% accuracy for resumes, and 28.5% accuracy for the children's essays. On the other hand, the CNN model had 74.88%, 40.15%, and 51.02% accuracy, respectively.

Mehul Patel et al. [10] use resumes stored by employees in a dedicated database. Natural language preprocessing techniques, which include tokenization and stop word removal, are done to the resume database. Relevant information is extracted from the database using NLP techniques. The system uses machine learning techniques to match preprocessed resumes with employers' required job descriptions. The model gives the top 3 resumes as a result and displays the percentage match of the resumes to the employer's requirements.

Shubham Bhor et al. [11] propose a job portal where job seekers can upload their resumes. Their resumes will be parsed using NLP techniques and extract important information such as education, experience, project, address etc. A structured information resume will be generated and the resume will be ranked according to the skills of the applicant and the requirements of the company. The proposed job portal consists of three modules, namely, the employee module, the admin module, and the company module.

III. Methodology

Machine learning combined with advanced Natural Language Processing techniques are implemented for the efficient process of classification. The steps involved are as follows:

A. Data Description

The dataset was acquired from Kaggle, which is the largest data repository for machine learning applications. The dataset consists of 962 resumes which were analyzed and labelled into 25 distinct job fields. The content of the resume includes skills, Tools and Technologies known, Years of Experience, Projects, and the details of the Company. The number of resumes in each category can be visualized easily using the matplotlib library and is shown in Fig 1.

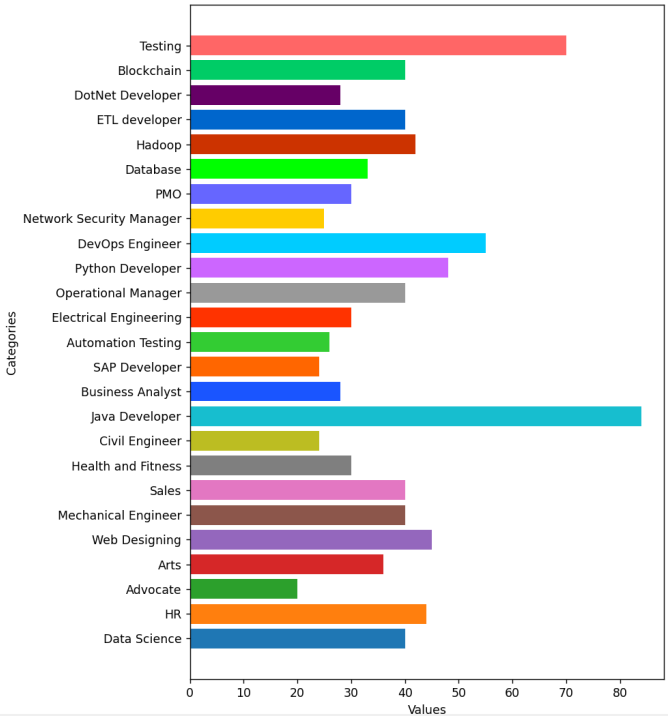

Fig. 1. Job description and their application counts

B. Data Preprocessing

In the process of data preparation, initial unprocessed data is converted into relevant data for machine learning. This involves cleaning the raw text, especially in text classification, by removing superfluous or unnecessary information, filling in missing values, and standardizing the data to guarantee consistency and comparability. Considerable efforts were made to efficiently preprocess the textual material in order to get ready for the Resume Classification assignment. With respect to our dataset, which has resumes, Several steps are involved in cleansing, which includes the removal of URLs to prevent them from interfering with the classification process. This is done using Regular expressions(regex) patterns. Non-alphabetic characters, other than certain symbols, are too excluded from using these patterns. All text in the dataset is converted from uppercase to lowercase for easier classification processes. Next, the text is split into tokens in which the words are broken down into small units for efficient analysis. A process called stemming is introduced, where words are reduced to their root elements. For instance, the words ”running” and ”runs” are stemmed down to their root word, which is ”run”. This is done with the help of Porter Stemmer [12], a popular algorithm used for the stemming of English words. Additional processes include removal of stop words, Addressing missing values to eliminate bias, removal of redundant white spaces, standardizing the text, and enhancing the overall quality of the text.

Various preprocessing techniques mentioned above can be effectively performed with the help of the Natural Language Toolkit(NLTK) library [13]. It provides tools for several processes, such as tokenization, stop word removal, stemming, etc. By streamlining the preprocessing pipeline, NLTK makes text analysis and modeling more productive and successful.

