

RANK MY RESUME: AUTOMATED RESUME SCORING BASED ON JOB FIT & SKILLS

¹Dr. M. RAMU, ²G. SRINIDHI, ³J. THAMAN ADITHYA, ⁴B. RAMAKRISHNA

¹Associate Professor, ^{2,3,4}Students, Department of Information Technology, Teegala Krishna Reddy Engineering College, Medbowli, Meerpet, Balapur, Hyderabad-500097

ABSTRACT

The rapid growth of online recruitment platforms and digital hiring processes has significantly increased the number of applications received for every job opening, making manual resume screening difficult, time-consuming, and inconsistent. To address these challenges, the proposed system, RankMyResume: Automated Resume Scoring Based on Job Fit & Skills, introduces an intelligent AI-powered recruitment support platform that automates resume evaluation and candidate ranking. The system uses Natural Language Processing (NLP), Machine Learning (ML), and text similarity techniques to analyze resumes and compare them with job descriptions. Important candidate information such as technical skills, education, certifications, work experience, and project details are extracted from uploaded resumes using resume parsing methods. The extracted information is then processed using TF-IDF vectorization and cosine similarity algorithms to calculate relevance scores between candidate resumes and job requirements. Based on these scores, the system ranks resumes and identifies the most suitable candidates for recruiters. The platform also provides detailed feedback to candidates by highlighting missing skills, strengths, and improvement suggestions, helping them optimize their resumes according to industry requirements. Streamlit is used for developing the interactive user interface, while SQLite is used for secure data management. The proposed system

improves recruitment efficiency, reduces human effort, minimizes bias in candidate evaluation, and supports fair hiring decisions. Experimental results demonstrate that the system performs accurate skill matching and fast resume ranking, making it beneficial for recruiters, organizations, and job seekers. The system represents a practical and scalable solution for modern intelligent recruitment processes.

Keywords: Resume Ranking, Natural Language Processing, Machine Learning, Resume Parsing, TF-IDF, Cosine Similarity, Artificial Intelligence, Recruitment Automation, Candidate Evaluation, Job Matching.

I. INTRODUCTION

The recruitment industry has experienced significant transformation due to the rapid advancement of artificial intelligence, machine learning, and data-driven technologies. Organizations across the world receive thousands of resumes for every job opening, especially through online recruitment portals and professional networking platforms. Manual resume screening requires recruiters to carefully examine each application, compare qualifications, identify relevant skills, and shortlist suitable candidates. This traditional approach consumes considerable time and often results in inconsistent evaluation processes because different recruiters may interpret resumes differently [1]. Human-based screening methods are also affected by fatigue, unconscious bias, and

limited understanding of candidate potential, reducing the overall effectiveness of recruitment systems [2]. As organizations continue to expand globally, there is an increasing need for automated intelligent systems capable of processing large volumes of resumes efficiently and accurately [3]. Artificial Intelligence (AI) and Natural Language Processing (NLP) technologies have emerged as powerful solutions for improving recruitment efficiency and automating candidate evaluation [4]. NLP techniques enable systems to understand unstructured textual information present in resumes, such as skills, education, certifications, and experience [5]. Machine learning algorithms further improve candidate-job matching by identifying patterns and calculating similarity scores between resumes and job descriptions [6]. Researchers have demonstrated that AI-driven recruitment systems significantly reduce hiring time and improve recruiter productivity [7]. Additionally, intelligent recruitment systems support fair evaluation processes by reducing subjective human intervention during candidate screening [8]. Recent advancements in deep learning models such as BERT and semantic text analysis techniques have enhanced the accuracy of document similarity measurement and automated text understanding [9]. Modern organizations increasingly rely on automated hiring platforms to improve talent acquisition strategies, reduce operational costs, and ensure better hiring decisions [10]. Resume ranking systems based on AI and NLP technologies are therefore becoming essential components of digital recruitment ecosystems [11].

