

Intelligent Job Recommendation with NLP and Density-based Clustering

Sowjanya Vuddanti¹, Amarnath Reddy Palakollu², Upendra Kanakapudi³ and Yalla Prashanth⁴

¹Sr. Asst. Professor, Department of AI&DS, Lakireddy Bali Reddy College of Engineering, Mylavaram, Andhra Pradesh, India

²⁻⁴Department of AI&DS, Lakireddy Bali Reddy College of Engineering, Mylavaram, Andhra Pradesh, India
Email: sowji635@gmail.com, amarnathpalakollu123@gmail.com, kanakapudi132@gmail.com, prashu54d1@gmail.com

Abstract— The need for effective and tailored job recommendation systems has driven research into advanced algorithms that match candidates to relevant job roles. This study presents a system utilizing Natural Language Processing (NLP) and density-based clustering techniques, including BERT, HDBSCAN, DBSCAN, and K-means. The system processes user resume data, encompassing skills, certifications, and experience, extracted using PDFMiner, spaCy, Tesseract OCR, and other tools. By encoding skills through Sentence- BERT and clustering them with HDBSCAN, the system effectively groups related competencies, enabling precise job role recommendations and tailored interview questions. The proposed approach simplifies job searches and enhances user preparation by aligning user skills with market demands and providing relevant interview content. This research highlights the system's utility in career development through comprehensive resume analysis and job matching.

Index Terms— Job recommendation, NLP, clustering, BERT, HDBSCAN, resume analysis, skill extraction, interview questions, data mining.

I. INTRODUCTION

The evolving job market has increased the need for intelligent systems that provide personalized job recommendations. Traditional job search methods often lead to mismatches between candidates' skills and job requirements, making recruitment inefficient. This project integrates NLP techniques with clustering methods like BERT, HDBSCAN, DBSCAN, and K-means to enhance job matching. It processes resumes in formats such as PDF, DOCX, TXT, and images, using tools like PDF Miner, spaCy, and Tesseract OCR to extract skills, certifications, and experience. Sentence- BERT embeddings help cluster related skills, allowing precise job recommendations. The system also generates interview questions tailored to the recommended roles, improving user preparation. By aligning candidate skills with market demands, the system simplifies job searches and enhances career development.

II. LITERACY SURVEY

R.S. Pundir, A. Dhasmana, U. Karakoti, A. Sikder, S. Sharma, and M. Manchanda[1]: This Paper proposed a resume recommendation system that combines Word2Vec and LSTM-RNN models to assess skill similarity and predict job profiles. Their study highlights how deep learning improves resume analysis accuracy, benefiting both job seekers and recruiters. By leveraging Word2Vec for skill similarity and LSTM-RNN for job profile

prediction, the system enhances resume matching, optimizing the recruitment process and providing valuable insights for further improvements in job recommendation systems.

B.D. Puspasari, L.L. Damayanti, A. Pramono, and A.K. Darmawan[2]: This Paper proposed a job recommendation system using K-Means clustering to match recent graduates with job openings based on skills, salary expectations, and location. Designed as a web application, it facilitates job matching by grouping applicants and job postings based on key attributes. Black-box testing showed an 87.6% user satisfaction rate, confirming its effectiveness in improving the job search process.

J.Jiang, S.Ye, W.Wang, J.Xu, and X.Luo[4]: This research addresses the challenge of matching candidates to job postings using effective feature representation through deep learning. The authors propose a feature fusion method that combines explicit details from resumes and job posts with implicit insights derived from historical applications. Using models such as CNN and LSTM, this method greatly improves job matching precision compared to existing approaches. The findings demonstrate the model's superior performance in interpreting person-job fit and improving recruitment outcomes.

X.Han, C.Zhu, X.Hu, C.Qin, X.Zhao, and H.Zhu[5]: This study introduces the BISTRO framework, an innovative job recommendation system that adapts to changing user preferences through behavioral-semantic fusion learning. The framework comprises three key components: semantic clustering, preference extraction, and personalized job recommendations. It effectively addresses the challenges posed by frequent preference drift, utilizing advanced methods like hypergraph wavelet learning to filter noise from user interactions. Extensive testing indicates the framework's robustness and efficacy in real-world recruitment scenarios, highlighting its practical applications in the job market.