C. Text embedding

Text embedding is a crucial part of NLP tasks such as semantic similarity analysis, text classification, and information retrieval, which involves handling text segments like paragraphs and sentences. It is a process of mapping these text segments into high-dimensional vector space. This numerical representation allows for the comparison of text using mathematical operations. For our approach, we have used the modules of Bidirectional Encoder Representations from Transformers (BERT) [4] model as transformer-based models are very proficient in preserving the contextual information of a word in large text segments.

After all the pre-processing steps mentioned in the above section. The text embedding is implemented by two separate TensforFlow saved models. Initially, The processed text data is fed to the selected preprocessing model(bert-en-uncased-preprocess) from TensorFlow HUB which tokenizes and transforms the text data into fixed-length numeric tensors. These tensors are then, fed into the pre-trained encoder module of BERT (en-uncased-l-12-h-768-a-12)[4] as inputs. This encoder returns the tensors with content-aware embeddings of each token. these embeddings provides condensed, high-dimensional semantic representations of the resumes. A pandas dataframe is used to save the obtained embeddings in a structured manner. Each row in the dataframe is represented by the processed resume and each column in the dataframe is represented by the features extracted from the BERT model.

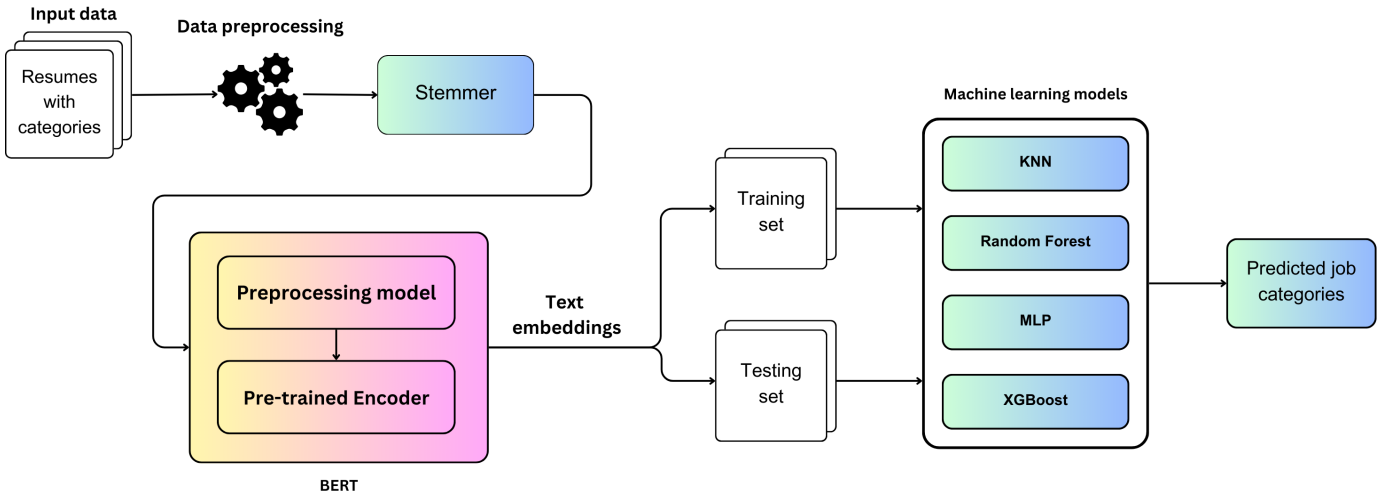

Fig. 2. Proposed Methodology for the classification of resumes

D. Model Selection

The BERT text embeddings are obtained using the processes mentioned in the previous section. These embeddings and the corresponding target labels are split into training and testing sets in the ratio of four to one. (ie. 80% for train and 20% for test) using Scikit-learn library. Few supervised machine learning models were chosen for the task of classifying the obtained embedding vectors of the resumes. For the experimental setup, the following classifiers are used :

1) K-Nearest Neighbours: K-Nearest Neighbours(KNN) is the simplest machine learning algorithm among the other chosen models for the classification task. where a neighbor's k closest neighbors' majority class is used to assign a label. It is also known as a lazy learner classifier since it uses the simplest method of a Euclidean algorithm for classification purposes. [15] The K parameter is set to 5, which establishes the number of closest neighbors, which helps to obtain the ideal value for properly categorizing resumes into the right work categories.

