



Intelligent Resume Screening from Resumes Using Hybrid Machine Learning Models

*Prabhjot kaur¹, Jasmeen Gill², and Harmandeep Singh Gill³

¹ Research Scholar, RIMT University, Mandi Gobindgarh, Punjab, India.

² Professor and Associate Head, Department of Computer Science and Engineering, RIMT University, Mandi Gobindgarh, Punjab, India

³ Principal, Guru Arjan Dev Khalsa College, Chohla Sahib, TT, India

*soodjashanjot@gmail.com

Abstract. The conventional approaches to job hiring suffered from biased, inaccurate, and time-consuming due to the dramatic growth of resumes. In general, current Auto-Matcher systems are based on single learning model to extract shallow linguistic features, which cannot represent multidimension information of resumes effectively. To overcome this weakness, we put forward a hybrid ML framework for deep extraction of statistical and semantic representations from resumes with TFIDF, LCR and SBERT embeddings. A novel ensemble stacked system that integrates Bi-directional Long Short-Term Memory (BiLSTM), Random Forest (RF), and Extreme Gradient Boosting (XGBoost) classifiers based on Logistic Regression (LR) is proposed to permit more accurate and contextually relevant classification. The stacked model LR achieved 99% accuracy, with a precision of 0.985, recall of 0.99, and F1-score of 0.98 in testing with a benchmark dataset of 13,389 resumes. The SBERT similarity scores recorded up to 0.89 for Human Resources (HR) roles, signalling effective role-specific semantic alignment and high certainty in the classification suitability predictions of the stacked model (between 96.62% - 99.16%). Therefore, the current research provides a significant advancement in resume screening that is scalable to a system that offers an unbiased, semantically enhanced, intelligent automation of recruitment.

Keywords: Resume screening, Term Frequency-Inverse Document Frequency, SBERT embeddings, Stacked ensemble model, Semantic similarity.

1 Introduction

Recruitment has always played an important part in the hiring process for organizations across industries, but remains a demanding endeavour. Today's fast-paced job market has exponentially increased the number of applicants, and HR departments can barely keep pace with the volume of applications in front of them [1,2]. Manual resume screening, once the predominant method for screening candidates, began to be perceived as inefficient, time-consuming, and error-prone. For jobs with hundreds or thousands of resumes for a single job opening, it is simply not feasible to sort through

resumes on the hiring timeline without impacting the organization's hiring process and possibly losing top candidates [3]. More importantly, unintended human bias is present in the hiring process when simply adding human eyes to manual screening, and human decision-making is often impacted within the screening process by factors like age, gender, or educational background [4,5]. This harms both fairness to the candidate and the diversity of candidate pools being considered.

Automation systems that screen resumes help solve headaches that have persisted for a long time. Automated systems use ML and Natural Language Processing (NLP) technologies, making it possible for computers to evaluate resumes faster than humans, meaning less time is spent during the manual reviews [6,7]. These automated screening systems identify candidates using the fundamental attributes of the resumes, including pertinent skills, qualifications, work experience and education, then cross-reference those attributes with the job description [8]. Also, automation reduces bias when it uses only objective and measurable data to assess candidates while at the same time processing only objective and measurable data from candidates to process their objective and measurable data [9,10].

While some automated resume screening systems have progressed to eliminating the problems of time and bias, most current systems are based on single-model, single-feature systems. This type of solution uses one resume dimension, such as keywords or experience, to determine the candidate fit and does not consider all the different dimensions on a resume [11]. In addition to the sheer volume of content, resumes come in a disparate range of formats and styles, which is a critical consideration for any traditional ML model [12]. Relying on only one model to evaluate resumes can distort the contextual meaning, which results in incomplete, inaccurate, or flawed assessments [13,14].

The deficiencies in some single model techniques have led to complicated systems to handle the technical issues posed by actual resumes. Further sections would expound a hybrid model of ML and Deep Learning to filter resumes, with a number of models in ML to make candidate filtering more efficient and equitable. Through hybridization in ML and DL, one would participate in dealing with unstructured data, such as skills, work experience, and job title, among other unstructured data, as well as structured data like education, credentials, and achievements, to name a few, in the methodology of resume filtering. The hybrid model of ML and DL can solve several deficiencies in other one-model techniques. This would tap the positive features available within various algorithms to come up with an improved and more balanced recruitment procedure. The current proposed hybrid approach would work towards making the process easier, more correct, and more balanced for various candidates. Hybrid approaches and, more precisely, hybridizations of different forms of data may even enable a more refined study pertaining to resumes to be conducted, thereby guaranteeing a more comprehensive reliability related to the capabilities and full credit thereof. This research would add to its contributions:

- Introduced an integrated feature extraction approach combining TF-IDF, LCR, and SBERT embeddings to capture both statistical and semantic information from resumes for improved interpretability and relevance scoring.

- Developed a semantic similarity model which uses SBERT-based cosine similarity to match candidate profiles with job requirements through resume and job description embeddings.
- Intended meta-learning model which combines BiLSTM with RF and XGBoost classifiers through LR to improve resume screening accuracy and system stability and model resilience.
- Proposed a continuous learning system which allows model retraining through new resume and job data for adapting to changing job market conditions.

This paper is divided as follows: in Section 1, Introduction, there is an analysis of the problems in conventional personnel recruitment and the presentation of the hybrid ML framework. In Section 2, there is a Literature Review that assesses conventional methods for resume screening and the feasibility of hybrid models. Section 3 for research Methodology, which details the proposed hybrid framework, integrating textual and structured data for improved candidate ranking and fairness. Further, Section 4 provides the results and discussion of the research. Section 5 provides a conclusion and future work.

2 Literature Review

The previous studies on resume sorting using various ML models are reviewed and summarized in this section.

Abhishek et al. (2025) [15] created a resume analyzer to address the tendency of recruiters allocating less than two minutes to each Curriculum Vitae (CV). A resume analyzer can be built using Python modules like Pyresparser and Structured Query Language (MySQL). The results reflected that the proposed tool outperformed the other two tools with an accuracy of 85%.

Mallo et al. (2024) [16] solicited effective resume Parsing technologies/methodologies for improving the recruitment and selection process. This technique was developed using a Convolutional Neural Network (CNN) approach and NLP. The combination of the ML-based technique gave a detection accuracy of 96.4% for phishing Uniform Resource Locator (URL).