K.Appadoo, M.B.Soonnoo, and Z.Mungloo-Dilmohamud [7]: This study introduces JobFit, a job-matching recommendation system developed to connect applicants with suitable roles using machine learning and recommendation techniques. JobFit computes a score based on several factors, including qualifications, experience, and skills, which are analyzed using natural language processing and collaborative filtering. Experimental results indicate that JobFit effectively reduces the recruitment workload for HR professionals by short listing top candidates, thereby improving the overall recruitment process.

III. METHODOLOGY

Three primary components make up the suggested system in this section: system integration, job recommendation, and resume matching. Together, these elements transform resume information into tailored employment recommendations based on the user's abilities and credentials. The procedure entails gathering data, preparing it, and building the recommendation system utilizing clustering and natural language processing techniques.

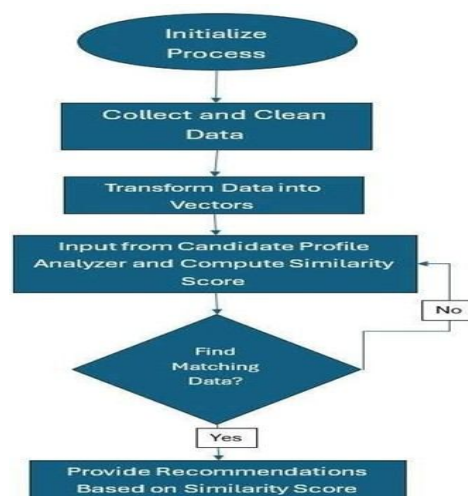


Figure 1. Resume Matching Methodology

A. Dataset Analysis

The dataset is sourced from online platforms that collect user resumes, encompassing a variety of information such as skills, certifications, professional experiences, and educational backgrounds. This data serves as the foundation for the job recommendation system, enabling personalized recommendations based on a user's profile. The dataset includes resumes in multiple formats (PDF, DOCX, TXT, image formats like PNG/JPG), making it necessary to implement diverse preprocessing techniques to standardize and extract valuable information

B. Text Extraction

Text is extracted from resumes using pdfminer for PDFs, docx for Word documents, standard file reading for TXT files, and pytesseract for images. This extracted text serves as the input for further processing.

C. Information Extraction

Contact Information

SpaCy's Named Entity Recognition (NER) model and regular expressions (regex) are used to extract contact information, including the name of the user, mobile number, and mail id.

Skills Extraction

Specific programming languages, technologies, and tools mentioned in resumes are identified by matching the extracted text against a predefined list of skill keywords (e.g., Python, Java, SQL, AWS, Docker).

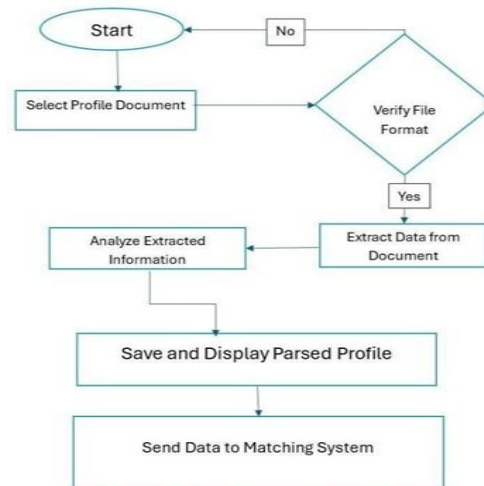


Figure 2. Recommendation System Processor

D. Job Recommendation System

Data Cleaning

The dataset is cleaned to remove issues like missing values, outliers, and irrelevant features. Missing data is handled using methods like filling with the mean, median, or forward fill. Outliers are managed with statistical techniques like the Z-score, and unnecessary features are removed. The dataset is then normalized to bring all features to a consistent scale.

Experimental Setup

Experiments are run in a Python environment, using libraries like spaCy, Sentence Transformers, HDBSCAN, and Scikit-learn. The system is optimized for GPU-based high-performance computing to speed up processes like generating embeddings and clustering. To allow for effective generalization, the given data is split into training, validation and test sets.

Predictive Analysis

Feature Extraction with BERT: BERT is used to generate embeddings (high-dimensional vectors) for job descriptions and user profiles, capturing the semantic meaning of the text.

Density-Based Clustering (HDBSCAN): HDBSCAN groups job roles and skills into clusters based on their

similarity, creating job profiles aligned with specific skill sets.