2) Random Forest Classifier: Many decision trees are generated using Random Forest(RF), a supervised algorithm that outputs the class that receives the majority votes from individual decision trees. [16]. The usage of majority voting to combine predictions of individual decision trees ensures a resilient and reliable classification outcome. Bagging, an approach used in Random Forest, increases the accuracy of classification as it involves training many models on random subsets of the training data. Another approach, stacking, is used in RF, where the predictions of one model are given as input to another model, which eventually tends to increase the performance. Thus, the use of Random Forest makes the process of classifying resumes more reliable for HR and the recruitment team.

3) MultiLayer Perceptron Classifier: Multilayer perceptron (MLP) Classifier is a simple neural network. where the input and output layers are separated by several hidden layers [17]. The neurons are structured in layers. it is a type of feed-forward neural network where the connections are directed only in one direction from lower to upper layers. There are no connections between the neurons in the same layer. In the input layer of the neural network, the number of neurons is the same as the number of measurements of the resume. and the output layer contains the same number of neurons as the number of job labels. The MLP classifier model was configured with two hidden layers containing 100 and 50 neurons, respectively, and the output layer contains 25 neurons. Then the activation function used in these hidden layers is Rectified Liner Unit(ReLU). and for the solver algorithm, Adam has been used to optimize the weights of the networks during training. and The text embeddings were standardized before feeding into the MLP classifier.

4) eXtreme Gradient Boosting: Extreme Gradient boosting (XGBoost) is a scalable implementation of gradient boosting, which is a supervised machine learning technique. Gradient boosting is a technique that involves the sequential combination of multiple decision trees; these decision trees are constructed, with each tree's training dependent on trees that are previously trained [18]. Given a predefined objective function, each tree is actually trained to predict the pseudo-residuals of the tree that came before. Each tree contributes its residual to the overall outcome when using inferencing for new instances in an additive way. XGBoost introduces improvement upon traditional gradient boosting methods. by incorporating a regularization component to the loss function. The XGBoost classifier was set up for multi-class classification, and the softmax function is used to find probability distribution over multiple classes. The target classes were label-encoded as the model is not natively compatible with string labels.

E. Evaluation metrics

For verifying the effectiveness of these supervised machine learning models in classifying the BERT text embeddings.

Four model evaluation metrics were used: Accuracy, Precision, Recall, F1-Score [19]. These metrics are discussed in detail below

1) Accuracy: The percentage of correctly identified examples, or predictions, out of all the instances in the dataset is called accuracy. The formula used to calculate accuracy is given as follows.

$$\text{Accuracy} = \frac{(TotalNegatives + TotalPositives)}{\text{Total Predictions}} \quad (1)$$

where,

$$\text{Total Predictions} = (TN + TP + FN + FP)$$

True Positives(TP) represent the case where the model correctly identifies the resume belonging to a specific class. True Negatives(TN) covers instances where the model accurately detects resumes that do not fit into a particular class. False Positives(FP) denotes the case where the model falsely classifies a resume as belonging to a specific class. Conversely, the resumes that are classified incorrectly that they do not belong to a specific class are represented by False Negatives(FN).

2) Precision: The precision of the model is defined as the ratio of True positives (TP) to the number of Total positives (TP+FP).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2)$$

A high precision score indicates that there are less resumes in the group which are labelled as False positives

3) Recall: Recall, also known as sensitivity is used to measure the original positive cases identified by the model and can be calculated by the formula given below.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

A good recall score ensures that the model identifies relevant resumes effectively

4) F1-Score: F1 Score is the harmonic mean of recall and precision and can be calculated by using the below formula.

$$\text{F1 Score} = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (4)$$

A high F1 score denotes that the model efficiently handles recall and precision, thus indicating resilient resume classification.

IV. Results

The Experimental outcomes of the above-mentioned models will be discussed in this section. Table 1 presents the Recall, Precision, F1 score, and accuracy of the four different models used for the classification of resumes.

Among the four models evaluated, KNN scored the least accuracy of 87.5%. It shows some limitations in accurately classifying resumes, especially for job categories such as Advocate and Health and Fitness, as they had fewer instances. Random Forest classifier displays a significant improvement over KNN in overall performance, achieving an accuracy of

TABLE I
Evaluation of Performance

Model Trained	Precision	*Recall*	*F1-Score*	*Accuracy%*
KNN	0.896	0.875	0.864	87.5
RandomForest	0.986	0.984	0.983	98.4
MLP	0.995	0.994	0.994	99.44%
XGBoost	0.995	0.994	0.993	99.48%