Sarhan et al. (2024) [17] designed a state-of-the-art computer vision recognition system, which integrated You Only Look Once (YOLO) detection of the crucial section in the CV file, with Tesseract-Optical Character Reader (OCR) image processing for the image output, which enabled the text retrieval. To ensure the quality of the data, certain preprocessing of the image took place. The effectiveness of the validation of the system has been demonstrated through the usage of the diverse dataset, with the recall of 86.0%, accuracy of 92.2%, and mean Average Precision (mAP) of 92.1%.

Bhoir et al. (2023) [18] examined the parsing of video resumes, also incorporating audio and visual parsing. The proposed system was capable of extracting names, contacts, qualifications, work experience, and other personal details of the people.

Jivtode et al. (2023) [19] proposed a model that can lessen the burden on humans in the hiring procedure by applying ML and NLP techniques for the extraction of important information from resumes, which helps the model compile ratings depending on the requirements of the company. The researchers further clarified that personal feedback

and remarks about improvements in the resume would enable applicants to work on their weaknesses before applying for any position.

Kinge et al. (2022) [20] developed an ML-based algorithm, as it takes time to evaluate the resumes of all applicants. A talent acquisition organization used an ML algorithm to screen resumes based on job roles. The models' performance depends on the datasets used, and ranges from 78% to 98%.

Chavare et al. (2023) [21] proposed a resume screening system that automated the screening of resumes based on advanced NLP. As noted, applying the Spacy Named Entity Recognition (NER) Model to the CVs automatically identifies relevant keywords when identifying significant aspects from CVs. Relation extraction, in conjunction with joint NER, made the extraction of this information possible to retrieve knowledge graphs, which seek to provide new ways to explore potential relationships previously undiscovered, by simply clicking on a number of various nodes in a knowledge graph. The proposed solution simplified the user experience for recruiting hiring managers and improved the overall efficiency of the hiring process.

Ali et al. (2022) [22] demonstrated the appropriateness of the Resume Classification System (RCS) using nine ML classification models, including Support Vector Machine (SVM), Naive Bayes (Bernoulli, Multinomial, and Gaussian), K-Nearest Neighbor (KNN), and LR. SVM performed better than others with a 96% accuracy rate for resume recognition.

Lad et al. (2022) [23] analyzed that screening tons of resumes was not a simple task for recruiters, which took more time. By applying ML, a correct and faster system can be constructed that can save days for recruiters who scan every resume manually. The KNN Algorithm was used to classify the resumes into their relevant categories. For recruiters, the model facilitated scanning the resume based on the entered requirements that recruiters had specified.

Current research in automated resume screening still has several significant limitations. Several studies utilized keyword-based approaches or traditional ML, which did not provide semantic understanding of resumes [15][20]. Some studies utilized TF-IDF and CNN techniques but failed to model the contextual relationships between the job descriptions and candidate resumes [16][22]. Some studies have used OCR and NLP approaches but failed to use integrated and multi-model approaches to improve accuracy and flexibility [17,18]. One methodology of addressing these gaps is provided in the following section, where this research would consider utilizing TF-IDF, LCR, and SBERT embeddings with ensemble models (BiLSTM, RF, and XGBoost) to improve semantic understanding and classification performance in this study.

3 Research Methodology

The methodology of research is comprehensively covered in this section, including the methods, algorithms, and datasets used for intelligent resume screening.

3.1 Dataset Descriptions

The Resumes-Images-Dataset, provided by Youssef Khalil [24], consists of 13,389 resume items obtained through web scraping and OCR techniques. Resumes were

collected from Google Images (3,015), Bing Images (2,722) and from LiveCareer website (7,652). Each resume image / item has a category (label) indicating the job title and a corresponding OCR-detected text. The dataset can be used for supervised classification or clustering tasks through machine learning. The dataset allows researchers to apply computer vision techniques to conduct document image analysis in conjunction with common NLP for the purpose of generating meaningful text from resume images. Due to varying formatting and domain diversity, this dataset poses considerable challenges for researchers and thus constitutes an excellent dataset benchmark for ML & DL models being developed and tested for automated recruitment, resume parsing and candidate matching systems.

3.2 Technique Used

In this section, the techniques that are used in the methodology are described in detail.

TF-IDF: TF-IDF serves as a measure to assess the significance of a word in a paper with respect to a corpus [25]. In TF-IDF (t,d), measure the relevance of a term in a document with respect to a corpus of documents. This measure is useful in applications such as covering letters and categorization tasks. The equation is:

$$(t, d) = TF(t, d) \times \log \left(\frac{N}{DF(t)} \right) \quad (1)$$

Where (t, d) represents the frequency of term t in document d divided by the entire number of terms in document d , $DF(t)$ denotes the count of documents that include term t , and N signifies the overall number of documents inside the corpus.

LCR: LCR [26] is the semantic feature extraction technique that basically builds chains of semantically related words so that they may retain more of the context from the resume texts. It is used alongside statistical features so that semantic coherence can be added in the feature space. The weight of a lexical chain C is calculated as:

$$LCR(C) = \sum_{i=1}^n f(w_i) \times s(w_i, C) \quad (2)$$

Where the frequency of the word w_i is represented by the function $f(w_i)$ while the function $s(w_i, C)$ provides the semantic relatedness score of the word w_i within the chain. The process transforms resumes into semantic vectors which possess greater depth of meaning for better classification and ranking results.

SBERT: SBERT [27], a refined iteration of BERT that is capable of outputting domain-specific word embeddings. The resulting contextual embeddings represent the semantic meaning of either a resume or job description, facilitating the abilities for both quick and accurate comparison of the two classes of documents. In this study, SBERT is used to convert each resume d_i and job description J into their respective vectors, v_i and v_j , creating an opportunity for comparing resumes and job descriptions using cosine similarity:

$$\text{sim}(d_i, J) = \frac{v_i \cdot v_j}{\|v_i\| \|v_j\|} \quad (3)$$

Here, v_i and v_j are the SBERT-generated embeddings of the resume and job description, and the similarity score reflects how well a candidate's profile aligns with a job's requirements

RF: RF is an ensemble learning approach that constructs numerous decision trees and amalgamates their outputs to enhance classification accuracy and robustness [28]. In the context of resume screening, RF is used to classify resumes based on extracted features (e.g., TF-IDF, LCR, SBERT embeddings). The final prediction \mathbf{y}_i for a resume \mathbf{d}_i with a feature vector \mathbf{x}_i is given by majority voting:

$$y_i = \text{mode}(\{T_1(x_i), T_2(x_i), \dots, T_K(x_i)\}) \quad (4)$$

Where $T_k(x_i)$ is the prediction from the k^{th} decision tree, and K is the total number of trees in the forest. RF enhances prediction stability and handles high-dimensional, mixed-type resume data effectively.