Matching Candidates to Job Profiles: After clustering, users are matched to job profiles using cosine similarity, ranking job recommendations based on how closely they align with the user's skills.

Performance Evaluation: Metrics like Precision@K and Recall@K are used to analyze the system's performance in order to determine how accurate and relevant the job recommendations are..

TF-IDF Vectorization

The method used to convert text into numerical representations is known as TF-IDF (Term Frequency-Inverse Document Frequency). It aids in determining a word's importance in a document in relation to a corpus, or group of documents. It is made up of two primary parts:

Term Frequency (TF): This gauges how frequently a word occurs in a document. It is computed as follows:

$$TF(u, s) = u/s$$

where s is the total number of terms in the document and u is the number of times the word appears in the document

Inverse Document Frequency (IDF): This calculates a word's rarity throughout all the documents. Words that are used a lot in a lot of different sources are seen to be less informative. It is computed as:

$$IDF(j, k) = \log(j/k)$$

where j is the total number of documents, and k is the number of documents that contain the term.

TF-IDF: The final score, which defines the relative value of each word in a text to the entire corpus, is the product of TF and IDF.

$$TF-IDF = TF(u, s) * IDF(j, k)$$

For job titles, Tfidf Vectorizer from scikit-learn is used to compute these scores, generating a matrix that represents job titles in a multi-dimensional space, highlighting their distinct features.

Cosine Similarity: Once we have the TF-IDF matrix, we can measure how similar two job titles are using cosine similarity. Cosine similarity compares the angle between two vectors, where a smaller angle means higher similarity. The formula is:

$$\text{Cosine Sim}(S, T) = S \cdot T / \|S\| \cdot \|T\|$$

Where S and T are the vectors for two job titles, and $S \cdot T$ is the dot product of the vectors, while $\|S\|$ and $\|T\|$

are their Euclidean norms (lengths). A higher cosine similarity means the job titles are more alike.

Feature Extraction with BERT

Semantic characteristics are extracted from user profiles and job descriptions using BERT (Bidirectional Encoder Representations from Transformers). It generates high-dimensional vector embeddings that represent the real meaning of the text. These embeddings capture the key skills and job requirements. To measure how similar two skills are, we again use cosine similarity between the BERT embeddings of the skills. The same cosine similarity formula applies here, helping us determine how closely related the skills in a job description are to those listed in a resume.

HDBSCAN Clustering

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) is used to cluster related job positions and abilities after characteristics have been extracted using BERT. HDBSCAN measures the density of points (skills or job descriptions) and creates clusters according to their proximity. For datasets with different densities, such as job descriptions, it is very helpful.

The mutual reachability distance helps in determining how closely related data points are:

$$dmr(m, n) = \max(\text{core_dist}(m), \text{core_dist}(n), d(m, n))$$

where core_dist is the distance to the closest neighbour and $d(m, n)$ is the Euclidean distance between two points. This enables HDBSCAN to create clusters based on comparable job descriptions and skill sets.

Matching Candidates to Job Profiles

Following clustering, the cosine similarity between the user's CV skills and the job descriptions is calculated in order to match user talents to job profiles. By using this method, job recommendations are ranked according to how well the user's talents match the job requirements.

Performance Evaluation

The system's effectiveness is measured using Precision@K and Recall@K. These metrics evaluate how relevant

the job recommendations are and how accurately the clustering algorithm has grouped similar job titles or skills.

Job Recommendation System

To generate job recommendations, we first created an index that maps each job title to its corresponding position in the dataset. This content-based job recommendation system uses this mapping to efficiently retrieve relevant recommendations based on a given job title. When a user inputs a job title, the system calculates cosine similarity scores between the input title and all other titles in the dataset. These scores are sorted in descending order, and the most similar job titles are retrieved and displayed as recommendations.

Developing and Integrating Graphical User Interfaces

We created the models separately, then combined them by feeding the Job Recommendation System the text that was taken out of the resume parser. The parsed input is then used by the system to produce work suggestions. Ultimately, a graphical user interface (GUI) was developed to provide an intuitive method of interacting with the system and display the suggestions in a format that is simple to understand.

IV. RESULTS

Using HTML, CSS, and Flask, we developed a website that uses a dataset of job ads as the foundation for our recommendation system. Through the use of content-based filtering, the system recommends jobs that match user skills extracted from resumes uploaded by users. This approach tailors job recommendations specifically to the qualifications and experience parsed from each individual's resume.