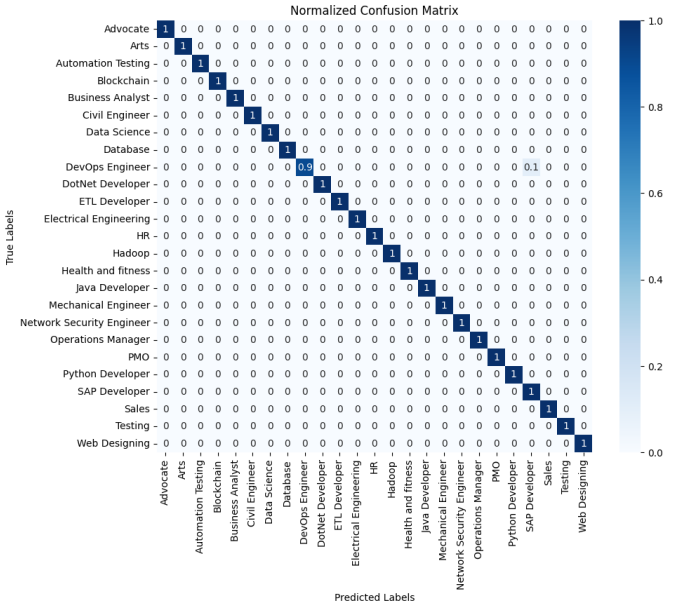

Fig. 3. Normalized confused matrix of XGboost Classifier

98.45%, and other evaluation metrics are also consistently high across most classes. MLP classifier outperforms both KNN and RandomForest. it attains an accuracy of 99.44%, illustrating its ability to capture meaningful features from fewer occurrences. XGBoost Classifier algorithm is found to have the highest accuracy of 99.48% and exhibit high precision, recall, and F1-score across all classes. It slightly improves upon MLP in terms of precision. This reflects the reliability of using this algorithm for the successful classification and categorization of resumes. It is vital to take into consideration the findings given by the confusion matrix while evaluating the results of the categorization methods. [20]

Fig. 3 shows the Normalized confusion matrix comparing the predicted categories and their actual categories of the resumes obtained from the XGBoost Classifier. It indicates that the model demonstrates excellent performance in classifying all the job categories. A normalized confusion matrix based on the results of KNN, which performed the least among the chosen models, is shown in Fig. 4. It clearly shows that the algorithm struggled to find features in classes with fewer instances. These results show that these supervised machine learning models have performed well in classifying the text data derived from resumes. without the need for complex deep learning models, which generally require more computation.

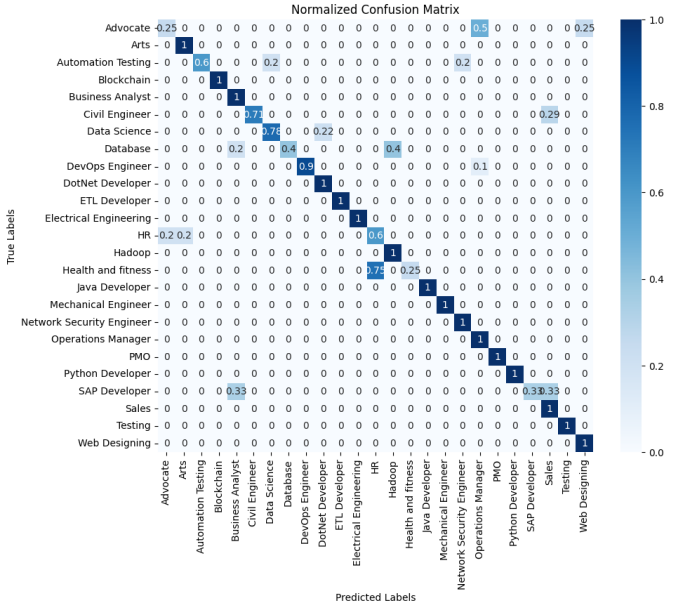

Fig. 4. Normalized confusion matrix of KNN Classifier

V. Conclusion

The resume classification significantly reduced the manual human effort and achieved a higher level of accuracy and automation with minimal human intervention. The task of resume classification automatically categorizes resumes or CVs into predefined domain categories or classes based on their content. This task is essential for the job recruitment process, particularly when organizations receive a large number of applications for various positions. Through this analysis, the proposed language model BERT embedding as features enhances the classification accuracy efficiently. Using the vectors that are obtained from BERT, various machine learning algorithms were implemented, where the XGBoost Algorithm had the highest accuracy of 99.48%. This demonstrates the application of BERT, NLP techniques and machine learning approaches to automate the process, thus increasing the efficiency and accuracy in the process of successful classification of resumes. Our future work focuses on experimenting with more recent transformer models for generating embeddings and exploring more domain-related tasks, which include slot filling by analyzing the information given in the resumes.