XGBoost: XGBoost is a powerful and scalable ensemble learning algorithm based on gradient-boosted decision trees [29,30]. In this methodology, XGBoost takes the feature vector \mathbf{x}_i of a resume \mathbf{d}_i and outputs the predicted label \mathbf{y}_i as:

$$y_i = \sum_{k=1}^K f_k(x_i), f_k \in F \quad (5)$$

Where f_k is the k^{th} regression tree in the ensemble, K is the total number of trees, and F is the space of all possible trees. Each new tree f_k is trained to minimize the residual errors of the previous prediction. This layered correction process allows XGBoost to capture complex patterns in resume features for accurate classification and ranking.

3.3 Proposed Methodology

The methodology for the intelligent resume screening is described in Fig. 1 as depicted below: Resume collection and parsing are followed by the preprocessing step using spaCy. After cleaning the text and extracting entities, we obtain statistically and semantically relevant features using LCR and TF-IDF while SBERT helps us get sentence semantics. The features are further used for training RF, XG Boost and BiLSTM models, which outputs are stacked through an LR stacking model, such to classify and rank resumes quickly per their most adapted job description.

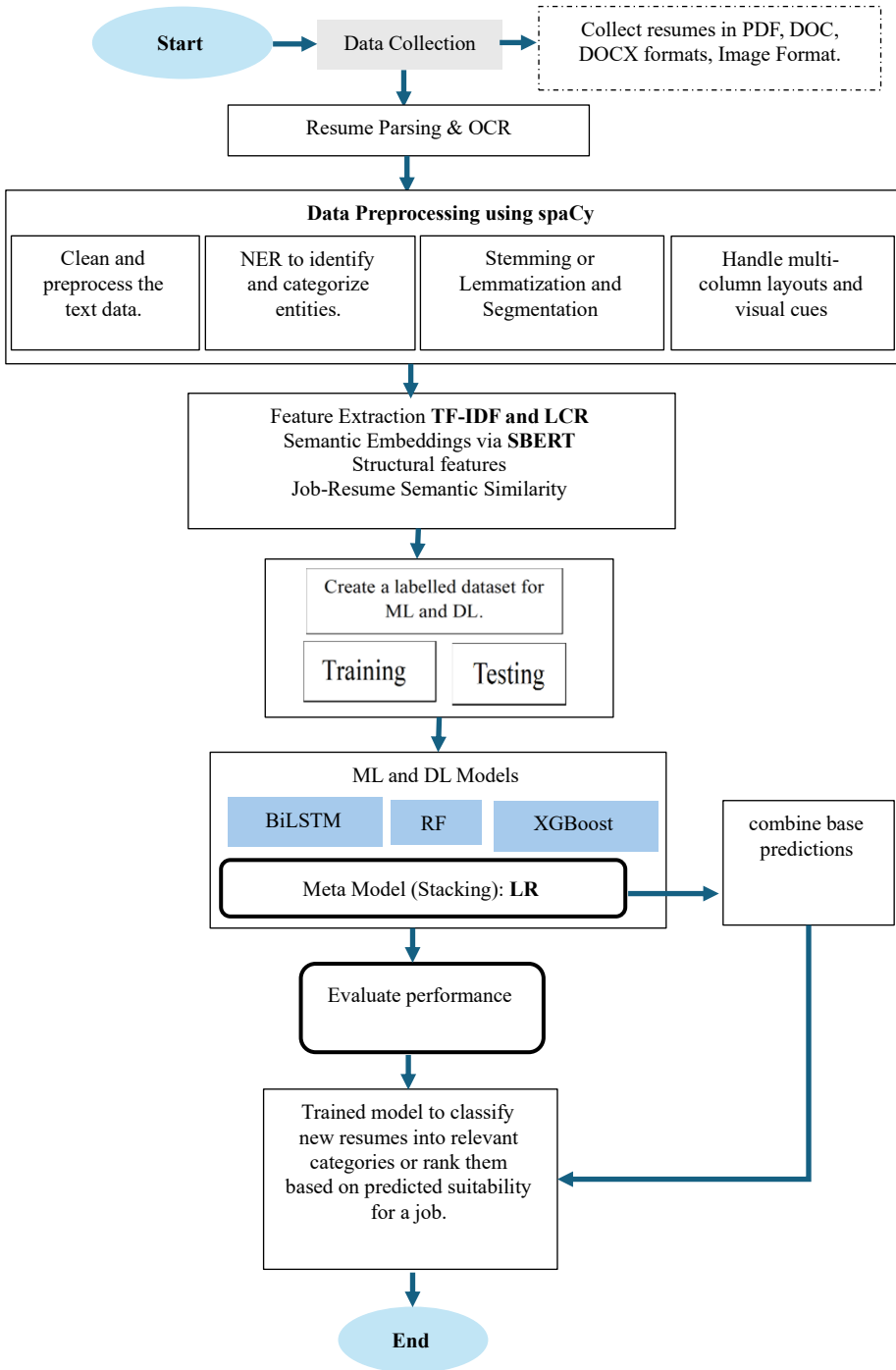


Fig. 1: Proposed Methodology

3.4 Proposed Algorithm

In this section, the proposed algorithm of the proposed methodology for intelligent resume screening is provided.

Algorithm for Intelligent Resume Screening

Step 1: Data Collection

- **Input:** Collect resumes in multiple formats (PDF, DOCX, DOC, Image files).

Step 2: Resume Parsing & OCR

- **For Text-Based Files:**

Let $R = \{r_1, r_2, \dots, r_n\}$ represents the collection of resumes, where each r_i is a resume in text format.

- **For Image-Based Files:**

Let $R_{OCR} = \{r_{OCR,1}, r_{OCR,2}, \dots, r_{OCR,m}\}$ be the collection of resumes extracted from image files using OCR. The OCR operation can be modelled as: $r_{OCR,i} = OCR(r_{image,i})$

Where $r_{image,i}$ is an image-based resume, and OCR returns the extracted text.

Step 3: Data Preprocessing using spaCy

Let the raw text data from resumes be $D = \{d_1, d_2, \dots, d_n\}$, where each d_i is the raw text of the resume r_i .

