2024 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT) | 979-8-3315-4068-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICONSCEPT61884.2024.10627825

An ML-based Resume Screening and Ranking System

*Vishaline AR, Riya Kallankattil Pramodh Kumar, Sai Pramod VVNS, Vignesh KVK, Sudheesh P**
Department of Electronics and Communication Engineering,
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India
*Email: p_sudheesh@cb.amrita.edu**

***Abstract*— In today's job market, resumes have flooded businesses, and the recruitment process has become an imposing task for companies. The traditional ways of resume screening are time-consuming and biased. To address these challenges, this research paper proposes a solution to automate this process through machine learning techniques. Based on existing research in this field, the aim is to improve the efficiency and accuracy of selecting candidates that are best suited for each job role without the use of complex techniques and provide equally good results. Existing methods involve the use of NLP, ML, and AI techniques to perform the screening process, each having its own advantages and disadvantages. Different ML models like SVM, Naive Bayes, and XGBoost have been explored and compared to find the most suitable and accurate approach. To improve accessibility and to implement a user-friendly system, a web application has also been created. On comparing the performance of these models using various performance metrics such as accuracy, precision, and F1 score, XGBoost is found to be the most suitable model with 0.89 AUC-ROC value and is used in the web application. As a novel approach, the candidates' resumes are assigned scores based on the user's requirements and then ranked enabling the recruiter to shortlist suitable candidates for the job. This approach aims to simplify the resume screening process and revolutionize the hiring process.**

***Index Terms*—ML, SVM, Naive Bayes, XGBoost, web application, performance metrics.**

I. INTRODUCTION

With the increasing number of candidates graduating college every year, choosing the right candidates from the huge numbers could be a hectic task for the recruiters. Identifying the individuals with the right skill set for a job and automatically ranking them would aptly ease the process of shortlisting. This project has been designed with an overview of easing recruitment by automatically screening and ranking resumes, helping to weigh down the difficulties of manually shortlisting candidates. In this project the webpage serves as the entry point collecting information from the user, which is sent to the backend for further processing. The final stage comprises ranking the resumes classified by sorting them in terms of their rank. The integration of the technologies like NLP and ML has automized the resume screening and ranking process. It can also be noted that the model may face challenges while dealing with unstructured data from the resumes and to deal with this, a major condition of having the resumes be in a particular structure that compliments the working of the model and overall processing is expected to be maintained. However, by using ML continuous learning and increased accuracy over time can be some of the key factors to look at.

II. LITERATURE SURVEY

The current state of unemployment is proof that there is an increase in the number of job seekers for any job. This leads to a large number of resumes being submitted for every job profile and the shortlisting process results. This consequently lengthens the shortlisting process if performed manually. With the shift to software tools in the hiring and job-seeking field, various alternatives are required. The integration of NLP, ML, and AI techniques in the recruitment department represents a significant shift in traditional recruitment practices.

Employment-focused websites have recommendation engines that provide job recommendations based on the user's profile[1]. These engines employ ML/AI tools and techniques to perform these recommendations. Some of them also provide the option to create a resume using the user information. Resume rating websites also exist; these websites rate how good a candidate's resume is and provide insights to improve the quality of the resumes[2][3].

With many companies having their own ATS(Applicant Tracking System)[4][3] or resume screening software, it is important to find suitable equivalents that can be accessed by everyone since such software are often limited to the companies themselves and cannot be used by startups or other organizations.

Such systems have brought about radical changes to HR management practices. The efficiency of the recruitment process has increased due to such systems which are powered by ML or AI-based techniques. ML models such as SVM, Naïve Bayes' Classifier, XGBoost, KNN and Random Forest[5] are known to be effective models for performing text classification. Such models are used to decide whether the candidate is suitable for the job based on the content.

III. METHODOLOGY

1. Web Design

The design of the webpage, in this project, plays an important role in obtaining the input parameters that pave the way for the completion of the objectives for this project. Using the programming languages HTML, CSS, JavaScript, and the

979-8-3315-4068-5/24/$31.00 ©2024 IEEE

React Library a web page has been formulated that takes multiple resumes together as a zip folder, which is then stored in the database [1].