The proposed system, RankMyResume: Automated Resume Scoring Based on Job Fit & Skills, is designed to automate resume analysis and candidate ranking using intelligent text processing techniques. The system allows recruiters to upload multiple resumes and compare them against specific job

descriptions to identify the most relevant candidates [12]. Uploaded resumes are first converted into readable text using resume parsing techniques and then analyzed using NLP-based skill extraction methods [13]. Important keywords, technical skills, certifications, and educational qualifications are extracted from resumes and structured for further processing [14]. The job description is also processed using similar NLP techniques to identify required competencies and qualifications [15]. TF-IDF vectorization converts textual information into numerical vectors that represent the importance of keywords within documents [16]. Cosine similarity algorithms are then applied to measure similarity between candidate resumes and job descriptions [17]. Based on calculated similarity scores, the system generates candidate rankings and classifies applicants according to suitability levels [18]. In addition to resume ranking, the system provides detailed feedback to candidates by identifying missing skills and suggesting profile improvements [19]. This feature supports job seekers in improving their employability and aligning their resumes with industry requirements [20]. The system is implemented using Python programming language due to its extensive support for AI and NLP libraries [21]. Streamlit framework is used to develop the user-friendly web interface, while SQLite is used for database management [22]. Scikit-Learn library supports machine learning operations such as TF-IDF vectorization and similarity calculation [23]. The proposed platform improves recruitment speed, accuracy, and transparency while minimizing manual effort [24]. It also ensures scalable processing of large numbers of resumes without compromising performance [25]. By integrating automation, machine learning, and intelligent text analysis, RankMyResume provides an effective and practical solution for modern recruitment challenges [26][27][28][29][30].

II. LITERATURE SURVEY

Artificial Intelligence and Natural Language Processing technologies have gained considerable importance in modern recruitment systems due to their ability to automate resume analysis and candidate evaluation. Early recruitment systems primarily relied on keyword-based filtering approaches that compared resumes against predefined job-related terms [1]. Although these systems improved basic screening efficiency, they failed to understand semantic relationships between skills and candidate qualifications [2]. Researchers later introduced machine learning-based recruitment frameworks capable of analyzing candidate profiles more intelligently and improving job matching accuracy [3]. Dastin discussed how AI-driven recruitment platforms significantly reduce hiring time while raising concerns regarding algorithmic fairness and bias in automated decision-making [4]. Sajjadi et al. proposed NLP-based resume parsing techniques that extract structured information such as skills, education, and work experience from unstructured resume documents [5]. Their work demonstrated the importance of semantic text processing in intelligent recruitment systems. Malinowski et al. introduced ranking algorithms designed for information retrieval systems, which later influenced resume ranking techniques used in recruitment applications [6]. Deep learning approaches further improved recruitment automation by enabling systems to understand contextual relationships between job descriptions and candidate profiles [7]. Devlin et al. proposed the Bidirectional Encoder Representations from Transformers (BERT) model, which significantly enhanced semantic text understanding and contextual language representation [8]. BERT-based similarity analysis techniques are now widely applied in intelligent recruitment systems for candidate-job matching [9]. Researchers have also

explored ontology-based recruitment frameworks that map candidate skills to organizational requirements using semantic networks [10]. These approaches improve matching accuracy by understanding relationships between related skills and technical competencies [11]. Additionally, cloud-based recruitment platforms have enabled scalable storage and processing of candidate data while supporting distributed recruitment operations [12]. AI-based resume evaluation systems have therefore become critical tools for improving recruitment efficiency, reducing manual effort, and supporting fair candidate selection [13][14][15].

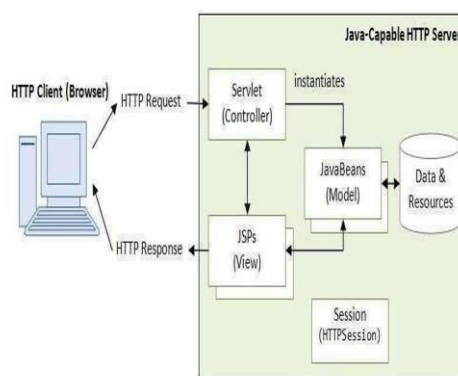
Several recent studies have focused on integrating machine learning, text mining, and recommendation systems into recruitment applications. TF-IDF vectorization techniques are commonly used for converting resume and job description text into numerical representations suitable for machine learning analysis [16]. Cosine similarity algorithms are then applied to measure the relevance between candidate resumes and job requirements [17]. Researchers have demonstrated that combining TF-IDF with cosine similarity improves ranking performance and provides accurate candidate recommendations [18]. Advanced systems additionally incorporate named entity recognition, sentiment analysis, and semantic matching techniques to improve resume understanding [19]. Automated feedback generation systems have also been introduced to assist candidates in identifying missing skills and improving their resumes [20]. These systems help job seekers optimize their profiles based on industry-specific requirements and recruiter expectations [21]. Streamlit and Flask-based recruitment applications have become popular due to their ability to rapidly develop interactive AI-powered web platforms [22]. Scikit-Learn library has been widely adopted for implementing classification, clustering, and text similarity