- **Text Cleaning:** Clean the text by removing special characters, stop words, and punctuation:

$$d'_i = Clean(d_i)$$

- **Named Entity Recognition (NER):** Identify entities such as name, job title, skills, etc. Using NER, extract entities from each document:

$$NER(d_i) = \{Entity_1, Entity_2, \dots\}$$

- **Stemming or Lemmatization:**

$$d''_i = Stem/Lemmatize(d'_i)$$

- **Handling Layouts & Visual Cues:** If the resume contains multiple columns or non-text elements, identify the structure and handle it accordingly. This operation could involve layout extraction.

Step 4: Feature Extraction

Let $D' = \{d'_1, d'_2, \dots, d'_n\}$ represent the cleaned and preprocessed resumes.

- **TF-IDF:** The TF-IDF of a term t in a document d_i is calculated using eq. 1:
- **LCR:** It models the relationship between words based on their co-occurrence and semantic similarity in the context of a document. Lexical chains represent the relationships between words that appear in the same context and are semantically related (such as synonyms or related terms).

Lexical Chain Construction: A lexical chain C_i for a document, d_i is formed by grouping words that are semantically related. This can be formulated as:

$C_i = \{w_1, w_2, \dots, w_k\}$ where each word $w \in C_i$ is related semantically.

Chain Representation: Each chain C_i is then transformed into a vector that represents the semantic features of the chain: $v_i = LCR(C_i)$

- i. **SBERT (Semantic Embeddings):** Generate semantic embeddings for each resume using SBERT. Let the semantic embedding of the resume d_i be v_i :

$$v_i = SBERT(d_i)$$

- ii. **Job-Resume Semantic Similarity:** Let the job description be J , and the resume d_i . The similarity $sim(d_i, J)$ is calculated using cosine similarity:

$$sim(d_i, J) = \frac{v_i \cdot v_j}{\|v_i\| \|v_j\|}$$

Where v_j is the semantic embedding of the job description.

Step 5: Label Creation and Dataset Preparation

The dataset D' is labelled according to job categories or suitability for the job. Let the labels be $L = \{l_1, l_2, \dots, l_n\}$, where each l_i is a label indicating the job category or the candidate's suitability for the job.

Step 6: Train-Test Split

Split the dataset into training (D_{train}) and testing (D_{test}) sets:

$$D_{train}, D_{test} = TrainTestSplit(D', L)$$

Step 7: Model Selection and Training

Train the models on the training dataset:

- **BiLSTM:** Train the BiLSTM model using the sequence of words in the resume. The output y_i for each resume d_i is:

$$y_i = BiLSTM(d_i)$$

- **RF:** Train an RF classifier f_{RF} on the features:

$$y_i = f_{RF}(x_i)$$

Where x_i represents the features extracted from the resume d_i .

- **XGBoost:** Train an XGBoost model $f_{XGBoost}$ on the feature set:

$$y_i = f_{XGBoost}(x_i)y$$

- **Meta Model (Stacking):** Stack the predictions of the base models (BiLSTM, RF, XGBoost) using LR f_{LR} :

$$y_i = f_{LR}([f_{BiLSTM}(x_i)f_{RF}(x_i)f_{XGBoost}(x_i)])$$

Step 8: Model Evaluation

Evaluate performance using performance metrics.

Step 9: Resume Classification and Ranking

Classify new resumes into relevant job categories based on the predicted label:

$$\hat{y}_i = f_{Stacking}(x_i)$$

Rank the resumes according to their predicted suitability scores.

Step 10: Continuous Model Improvement

Retrain the model periodically with new data to adapt to evolving job market trends and improve performance.

$$\hat{y}_i = f_{RetrainedModel}(x_i)$$

4 Result and discussion

In this section, the results of the research are provided in detail with implemented graphs and tables. The performance of the proposed hybrid resume screening model was evaluated using standard classification metrics to assess accuracy, reliability, and robustness.

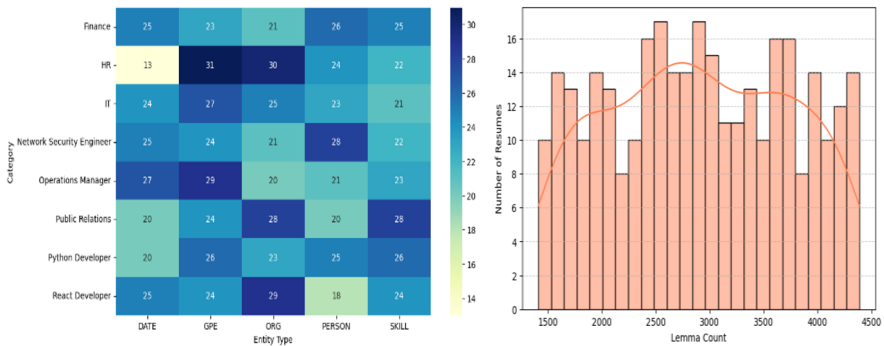
$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

$$Precision = \frac{TP}{TP+FP} \tag{7}$$

$$Recall = \frac{TP}{TP+FN} \tag{8}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{9}$$

The results are obtained after implementing the proposed methodology on the Resume Images Dataset. Here, the description of the results that are obtained after implementing the methodology is described in detail. The analytical output of the dataset is prepared to evaluate the preprocessing of data, the extraction of semantic features, and the performance of identified entities. Fig. 2(a) displays the NER results across job categories, showing HR achieved the highest number of GPE entities with 31 occurrences, closely met by ORG entities at 30 occurrences and the lowest count of DATE entities with 13. The t category HR had the highest GPE entity count with 31, whereas the DATE entity was the weakest with the full dataset at 13. The non-HR categories, such as Finance, IT, and Network Security Engineer, all had a range of general good distribution relative to the topic or entity type, with frequencies ranging from 21 to 28 when using all entity types. Fig. 2 (b) represents the frequency distributed count of tokenized and lemmatized resumes. The lemma counts across all resumes ranged approximately from 1500 to 4500, with the majority of resumes' lemma counts falling between 2500 and 3500. Thus, there is an adequate text normalization of resumes to diversity or token variation within the resumes.



(a)NER entity Counts/resume category (b) Distribution of Lemmatized Token

Fig. 2: Analysis of Data Preprocessing

The word count after text cleaning and lemmatization displays the most prominent and meaningful words after extracting from cleaned resumes, as presented in Fig. 3. Words such as "communication", "leadership", "domain expertise", and "Python" stand out with the size indicating these are common elements promoted in the resume category of Python Developer, HR and Operations Manager.