Subsequently, the recruiter gets the availability to navigate to the requirements page which consists of different categories that need to be filled; this then serves as the recruiter requirement that would help in the ranking of the resumes based on the level of match between these and the information present in each of the resumes. This functionality includes the possibility for the recruiter to input the priority of entry under certain categories by using the star rating method. To be precise, the overall categories included are Education, Technical Skills, Technical Interests, Internship, Projects, Certification, and Languages of which the priority marking would be enabled for all the categories except the Education field.

These fields are also occupied with the availability of a drop-down menu that could reduce the possible errors prone to occur due to spelling mistakes or wrong entries. It also works as a suggestion for the recruiter on what skills could be present in a candidate's resume. All these gathered data are then sent to the backend for further processing after which the final output consisting of the ranked resumes will be displayed in a sorted manner, making it easy for the recruiter to choose the suitable ones from the vast number of applicants resumes. This can then be downloaded in a sorted manner; thereby making it useful for future purposes. This web app also consists of a Help page that gives a guide on how the website needs to be used, and what the standard rules that each resume must follow are. The Contact Us page allows the user to contact the developers of this application for any related queries. The website was designed in terms of structure and styling by HTML and CSS, whereas JavaScript and React JS contribute to the creation of dynamic and interactive content that makes the user interface user-friendly. The use of react for designing the functionality of the webpage helps the developer to manage and maintain complex UI structure by dividing it into individual reusable components that form the building blocks of the whole UI and thereby using it for this purpose makes it more efficient to attain the objective. React JS also combines the speed and efficiency of JavaScript attempting to harness the full potential of a web application. It also plays a role in making the webpage responsive such that it would perfectly fit into any screen size and adjust the component placements accordingly within the webpage, thereby making the webpage render its functionality in full fledge under all settings[6].

2. MongoDB

It is MongoDB that manages the database. For ML-based resume screening, MongoDB's primary benefit over alternative database management models like Cassandra, MySQL, and Postgresql is its ease of handling unstructured input data[7][8]. Given that MongoDB is schema-less, it does not require frequent updates to the repository, in contrast to relational databases like MySQL and PostgreSQL[9][10]. It is also skilled at managing data in multiple formats, which is necessary to support a wide range of resume data. MongoDB is the clear choice for resume analysis in machine-learning systems because of its adaptability and capacity to manage a broad range of data structures[7].

The BSON (Binary JSON) format is used by MongoDB for data storage. The data is stored as schema-less documents that may contain complex structures. Privacy has been preserved by masking the data. Because the document is part of a collection, searching and retrieving it will be easy. In addition to MongoDB's document-based storage, indexes help the system perform quickly even when dealing with large volumes of resume data by accelerating data retrieval.

3. ML model

Machine Learning has found its application in various domains ranging from speech enhancement to object tracking systems. However, not all such applications require ML to be solved[11][12]. In the context of resume screening, ML is considered to be one of the most suitable ways to develop an efficient solution. The integration of ML, AI, and NLP techniques can revolutionalize the recruitment department, by making them more accurate and efficient. These techniques play a major role in designing systems for various steps of the recruitment systems.

All resumes attached to a job application go through numerous steps of screening and shortlisting to find a suitable candidate for the job. However, this might prove to be extremely time-consuming if performed manually. This might also result in suitable candidates being missed out and biased for certain candidates. It is necessary to analyze the candidate's skills while performing this screening process to provide equal opportunities to qualified and skilled candidates. This information can be obtained only through the candidates' resumes. Therefore, ML is found to be a suitable method to perform this screening process.

3.1. Data Collection

Over 500 resumes were collected and converted to a universal format for the dataset creation. Text extraction was performed, and the resulting data was labeled 'required' and 'not required'. This labeling was performed based on the presence of skills required for a software job. Data labeling has been performed based on the skills present in the candidate's resume. For a resume to be labeled 'required', it must contain a minimum of 3 CSE skills. This labeling enables the model to learn the underlying patterns and classify other such resumes for similar jobs. The labeling of the dataset has been designed to work only for software jobs and so will not provide results for any other domain.[13]

3.2. Data Preprocessing

Text extraction must be performed on the resume PDFs to extract the information present in it for further processing[14]. All PDFs consist of structured elements including text, images, font, and layout information. The retrieved byte streams are

decoded and mapped back to the corresponding encryption resulting in the encoded text as the output[15].