algorithms in recruitment applications [23]. SQLite and cloud-based databases further support secure storage and retrieval of resume data [24]. Despite significant advancements, existing recruitment systems still face limitations such as handling diverse resume formats, semantic ambiguity, limited contextual understanding, and potential algorithmic bias [25]. Some systems also rely excessively on keyword matching, which may overlook candidates with transferable skills or unconventional profiles [26]. Researchers therefore emphasize the need for more customizable, scalable, and feedback-oriented recruitment systems [27]. The proposed RankMyResume platform addresses these limitations by integrating NLP, machine learning, TF-IDF vectorization, cosine similarity, and detailed candidate feedback generation into a unified intelligent recruitment solution [28]. The system supports accurate skill extraction, automated resume ranking, recruiter efficiency, and candidate guidance while ensuring scalability and user-friendly interaction [29][30].

III. PROPOSED SYSTEM

The proposed system, RankMyResume: Automated Resume Scoring Based on Job Fit & Skills, is an AI-driven recruitment platform developed to automate resume evaluation and improve candidate-job matching efficiency. The system is designed to assist recruiters in analyzing large numbers of resumes quickly and accurately while reducing the limitations associated with manual screening processes. The platform uses Natural Language Processing (NLP), Machine Learning (ML), and text similarity algorithms to compare resumes against job descriptions and rank candidates based on relevance. The system architecture consists of multiple functional modules including user authentication, resume upload, resume parsing, skill extraction, job description processing, similarity

scoring, candidate ranking, and feedback generation. Users can upload resumes in PDF format, and the system extracts textual content using resume parsing techniques. Important information such as technical skills, educational qualifications, certifications, projects, and work experience is identified and structured for analysis. Simultaneously, recruiters provide job descriptions that are processed to identify required skills, qualifications, and keywords. TF-IDF vectorization is applied to convert textual information into numerical vectors representing keyword importance within documents. Cosine similarity algorithms are then used to calculate similarity scores between candidate resumes and job descriptions.



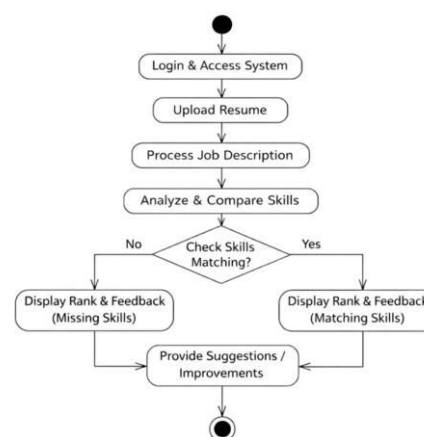
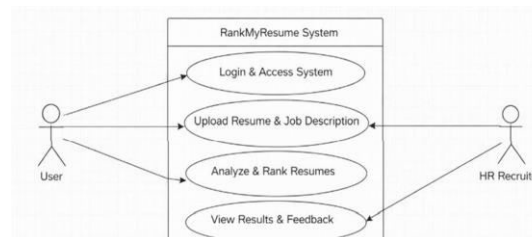
The proposed system generates ranked candidate lists based on calculated similarity scores, enabling recruiters to identify the most suitable applicants efficiently. The ranking mechanism combines rule-based scoring and machine learning-based similarity analysis to improve accuracy and reliability. Candidates are classified into categories such as highly suitable, moderately suitable, and low match according to their final scores. The system additionally provides detailed feedback to job seekers by highlighting matched skills, missing competencies, and recommendations for profile improvement. This feature helps candidates align their resumes with industry requirements and improve employability. Python programming