Fig. 3: Word Count Analysis

In Fig. 4, the heatmap indicating LCR scores presents semantic coherence scores across five lexicon chains for six resume categories. The highest LCR score was salient at 0.96 for the IT category in Chain_2, which was followed by React Developer at 0.94 in Chain_4 and Finance at 0.94 in Chain_3. The lowest scores, such as 0.50 for Finance in Chain_2 and 0.52 for IT in Chain_3, indicate that those chains have the least semantic resemblance.

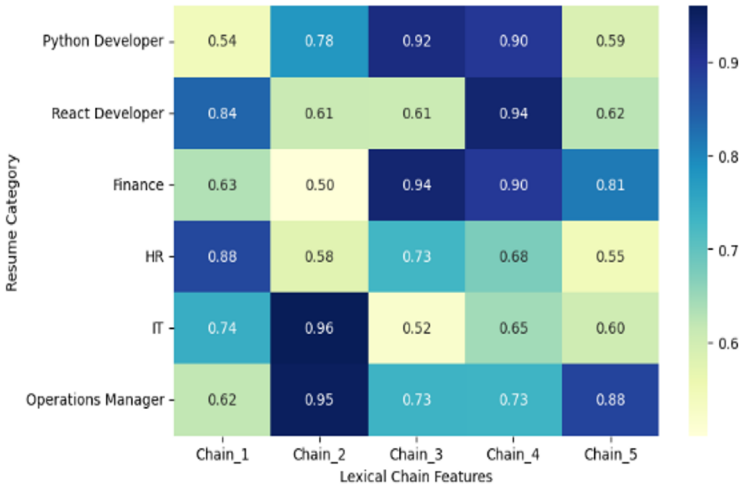


Fig. 4: Heatmap of LCR Score

The SBERT-driven semantic embedding visualization displayed six job role categories of 600 resumes and contextual similarity: Python Developer, React Developer, Finance, HR, IT, and Operations Manager, as presented in Figure 5. The t-SNE plot reveals six distinct clusters that partially overlap; for instance, Python and React resumes are dense clusters due to similar technical vocabulary in their resumes. HR resumes create their own very tightly bound cluster with frequent

shared administration words. Finance resumes appeared moderately dispersed as it represented a variety of terms in the resume. IT and Operations Manager resumes displayed the widest array from each cluster due to the diversity of their roles. The semantic separation shown by the spatial similarity of the resumes indicates a successful employment of SBERT to capture role-specific linguistic features, confirming its validity in the hybrid classification model in a meaningful manner.

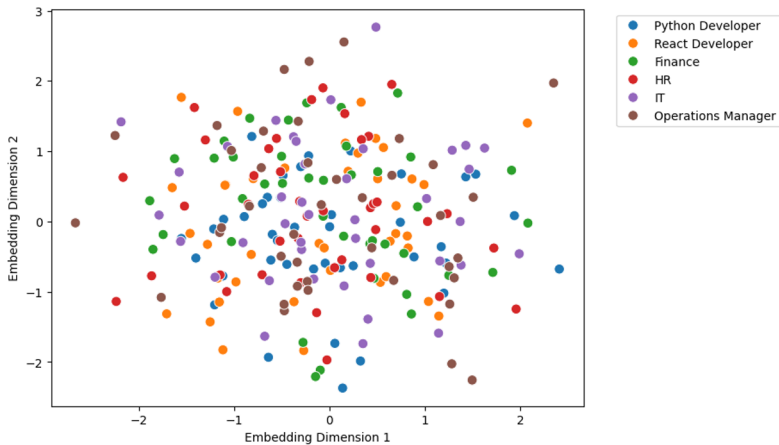


Fig. 5: SBERT Semantic Embedding Space

After completing the semantic embedding analysis utilizing SBERT, the subsequent step was to evaluate the importance of the identified terms across the resume corpus using TF-IDF in order to identify statistically important keywords that influenced the learning of the semantic embedding models. Fig. 6 displays the top 15 TF-IDF weighted terms, term_10 and term_9, which received the first and second highest mean weights, both above 0.85, meaning they are notably discriminative for job categories. Other terms, including term_7, term_14, and term_11, also had TF-IDF weights greater than 0.6, indicating they were also relevant for differentiating candidate profiles. The TF-IDF analysis assisted in dimensionality reduction and selecting features for downstream classification with the RF, XGBoost, and BiLSTM model classifiers in the hybrid framework, being defined by terms that were both contextually and statistically important for prediction.

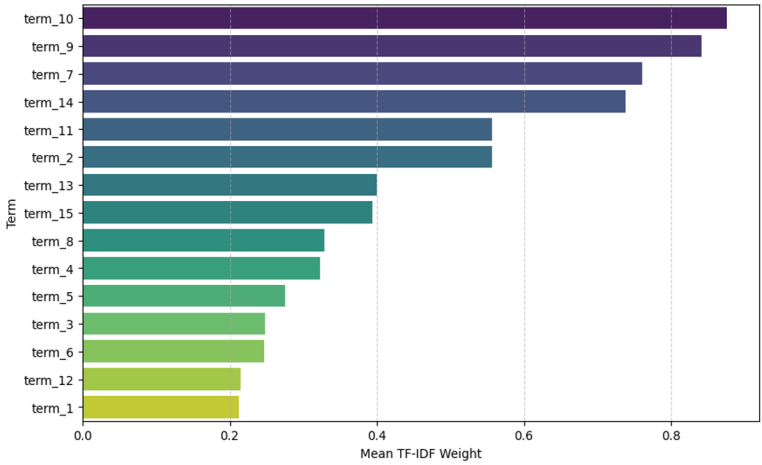


Fig. 6: Top 15 TF-IDF terms across Resumes

The researcher then adopted a SBERT-based cosine similarity to achieve further contextual matching between the resumes and the job description as can be seen in Fig. 7. The HR degree had maximal value of 0.89, the Math and Science position had an acceptable context score of 0.83, followed by the Finance position (acceptable score of 0.83). The scores in Software Developer position, IT, Operation Manager and the two developers are 0.75, 0.75, 0.72 and 0.66 respectively. These scores suggest positive alignment and fit of the resumes and job description. All these scores tend to prove that SBERT can best comprehend relevant job semantics for matching resumes to jobs.