Stop words such as ‘and’, ‘the’, ‘was’, etc. appear in the text and are not necessary for the classification process and hence are removed. To perform this, the extracted text is tokenized and filtered using a stop-word dictionary created for the purpose. Non-alphabetical characters and symbols are also removed, and the resulting text is devoid of such unnecessary characters. The removal of these characters helps improve the model’s learning and is an essential part of data cleaning[16][17].

Each resume is made of multiple parts, such as ‘Profile’, ‘Technical Interests’, ‘Projects’, ‘Technical Skills’, ‘Internships’, ‘Certifications’, ‘Achievements & Honours’, ‘Languages’ and ‘Extra-Curricular Activities’. These divisions are present as uppercase fonts to partition the sections. The text under each section is extracted and stored for further efficient model training and score computation.

During the preprocessing stage of resume analysis, addressing abbreviations is pivotal for ensuring comprehensive data handling before the application of machine learning algorithms. Incorporating abbreviation replacements is a critical step in this phase. By including predefined mappings for abbreviations like "Cpp" for C++, "py" for Python, "DL" for Deep Learning, "ML" for Machine Learning, "IOT" for Internet of Things, and others, the preprocessing stage normalizes the text, ensuring that these terms are recognized uniformly across all resumes. This normalization allows subsequent machine learning algorithms to accurately interpret and assess the content, thereby preventing potential oversight or biases based on varied representations of the same concept or term.

An ML model is incapable of understanding strings as they are, so there is a necessity to convert them into numerical representations to enable the model to effectively operate on the data. This is done through feature extraction where each document’s text is represented by a vector that indicates the frequency of each word’s occurrence within the document. These feature vectors are represented using a sparse matrix format where each row corresponds to a document and each column corresponds to a word in the vocabulary. This representation enables the model to learn the underlying patterns and perform classification on new data.

3.3. Candidate Screening

The recruiter's education requirements in terms of CGPA, Class 12, and Class 10 grades are used to filter the candidates and retain those who fit the requirements. The UG and PG degree requirements are also considered. To perform this, the input string is iterated to extract this information and to check whether the requirements match. The candidates who qualify are retained and the others are removed from the process.

3.4. Model Training

Using the labeled dataset, multiple ML models have been trained to analyze the performance of each and choose the most suitable one for this application.

3.4.1. SVM

Support Vector Machines are supervised ML models that analyze data for classification and regression. Although SVM works well for linear classification, it can also be used to efficiently perform non-linear classification due to the availability of kernels. For this dataset, the RBF kernel has been selected. The Radial Basis Function (RBF) kernel can capture non-linear relationships making it suitable for the resume screening process [18][19]. The model finds a hyperplane that best separates the data into ‘required’ and ‘not required’ classes[20]. The linearity of the relationships in the data cannot be understood at ease, and hence this non-linear decision boundary allows SVM to handle the non-linear relationships in the data[21][22].

3.4.2 Naïve Bayes

Naïve Bayes is a probabilistic classifier based on Bayes’ theorem with strong independence assumptions between features[23]. Despite this assumption, it performs well in text classification tasks, is computationally efficient, and can perform well in relatively small datasets. The algorithm builds a probabilistic model for each class by estimating the probability of a particular set of features in the class label. This model estimates the conditional probability of each feature given each class and the prior probability of each class. [19].

3.4.3 XG Boost

XG Boost is a gradient-boosting algorithm that combines predictions of multiple decision trees to make predictions. The objective function used here is a loss function that must be optimized and a regularization term to control the model complexity[16][24]. The objective function plays a pivotal role in penalizing deviations from the actual values. This algorithm builds each tree consecutively, with each new one trained to correct the errors of the combined ensemble[25]. It performs gradient optimization to determine the best direction to move on.[26].