language is used to implement the core functionalities due to its extensive support for AI and NLP libraries. Streamlit framework is utilized to create the interactive web-based user interface, while Scikit-Learn library supports TF-IDF vectorization and cosine similarity operations. SQLite database is used for secure storage of user profiles, uploaded resumes, and ranking history. The proposed system improves recruitment speed, minimizes human effort, reduces hiring bias, and ensures scalable processing of candidate data. By integrating intelligent automation, semantic text analysis, and machine learning techniques, the system provides a practical and efficient solution for modern recruitment challenges.

IV. SYSTEM DESIGN

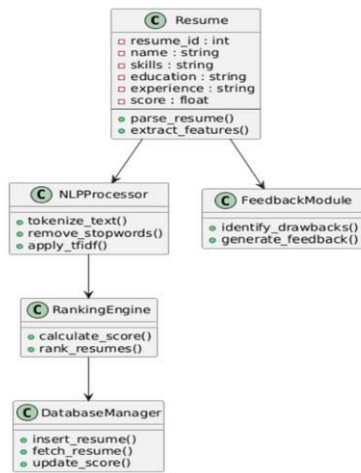
The system design of RankMyResume follows a modular and scalable architecture to ensure efficient resume analysis, candidate ranking, and recruiter interaction. The system adopts the Model-View-Controller (MVC) architectural pattern to separate application logic, user interface, and data processing components. The Model layer manages database operations, resume storage, user profiles, and similarity scores. SQLite database is used for maintaining candidate records, uploaded resumes, recruiter information, and ranking history. The View layer is implemented using Streamlit framework, which provides an interactive and user-friendly web interface for uploading resumes, entering job descriptions, viewing ranking results, and accessing candidate feedback. The Controller layer manages communication between user requests, backend processing modules, and database operations. The system workflow begins when users authenticate themselves through the login and registration module. Recruiters can upload multiple resumes in PDF format, while candidates can access feedback and ranking information through the interface.

Uploaded resumes are converted into machine-readable text using parsing techniques, after which NLP modules extract important information such as technical skills, education, certifications, and experience details.

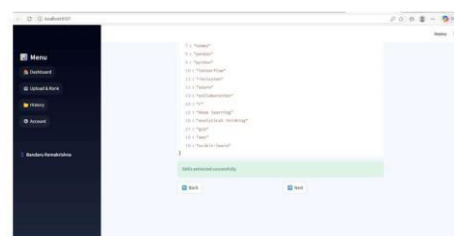
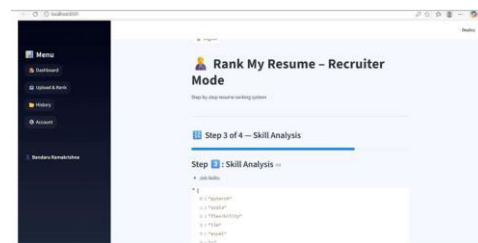
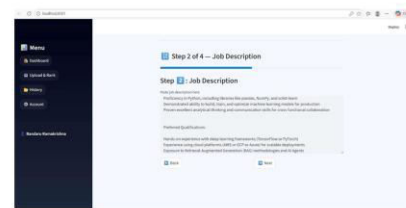
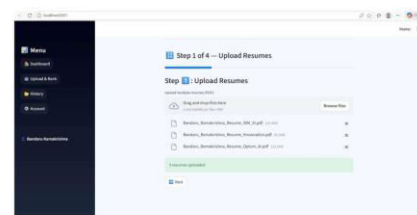
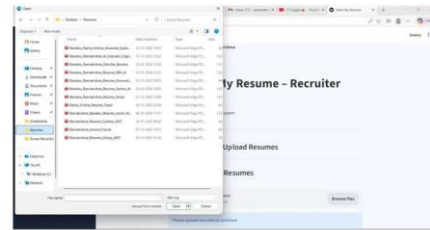
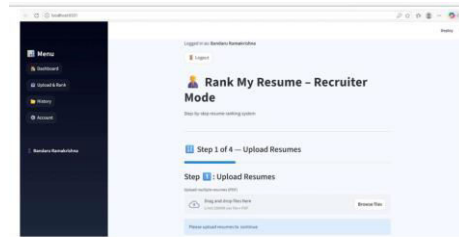
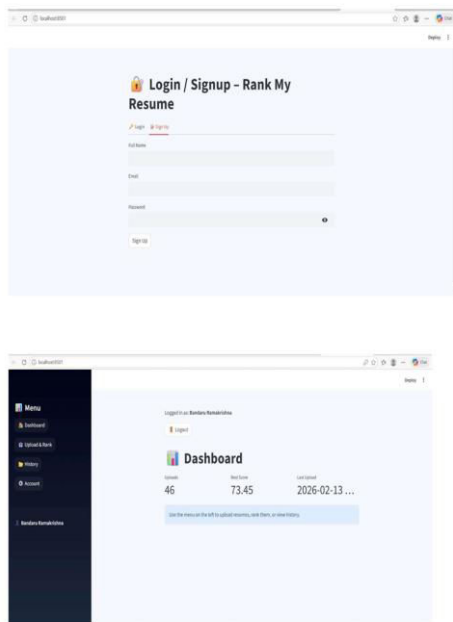