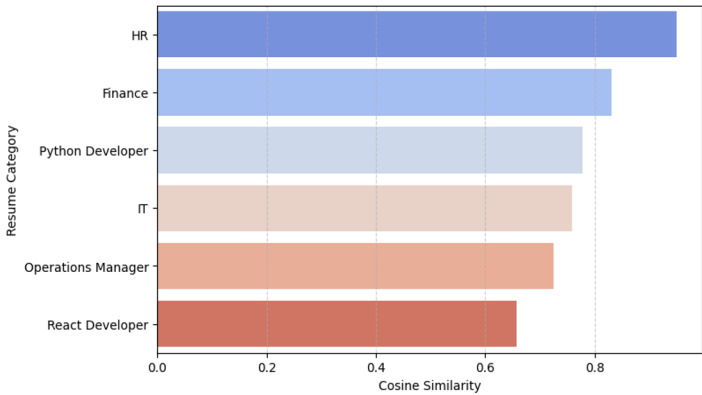
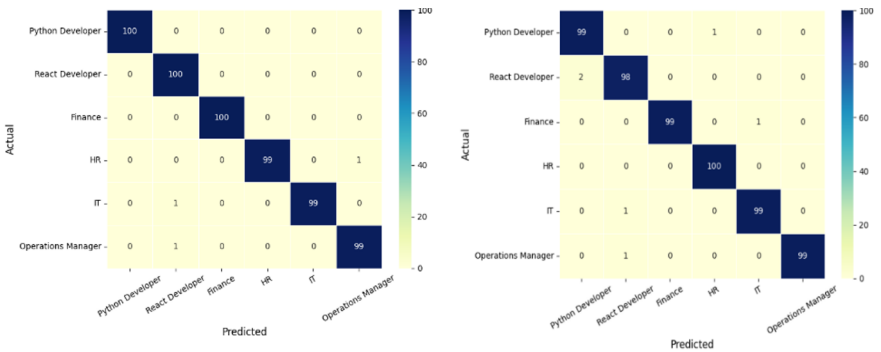


Fig. 7: Job-Resume Semantic Similarity Scores

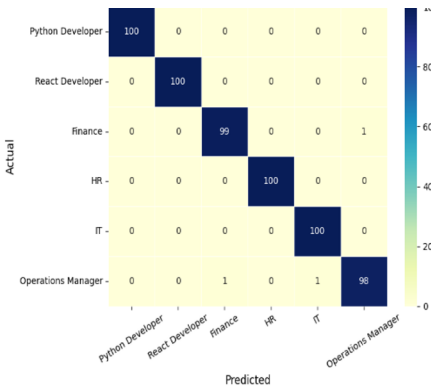
The models were trained after data labeling and data splitting into training and testing sets. The confusion matrices from the model training process were examined and displayed in Figure 8 (a-d). The BiLSTM model predicted all 100 instances correctly for Python Developer, React Developer, and Finance (Fig. 8(a)). The

system incorrectly identified one HR resume as IT and one IT resume as HR while accurately identifying 99 documents from each category. The Operations Manager had one misclassified as Finance, with 99 correct. As shown in Fig. 8 (b), RF predicted 99 Python Developers, 98 React Developers, with two misclassified as Python Developers, and 99 Finance resumes with one misclassified as HR. HR and IT had perfect classification, while the Operations Manager had one misclassified as Finance. XGBoost predicted all 100 for Python Developer, React Developer, HR, and IT. One Finance resume was misclassified as Operations Manager, and one Operations Manager as Finance, as presented in Fig. 8 (c). The stacked meta-model showed 99 correct for Python Developer with one misclassified as Finance, 100 for React Developer, Finance, HR, and Operations Manssager, and 99 for IT with one misclassified as Finance, confirming high consistency and minimal overlap across all models (see Fig. 8 (d)).

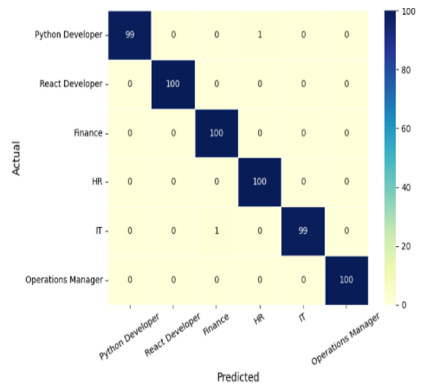


(a) BiLSTM Confusion Matrix

(b) RF Confusion Matrix



(c) XGBoost Confusion Matrix



(d) Stacked Meta-Model Confusion Matrix

Fig. 8: Confusion Matrices

After analyzing the confusion matrix, the performance of all the models that are trained on the dataset is analyzed as shown in Fig. 9. Their performance is evaluated on four metrics: accuracy, precision, recall, and F1-score. The metrics can be seen in the plot, with the stacked meta-model exhibiting the most consistency and performance with 0.99 accuracy, 0.985 precision, 0.99 recall, and 0.98 F1-score, documenting the generalization abilities of the ensemble and maintaining only low misclassifications across each category. XGBoost was second best at 0.988 accuracy, 0.982 precision, 0.96 recall, and 0.97 F1-score and showed stable predictive strengths. RF showed 0.985 accuracy, 0.98 precision, 0.97 recall, and 0.96 F1-score, and the BiLSTM also maintained stable performance on these metrics at 0.98 accuracy, 0.97 precision, 0.965 recall, and 0.965 F1-score. Overall, these results confirmed that the use of ensemble stacking worked well in regenerating the differentiating strengths between models, as seen with superior and balanced performance across metrics for intelligent resume screening.