4. Ranking

Out of the resumes that have been positively classified, there remains a requirement to rank them to assign scores and assess each resume's potential. The priority stars assigned by the recruiter in the webpage are converted to weights in the backend for processing; where if a field is assigned with ‘n’ stars, it gets converted to a weight of ‘n’ points. Against the different inputs of a certain category, the resumes may contain information on the level of expertise in that field, which can be categorized as Beginner, Intermediate, Advanced/Developer. These levels of expertise are assigned with their weights where a beginner gets 1 point, intermediate gets 2 points, advanced/developer gets 3 points, and by default if nothing has been mentioned 1 point would be allocated. These are then multiplied by the priority-based points assigned for that field. The sum of all the products

obtained from every possible field within the categories will then compute the final score of a resume, which can be put together in the form of a mathematical expression as Equation1.

$$\text{Total Score} = \sum P(i)*W(i) \quad (1)$$

Here, P(i) will be the points obtained after converting the priority to weights in the backend, and W(i) will be the points that are assigned to an input based on the level of expertise. Ranking ensures fair assignment of scores and opportunities to those individuals with the required skillset.

IV. RESULTS

Web Page

The recruiter must upload the zip folder containing resumes to the upload page shown in *Fig. 1*, obtained on navigation from the home page.

Fig. 1 Zip folder upload page

The next step involves taking in the recruiter requirements to rank the resumes. The recruiter can choose the requirements for each category from a dropdown menu and assign its priority, using the star rating method as displayed in *Fig.2.* Upon receiving these inputs, the backend processes the score based on the match between the skillset present in the resumes and the recruiter's requirements. *Fig. 3* shows the final page that displays the resumes in a manner ranked from the top rank to least rank, based off the ranking method computed that ensures fair assignment of scores.

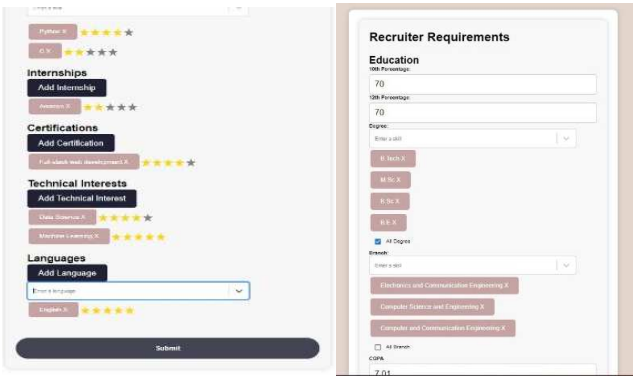


Fig. 2 Recruiter requirements page – requirements with priority

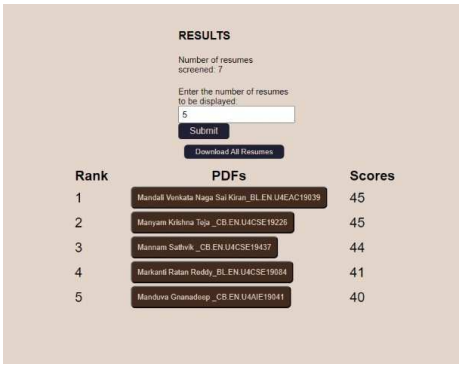


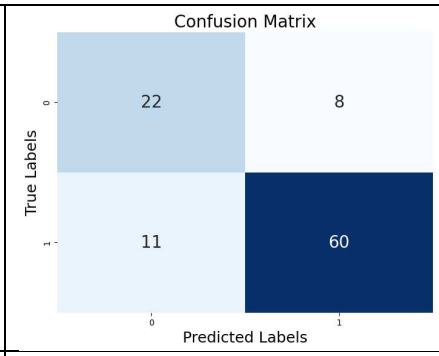
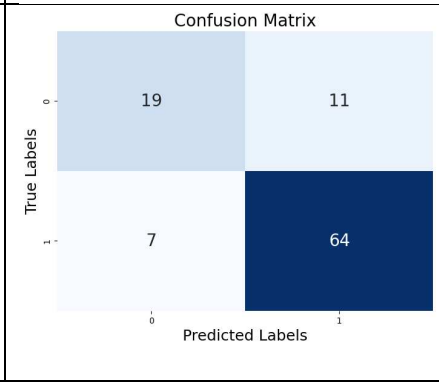
Fig. 3 Final display of result