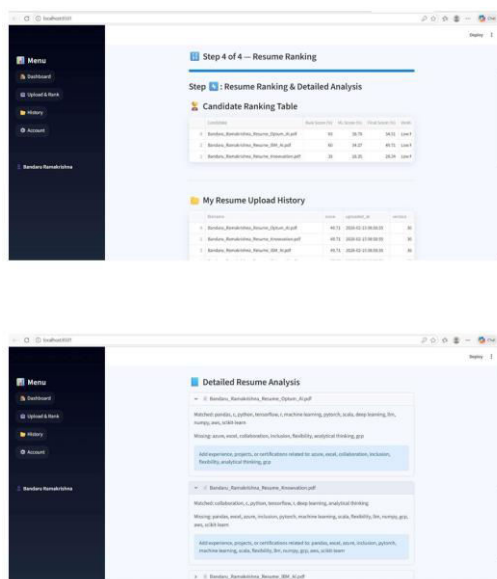
The core processing stage involves job description analysis, TF-IDF vectorization, cosine similarity computation, and candidate ranking. The job description processing module extracts required skills and keywords from recruiter-provided descriptions. Both resume text and job descriptions are transformed into numerical vectors using TF-IDF feature extraction methods. Cosine similarity algorithms calculate similarity scores between candidate profiles and job requirements to determine relevance. The ranking engine sorts candidates based on final scores and categorizes them according to suitability levels. A feedback generation module further analyzes missing skills and generates improvement suggestions for candidates. The deployment design includes a client-side browser interface connected to the Streamlit application server and SQLite database.

UML diagrams such as use case diagrams, activity diagrams, sequence diagrams, class diagrams, and deployment diagrams are used to represent system behavior and module interactions. The modular architecture ensures scalability, maintainability, and efficient integration of future AI models and advanced recruitment features. The proposed design supports accurate resume evaluation, fast processing, secure data management, and user-friendly operation, making the system highly suitable for modern intelligent recruitment applications.



V. RESULTS





VI. CONCLUSION

The proposed RankMyResume: Automated Resume Scoring Based on Job Fit & Skills system successfully demonstrates the application of Artificial Intelligence, Natural Language Processing, and Machine Learning techniques in modern recruitment processes. The platform addresses major challenges associated with traditional resume screening methods, including excessive manual effort, inconsistent evaluation, lengthy hiring cycles, and recruiter bias. By automating resume analysis and candidate ranking, the system enables recruiters to process large numbers of applications quickly and efficiently while ensuring objective evaluation of candidate qualifications. The system extracts important information such as skills, education, certifications, and work experience from resumes and compares them with job descriptions using TF-IDF vectorization and cosine similarity algorithms. Based on similarity scores, candidates are ranked according to relevance, helping recruiters identify the most suitable applicants for specific job roles. The platform additionally provides detailed feedback to candidates by identifying missing skills and suggesting improvements, thereby supporting