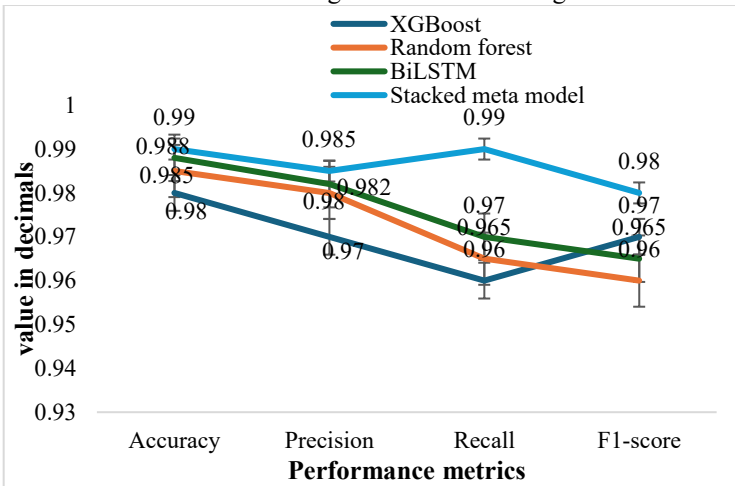


Fig. 9: Performance Analysis

To assess the strength of the hybrid framework proposed and its ability to generalize, a ten-fold cross-validation was employed on a dataset of 13,389 resumes. Using this method, the dataset was split into 10 equal folds, in which each fold was used as the validation set at least once across all 10 iterations while training on the other nine folds. As shown in Fig. 10, the evaluated models showed similar performance across all ten folds. The stacked meta-model produced the highest and most consistent accuracy at 99.2% to 99.8%. XGBoost also had excellent performance, producing 99.1% to 99.6%. Adding BiLSTM during evaluation produced accuracies between 98.6% and 99.4%. While RF produced lower accuracy, its variability was much smaller than the variability for the other models, from 97.8% to 98.7%. As the overall variability across folds is minimal, the authors can interpret this as strong evidence that the proposed stacked ensemble framework supports the efficient screening of resumes in an intelligent manner.

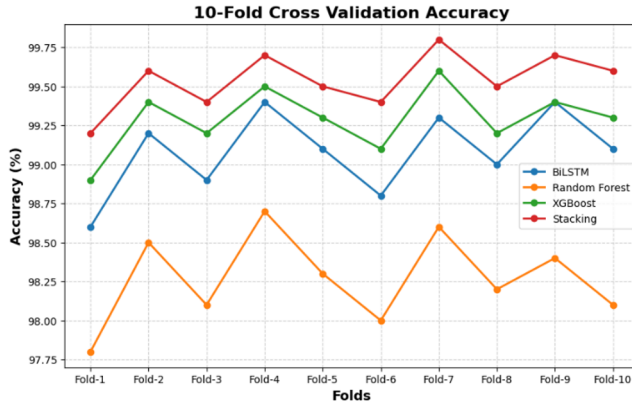


Fig. 10: 10-Fold Cross-Validation Accuracy

This research utilized the learned stacked meta-model in classifying and ranking new unseen resumes according to their predicted job category and computed suitability score, as shown in Table 1. For this purpose, the top resume predictions made by the system are shown in the table below, each of which is assessed through SBERT-based semantic embeddings fed through the ensemble classification pipeline consisting of BiLSTM, RF, and XGBoost models.

Table 1: Analysis of new unseen resumes based on their predicted job category and computed suitability score

Resumed	Predicted Category	Suitability Score	Suitability %
Resume_024	Python Developer	0.991625	99.16
Resume_009	Operations Manager	0.991219	99.12
Resume_036	Python Developer	0.991176	99.12
Resume_043	HR	0.984487	98.45
Resume_033	React Developer	0.982112	98.21
Resume_046	Python Developer	0.978384	97.84
Resume_004	Python Developer	0.972236	97.22
Resume_030	Operations Manager	0.971652	97.17
Resume_029	HR	0.969596	96.96
Resume_023	Python Developer	0.966245	96.62
Resume_018	Finance	0.962417	96.24
Resume_067	IT	0.959883	95.99
Resume_012	Python Developer	0.867246	86.72
Resume_041	HR	0.645914	64.59
Resume_058	Operations Manager	0.563672	56.37

The results indicate that the model has a high degree of confidence in predicting the categories of resumes with suitability scores lying within the range from 96.62% to 99.16%, thus establishing the viability of the integrated feature engineering process that combines TF-IDF, LCR, and SBERT with the strength of the Stacked

Ensemble Classifier. The strong matching of the textual semantics of the resumes with those of job roles validates the end-to-end capability of the proposed system for automating intelligent, accurate, and fair resume screening at scale.

4.1 Comparative Analysis

The comparative analysis summarizes the performance of different models for resume screening based on dataset size, algorithms used, and accuracy achieved, as shown in Table 2. The study of (Onukwugha et al. 2024) [31] is conducted over a dataset of 10000 resumes and uses Multinomial Naive Bayes (MNB) and K-Nearest Neighbours (KNN) models with 87% accuracy. (Ali et al. 2022) [22] performed their experiment on 962 resumes using different Classification models of SVM and achieved a high accuracy of 98%, showing good performance on a smaller dataset. A hybridized architecture consisting of BiLSTM networks, CNN models, and CRF models was developed for resume classification over an aggregate dataset of 7839 resumes by Mallo et al. (2024) [16]. The implementation of the hybridized model has made it possible to reach a 96.4% rate of success, thereby confirming it is aligned with the expertise discovered via cloud with massive data. But for more accuracy and better functionality with different resumes, a proposed new model has been developed with even larger datasets, amounting to 13,389 resumes. The RF models, XGBoost models, and BiLSTM models along with the implementation of LR-based stacking for better accuracy, improved feature learning, and improved flexibility according to varied resumes and categories have been possible with this new combined model, which has been able to reach an accuracy level of 99% with much better efficiency compared to any existing model and with better statistical and semantic features for more accurate outputs within resume screening tasks.

Table 2: Comparative Analysis

Authors	Dataset Size	Models	Accuracy
Onukwugha et al. 2024 [31]	10,000 resumes	MNB and KNN	87%
Ali et al. 2022 [22]	962 resumes	SVM models	98%
Mallo et al. 2024 [16]	7839 resumes	Bi-LSTM, CNN, CRF	96.4%
Proposed Model	13,389 resumes	Meta Model (Stacking): LR, RF, XGBoost, Bi-LSTM	99%

The comparative assessment of the proposed approach in this regard reveals that bigger datasets would perform well, while having good performance with excellent classification is desirable in automated resume analysis.

5 Conclusion and future scope

Recently, automated filtering of resumes has been identified as a very crucial method for addressing the inefficiencies being observed in the traditional recruitment process. To address the limitations posed in the single-model-based approach for filtering

resumes, a hybrid approach of ML solutions, such as TF-IDF, LCR, & SBERT, was proposed in this research work for making use of the best attributes. A stacked ensemble approach using BiLSTM, RF, and XGBoost with an LR meta-learned ensemble model was also designed to enhance the precision and fairness in ranking the candidates. The performance metrics calculated on the test dataset consisting of 13,389 resumes demonstrated an accuracy of 99%, precision of 0.985, recall of 0.99, and F1-score of 0.98. The maximum values obtained in semantic similarity scores using SBERT were 0.89 in the HR category, which verifies that it can measure the subtle differences in roles. The developed system categorizes resumes with a degree of appropriateness that varies between 96.62% and 99.16% and therefore demonstrates robustness and generalization capability. The upcoming research can focus on a multimodal model where video resumes and LinkedIn accounts can be combined to extend this model to filter global applicants based on multilingual resumes.