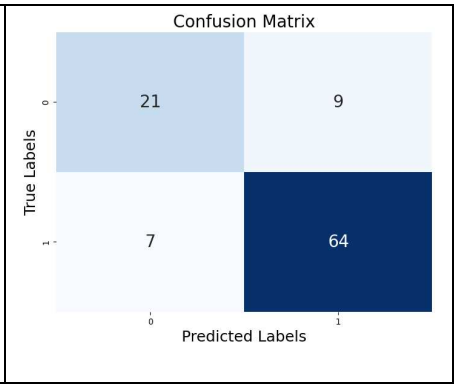
ML model analysis

The performance of every machine learning model is analyzed using performance metrics. These metrics provide us with details about how well the model performs the specified task. To compare the performance of SVM, Naïve Bayes' Classifier, and XGBoost models various metrics such as confusion matrix, precision, recall, F1 score etc. were analyzed.

Table 1 illustrates the confusion matrix generated by the ML models. On comparing the confusion matrices of XGBoost, Naïve Bayes, and SVM models, it can be observed that XGBoost demonstrates a relatively high TP rate, suggesting that it can effectively classify positive cases correctly. Naïve Bayes and SVM also exhibit similarly high TP rates. However, the FP rates of the latter models are lesser than that of XGBoost.

Table 1 - Comparison of the confusion matrices of the ML models

XG Boost	
Naïve Bayes	

SVM	

On comparing the classification report, as seen in *Table 2*, it can be observed that SVM proves to be the most effective model due to its high accuracy and precision values. Although SVM leads in performance, the other models are not far behind, and their metrics are relatively close.

Table 2 - Comparison of the classification report of the ML models

XG Boost	Classification Report: precision recall f1-score support 0 0.67 0.73 0.70 30 1 0.88 0.85 0.86 71 accuracy 0.81 101 macro avg 0.77 0.79 0.78 101 weighted avg 0.82 0.81 0.81 101
Naïve Bayes	Classification Report: precision recall f1-score support not required 0.73 0.63 0.68 30 required 0.85 0.90 0.88 71 accuracy 0.82 101 macro avg 0.79 0.77 0.78 101 weighted avg 0.82 0.82 0.82 101
SVM	Classification Report: precision recall f1-score support not required 0.75 0.70 0.72 30 required 0.88 0.90 0.89 71 accuracy 0.84 101 macro avg 0.81 0.80 0.81 101 weighted avg 0.84 0.84 0.84 101

It can be observed from *Fig. 4* that XGBoost has the highest AUC-ROC value of 0.89, suggesting good overall performance in distinguishing positive and negative instances. SVM, while having the highest accuracy, has the least AUC-ROC value of 0.83. Although SVM has the highest accuracy, with the other models not far behind, when dealing with imbalanced such as resumes whose distribution cannot be determined or predicted, XGBoost seems to be the most suitable due to its ability to distinguish between positive and negative instances across different threshold values. Since the resume data is assumed to be imbalanced, XGBoost is considered the most suitable ML model.