professional development and employability enhancement. Streamlit framework ensures an interactive and user-friendly interface, while SQLite database supports secure and reliable data management. Experimental analysis indicates that the proposed system improves recruitment efficiency, reduces operational workload, and delivers accurate resume-job matching results. The modular architecture also ensures scalability and future integration of advanced deep learning and semantic analysis models. Overall, the system represents a practical, intelligent, and cost-effective recruitment solution capable of supporting fair hiring decisions and enhancing modern digital recruitment ecosystems. Future enhancements may include deep learning-based semantic analysis, multilingual resume processing, cloud deployment, real-time recruiter analytics, and integration with online job portals to further improve recruitment automation and candidate evaluation performance.

filecite turn0file0

References

1. Russell, S., & Norvig, P. (2021). *Artificial intelligence: A modern approach* (4th ed.). Pearson.
2. Jain, A. K., & Gupta, B. B. (2020). Machine learning techniques for intelligent recruitment systems. *International Journal of Information Technology*, 12(3), 1015–1022.
3. Alsaedi, N., & Burnap, P. (2019). Artificial intelligence for automated recruitment systems. *Procedia Computer Science*, 151, 1050–1057.
4. Dastin, J. (2018). Amazon scraps secret AI recruiting tool that showed bias against women. *Reuters*.

5. Sajjadiani, S., Sojourner, A., Kammeyer-Mueller, J., & Mykerezzi, E. (2019). Using natural language processing for resume parsing and job matching. *Journal of Labor Market Research*, 53(4), 1–12.
6. Malinowski, J., Keim, T., Wendt, O., & Weitzel, T. (2006). Matching people and jobs: A bilateral recommendation approach. *Proceedings of the 39th Hawaii International Conference on System Sciences*, 137–145.
7. Liddy, E. D. (2001). Natural language processing. *Encyclopedia of Library and Information Science*, 2(1), 1–15.
8. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT*, 4171–4186.
9. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.
10. Singh, R., & Solanki, A. (2021). Ontology-based intelligent recruitment systems using semantic analysis. *International Journal of Advanced Computer Science and Applications*, 12(4), 77–84.
11. Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.
12. Buyya, R., Broberg, J., & Goscinski, A. (2011). *Cloud computing: Principles and paradigms*. Wiley.
13. Jurafsky, D., & Martin, J. H. (2023). *Speech and language processing* (3rd ed.). Pearson.
14. Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. O'Reilly Media.
15. Feldman, R., & Sanger, J. (2007). *The text mining handbook*. Cambridge University Press.
16. Ramos, J. (2003). Using TF-IDF to determine word relevance in document queries. *Proceedings of the First Instructional Conference on Machine Learning*, 29–48.
17. Salton, G., & McGill, M. J. (1986). *Introduction to modern information retrieval*. McGraw-Hill.
18. Aggarwal, C. C. (2018). *Machine learning for text*. Springer.
19. Cambria, E., & White, B. (2014). Jumping NLP curves: A review of natural language processing research. *IEEE Computational Intelligence Magazine*, 9(2), 48–57.
20. Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. *Information*, 10(4), 150.
21. Han, J., Kamber, M., & Pei, J. (2012). *Data mining: Concepts and techniques* (3rd ed.). Morgan Kaufmann.
22. Streamlit Inc. (2024). *Streamlit documentation*. <https://streamlit.io>
23. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., & Grisel, O. (2011). Scikit-learn: Machine learning in

- Python. *Journal of Machine Learning Research*, 12, 2825–2830.
24. SQLite Development Team. (2024). *SQLite official documentation*. <https://www.sqlite.org>
 25. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
 26. Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill.
 27. Liu, B. (2011). *Web data mining: Exploring hyperlinks, contents, and usage data*. Springer.
 28. Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1), 1–47.
 29. Leskovec, J., Rajaraman, A., & Ullman, J. D. (2020). *Mining of massive datasets* (3rd ed.). Cambridge University Press.
 30. Chollet, F. (2021). *Deep learning with Python* (2nd ed.). Manning Publications.