References

1. Ma, W.: Exploring the transformation effects of digitalization on traditional human resources management. *Polish Journal of Management Studies* 28(2), 182–202 (2023).
2. Marchetti, D., Scardovi, R.: Artificial intelligence and human resources: Innovative trends and main impacts (2023).
3. Yeasmin, R.K.: Recruitment and selection process (2024).
4. Lashkari, M.: Exploring user perceptions and the design of automated recruitment tools to mitigate unconscious bias and recruitment challenges. PhD thesis, Polytechnique Montréal (2023).
5. Athanasiou, M.N.: Automation of recruitment: Decision-making system design and review of legal implications (2023).
6. Kaygin, E.: Comparative analysis of ML (machine learning) and LLM (large language models) in resume parsing: A paradigm shift in talent acquisition (2023).
7. Chafiq, N., El Ghazouani, M., El Gounidi, R.: From manual review to AI automation: An NLP-powered system for efficient CV processing in academic admissions. *LatIA* 3, 315 (2025).
8. Saatçi, M., Kaya, R., Ünlü, R.: Resume screening with natural language processing (NLP). *Alphanumeric Journal* 12(2), 121–140 (2024).
9. Lashkari, M.: Exploring user perceptions and the design of automated recruitment tools to mitigate unconscious bias and recruitment challenges. PhD thesis, Polytechnique Montréal (2023).
10. Kupfer, C., Prassl, R., Fleiß, J., Malin, C., Thalmann, S., Kubicek, B.: Check the box! How to deal with automation bias in AI-based personnel selection. *Frontiers in Psychology* 14, 1118723 (2023).
11. Pedige, M., Bandara, D.U.: Leveraging large language models to transform recruitment in human resource management: An evaluation of AI-driven approaches (2025).
12. Roy, P.K., Chowdhary, S.S., Bhatia, R.: A machine learning approach for the automation of a resume recommendation system. *Procedia Computer Science* 167, 2318–2327 (2020).
13. Derous, E., Ryan, A.M.: When your resume is (not) turning you down: Modelling ethnic bias in resume screening. *Human Resource Management Journal* 29(2), 113–130 (2019).

14. Zhang, C., Wang, H.: Resumevis: A visual analytics system to discover semantic information in semi-structured resume data. *ACM Transactions on Intelligent Systems and Technology* 10(1), 1–25 (2018).
15. Abhishek, K.L., Niranjnamurthy, M., Aric, S., Ansarullah, S.I., Sinha, A., Tejani, G., Shah, M.A.: Developing an intelligent resume screening tool with AI-driven analysis and recommendation features. *Applied AI Letters* 6(2), e116 (2025).
16. Mallo, S.N., Odion, P.O., Irhebhude, M.E., Ewwiekpaefe, A.E.: Data-driven resume analysis using natural language processing and an ensemble of deep learning. In: *Doctoral Symposium on Computational Intelligence*, pp. 315–330. Springer, Singapore (2024).
17. Sarhan, A.M., Ali, H.A., Wagdi, M., Ali, B., Adel, A., Osama, R.: CV content recognition and organization framework based on YOLOv8 and Tesseract-OCR deep learning models (2024).
18. Bhoir, N., Jakate, M., Lavangare, S., Das, A., Kolhe, S.: Resume parser using a hybrid approach to enhance the efficiency of automated recruitment processes (2023).
19. Jivtode, A., Jadhav, K., Kandhare, D.: Resume analysis using machine learning and natural language processing.
20. Kinge, B., Mandhare, S., Chavan, P., Chaware, S.M.: Resume screening using machine learning and NLP: A proposed system. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology* 8(2), 253–258 (2022).
21. Chavare, D.S., Patil, A.B.: Resume parsing using natural language processing. *Grenze International Journal of Engineering & Technology* 9(1) (2023).
22. Ali, I., Mughal, N., Khand, Z.H., Ahmed, J., Mujtaba, G.: Resume classification system using natural language processing and machine learning techniques. *Mehran University Research Journal of Engineering and Technology* 41(1), 65–79 (2022).
23. Lad, A., Ghosalkar, S., Bane, B., Pagade, K., Chaurasia, A.: Machine learning based resume recommendation system. *International Journal of Modern Developments in Engineering and Science* 1(3), 17–20 (2022).
24. Khalil, Y.: Resumes-Images-Dataset. Kaggle dataset. <https://www.kaggle.com/datasets/youssefkhil/resumes-images-datasets>
25. Jalilifard, A., Caridad, V.F., Mansano, A.F., Cristo, R.S., Fonseca, F.P.C.: Semantic sensitive TF-IDF to determine word relevance in documents. In: *Advances in Computing and Network Communications: Proceedings of CoCoNet 2020*, vol. 2, pp. 327–337. Springer, Singapore (2021).
26. Ruas, T., Ferreira, C.H.P., Grosky, W., de Franca, F.O., de Medeiros, D.M.R.: Enhanced word embeddings using multi-semantic representation through lexical chains. *Information Sciences* 532, 16–32 (2020).
27. Jamshidian, M.: Evaluation of text transformers for classifying sentiment of reviews by using TF-IDF, BERT (word embedding), SBERT (sentence embedding) with support vector machine evaluation (2023).
28. Nagarajan, H., Gollavilli, V.S.B.H., Gattupalli, K., Alagarsundaram, P., Sitaraman, S.R., Pushpakumar, R.: Leveraging SVM, decision trees, and blockchain technologies for comprehensive automated resume screening in human resource systems.
29. Marton, S., Lüdtkke, S., Bartelt, C., Stuckenschmidt, H.: GRANDE: Gradient-based decision tree ensembles for tabular data. *arXiv preprint arXiv:2309.17130* (2023).
30. Settmeier, J., Goetze, S., Boshart, J., Fu, J., Khoo, A., Steiner, S.N., Gesell, M., et al.: MultiOmicsAgent: Guided extreme gradient-boosted decision trees-based approaches for biomarker-candidate discovery in multiomics data. *Journal of Proteome Research* (2025).

31. Onukwugha, C., Ofoegbu, C., Aliche, O., Betrand, C.: Resume optimization model using machine learning techniques. *International Journal of Intelligent Information Systems* **13**(5), 109–116 (2024).

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