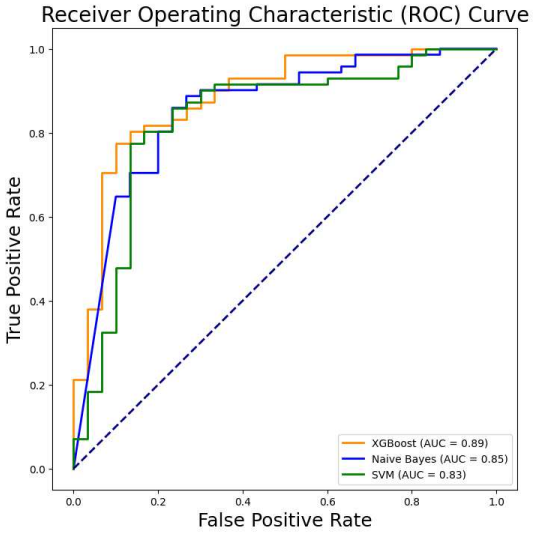


Fig. 4 AUC ROC Comparison

V. Conclusion

On a concluding note, this project focuses on reducing the hassle of the recruitment process by replacing the conventional scenario of manually reviewing resumes to find suitable ones. By using the combined powers of machine learning and web development, the effort and time that needs to be dedicated to the candidate shortlisting process has been eased, thereby making this a convenient method in today's time where an individual for a job vacancy needs to be chosen from a huge pool of applicants who might match to different requirements of the recruiter. By using XGBoost as the ML model, the shortlisting of resumes has been performed with an accuracy of 81 percent. It has also been proved as an ideal model for the resume dataset, based on the observation made from the AUC-ROC graph, thereby choosing it over the SVM and Naive Byes models. In this manner, this project is not only easing the effort and reducing the time put forward for the recruitment process but also optimizing the overall scenario of recruiting a suitable candidate for a job position by ensuring fair assignment of ranking based on the skill sets and qualifications possessed by each individual.

Acknowledgment

All authors have contributed equally to this work.

References

[1] S. Amin, N. Jayakar, S. Sunny, P. Babu, M. Kiruthika and A. Gurjar, "Web Application for Screening Resume," *2019 Int. Conf. Nascent Tech. in Engineering (ICNTE)*, Navi Mumbai, India, pp. 1-7, 2019.

[2] Pradeep Kumar Roy, Sarabjeet Singh Chowdhary, Rocky Bhatia, "A Machine Learning approach for automation of Resume Recommendation system", *Procedia Computer Science,* Volume 167, pp. 2318-2327, 2020.

[3] C. Petersheim, J. Lahey, J. Cherian, A. Pina, G. Alexander, and T. Hammond, "Comparing Student and Recruiter

Evaluations of Computer Science Resumes," in *IEEE Trans. Edu.*, vol. 66, no. 2, pp. 130-138, Apr. 2023.

[4] D. Pant, D. Pokhrel, and P. Poudyal, "Automatic Software Engineering Position Resume Screening using Natural Language Processing, Word Matching, Character Positioning, and Regex," *2022 5th Int. Conf. on Adv. Sys.& Emerg. Tech. (IC_ASET),* Hammamet, Tunisia, pp. 44-48, 2022.

[5] Xiaoyu Luo, "Efficient English text classification using selected Machine Learning Techniques", *Alexandria Engineering Journal,* vol. 60, no. 3, pp. 3401-3409, Jun. 2021.

[6] S. L. Anderson, C. P. Campbell, N. Hindle, J. Price, and R. Scasny, "Editing a Web site: extending the levels of edit," in *IEEE Trans. Prof. Comm.,* vol. 41, no. 1, pp. 47-57, Mar. 1998.

[7] G. Baruffa, M. Femminella, M. Pergolesi and G. Reali, "Comparison of MongoDB and Cassandra Databases for Spectrum Monitoring As-a-Service," in *IEEE Trans. Network & Service Management,* vol. 17, no. 1, pp. 346-360, Mar.2020.

[8] C. Győrödi, R. Győrödi, G. Pecherle, and A. Olah, "A comparative study: MongoDB vs. MySQL," *2015 13th Int. Conf. Engineering of Modern Electric Systems (EMES),* Oradea, Romania, pp. 1-6, 2015.

[9] M. M. Eyada, W. Saber, M. M. El Genidy and F. Amer, "Performance Evaluation of IoT Data Management Using MongoDB Versus MySQL Databases in Different Cloud Environments," in *IEEE Access*, vol. 8, pp. 110656-110668, Jun. 2020.

[10] J. Kumar and V. Garg, "Security analysis of unstructured data in NOSQL MongoDB database," *2017 Int. Conf. on Computing and Communication Technologies for Smart Nation (IC3TSN),* Gurgaon, India, pp. 300-305, 2017.