References

[1] Koyande, Bhagyashree Anilkumar, et al. "Predictive Human Resource Candidate Ranking System." International Journal of Research in Engineering, Science and Management 3.1 (2020)

[2] Al-Otaibi, Shaha T., and Mourad Ykhlef. "A survey of job recommender systems." International Journal of the Physical Sciences 7.29 (2012): 5127-5142.

[3] Lin, Yiou, et al. "Machine learned resume-job matching solution." arXiv preprint arXiv:1607.07657 (2016).

[4] Gomes, Luiz, Ricardo da Silva Torres, and Mario Lúcio Côrtes. "BERT- and TF-IDF-based feature extraction for long-lived bug prediction in FLOSS: A comparative study." Information and Software Technology 160 (2023): 107217.

[5] Pal, Riya, et al. "Resume classification using various machine learning algorithms." ITM Web of Conferences. Vol. 44. EDP Sciences, 2022.

[6] Gopalakrishna, Suhas Tangadle, and Vijayaraghavan Vijayaraghavan. "Automated tool for Resume classification using Sementic analysis." International Journal of Artificial Intelligence and Applications (IJAIA) 10.1 (2019).

[7] Roy, Pradeep Kumar, Sarabjeet Singh Chowdhary, and Rocky Bhatia. "A Machine Learning approach for automation of Resume Recommendation system." Procedia Computer Science 167 (2020): 2318-2327.

[8] Nasser, Shabna, C. Sreejith, and M. Irshad. "Convolutional neural network with word embedding based approach for resume classification." 2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR). IEEE, 2018.

[9] Nasser, Shabna, C. Sreejith, and M. Irshad. "Convolutional neural network with word embedding based approach for resume classification." 2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR). IEEE, 2018.

[10] Pendhari, Heenakauskar, et al. "Resume Screening using Machine Learning." 2023 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI). IEEE, 2023.

[11] Bhor, Shubham, et al. "Resume parser using natural language processing techniques." *Int J Res Eng Sci (IJRES)* 9.6 (2021): 01-06.

[12] Karaa, Wahiba Ben Abdessalem, and Nidhal Gribâa. "Information retrieval with porter stemmer: a new version for English." Advances in Computational Science, Engineering and Information Technology: Proceedings of the Third International Conference on Computational Science, Engineering and Information Technology (CCSEIT-2013), KTO Karatay University, June 7-9, 2013, Konya, Turkey-Volume 1. Springer International Publishing, 2013.

[13] Selva Birunda, S., and R. Kanniga Devi. "A review on word embedding techniques for text classification." Innovative Data Communication Technologies and Application: Proceedings of ICIDCA 2020 (2021): 267-281.

[14] Hayashi, Tomoki, et al. "Pre-Trained Text Embeddings for Enhanced Text-to-Speech Synthesis." INTERSPEECH. 2019.

[15] Yong, Zhou, Lishi Youwen, and Xia Shixiong. "An improved KNN text classification algorithm based on clustering." Journal of computers 4.3 (2009): 230-237.

[16] Xu, Baoxun, et al. "An improved random forest classifier for text categorization." J. Comput. 7.12 (2012): 2913-2920.

[17] Rad, Somaye Esmaeili, and Amir Rajabi Behjat. "Document classification base on ensemble classifiers support vector machine, multi-layer perceptron and k-nearest neighbors." J. Biochem. Tech 2 (2019): 174-182.

[18] Qi, Zhang. "The text classification of theft crime based on TF-IDF and XGBoost model." 2020 IEEE International conference on artificial intelligence and computer applications (ICAICA). IEEE, 2020.

[19] Hossin, Mohammad, and Md Nasir Sulaiman. "A review on evaluation metrics for data classification evaluations." International journal of data mining & knowledge management process 5.2 (2015): 1.

[20] Gaye, Babacar, and Aziguli Wulamu. "Sentiment analysis of text classification algorithms using confusion matrix." Cyberspace Data and Intelligence, and Cyber-Living, Syndrome, and Health: International 2019 Cyberspace Congress, CyberDI and CyberLife, Beijing, China, December 16–18, 2019, Proceedings, Part I 3. Springer Singapore, 2019.

[21] Uysal, Alper Kursat, and Serkan Gunal. "The impact of preprocessing on text classification." Information processing & management 50.1 (2014): 104-112.