[11] Namratha, T., Indra Kiran Reddy, B., Deepak Chand Reddy, M.V., Sudheesh, P., "Speech Enhancement Using Nonlinear Kalman Filtering." In: Ranganathan, G., Fernando, X., Shi, F. (eds) *Inventive Communication and Computational Technologies.* Lecture Notes in Networks and Systems, vol 311. Springer, Singapore, 2022.

[12] Akhil, C., Rahul, S., Reddy, K.A., Sudheesh, P., "Car-Like Robot Tracking Using Particle Filter." In: Raj, J.S., Shi, Y., Pelusi, D., Balas, V.E. (eds) *Intelligent Sustainable Systems.* Lecture Notes in Networks and Systems, vol 458. Springer, Singapore, 2022.

[13] C. G. P. Berdanier, M. McCall and G. M. Fillenwarth, "Characterizing Disciplinarity and Conventions in Engineering Resume Profiles," in *IEEE Trans. Profess. Comm*., vol. 64, no. 4, pp. 390-406, Dec. 2021.

[14] Han, J., Kamber, M., & Pei, J., *Data Mining: Concepts and Techniques*, Ed. 3, Waltham, USA: Morgan Kaufmann, 2011.

[15] Adobe Systems Incorporated, *PDF Reference: Adobe® Portable Document Format, Version 1.7*, 6th ed. (2008), Accessed: Oct. 4, 2023. [Online]. Available: https://tinyurl.com/adobepdfreference

[16] Murugan Anandarajan, Chelsey Hill, and Thomas Nolan. "Text preprocessing" in *Practical text analytics: Maximizing the value of text data*, Switzerland: Springer Nature Switzerland AG, pp. 45-59, 2019.

[17] Devaraja G, Krishna Vardhni, Dharshita R and A. Mahadevan, "A Comparative and Analytical Study of Text Classification Models using Various Metrics and Visualizations," *2023 OITS Int. Conf. Information Technology (OCIT)*, Raipur, India, pp. 1-6, 2023.

[18] A. Patle and D. S. Chouhan, "SVM kernel functions for classification," *2013 Int. Conf. Advances in Technology and Engineering (ICATE),* Mumbai, India, pp. 1-9, 2013.

[19] A. Angeles, M. N. Quintos, M. Octaviano, and R. Raga, "Text-Based Gender Classification of Twitter Data using Naive Bayes and SVM Algorithm," *TENCON 2021 - 2021 IEEE Region 10 Conference (TENCON),* Auckland, New Zealand, pp. 522-526, 2021.

[20] Imad Jamaleddyn, Rachid El ayachi, Mohamed Biniz, "An improved approach to Arabic news classification based on hyperparameter tuning of machine learning algorithms", *Journal of Engineering Research,* Volume 11, Issue 2, Art. no. 100061, Jun. 2023.

[21] Xinhua Zhang, "Support Vector Machines", *Encyclopedia of Machine Learning and Data Mining*, Ed. 2, New York, NY, USA: Springer Nature, pp. 1214–1220, 2017.

[22] S. M. Kuriakose, P. Basa Pati and T. Singh, "Prediction of Diabetes Using Machine Learning: Analysis of 70,000 Clinical Database Patient Record," *2022 13th Int. Conf. on Computing Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, pp. 1-5, 2022

[23] I. Belik, "A Comparative Analysis of the Neural Network and Naive Bayes Classifiers," in *SSRN Electronic Journal*, Apr. 2018. [Online]. Available: https://ssrn.com/abstract=3371889

[24] Z. Qi, "The Text Classification of Theft Crime Based on TF-IDF and XGBoost Model," *2020 IEEE Int. Conf. Artificial Intelligence and Computer Applications (ICAICA),* Dalian, China, pp. 1241-1246, 2020.

[25] Jason Brownlee, *XGBoost With Python: Gradient Boosted Trees with XGBoost and scikit-learn*, Machine Learning Mastery, Ed. v1.10, 2016.

[26] X. Chen, D. Yu, X. Fan, L. Wang, and J. Chen, "Multiclass Classification for Self-Admitted Technical Debt Based on XGBoost," in *IEEE Trans. Reliability*, vol. 71, no. 3, pp. 1309-1324, Sept. 2022.