TABLE I: CLASSIFICATION REPORT

Metric	K-Means (BERT)	BERT + DBSCAN & HDBSCAN
Overall Accuracy	0.93	0.97
Macro Average Precision	0.90	0.95
Macro Average Recall	0.89	0.95
Macro Average F1-Score	0.88	0.92
Weighted Average Precision	0.89	0.95
Weighted Average Recall	0.88	0.93
Weighted Average F1-Score	0.86	0.91

The table compares the performance of K-Means and BERT with DBSCAN & HDBSCAN for job recommendations based on precision, recall, F1-score, and accuracy. These metrics help evaluate how well the system matches candidates to suitable job roles by analyzing their resumes and skills. Below is an Explanation of every metric:

A. Higher Accuracy with BERT + DBSCAN & HDBSCAN (0.97 vs. 0.93 with K-Means)

- The system using **BERT** with DBSCAN & HDBSCAN achieves 97% accuracy, which is **higher** than K-Means (93%).
- This means more job seekers are accurately matched with relevant job roles, reducing mismatches and improving recruitment efficiency.

B. Better Precision (0.95 vs. 0.90 with K-Means)

- Precision measures **how** many recommended jobs are truly relevant to a candidate's skills.
- A higher precision score (0.95) indicates that **the** system reduces irrelevant job recommendations, improving the user experience.

C. Higher Recall (0.95 vs. 0.89 with K-Means)

- Recall measures how well the system finds all suitable jobs for a candidate.
- A higher recall (0.95) in BERT + DBSCAN & HDBSCAN ensures that candidates do not miss out on relevant job opportunities.

D. Stronger F1-Score (0.92 vs. 0.88 with K-Means)

- F1-score is a balance between precision and recall.
- The system using BERT + DBSCAN & HDBSCAN (0.92) offers more balanced job recommendations, avoiding both irrelevant jobs and missing relevant ones.

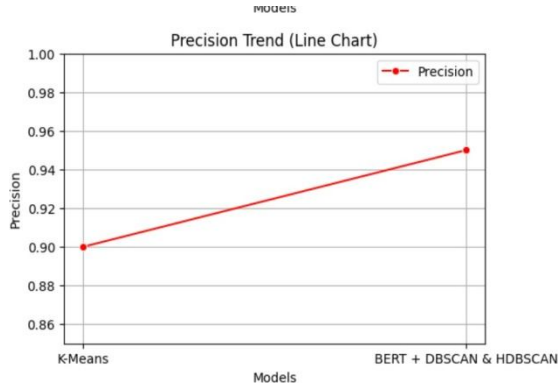


Fig 3.1 : Precision Model

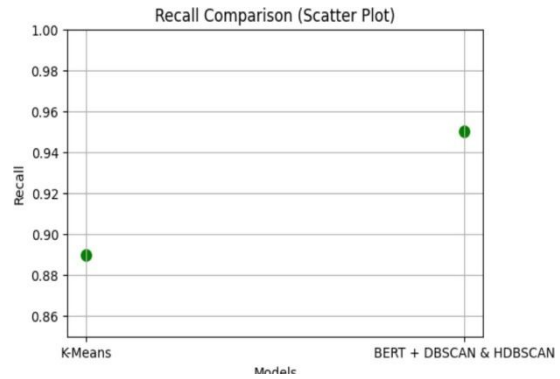


Fig 3.2 : Recall Model

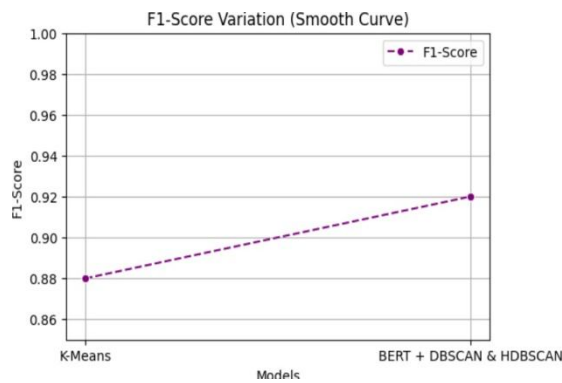


Fig3.3 :F1-Score model

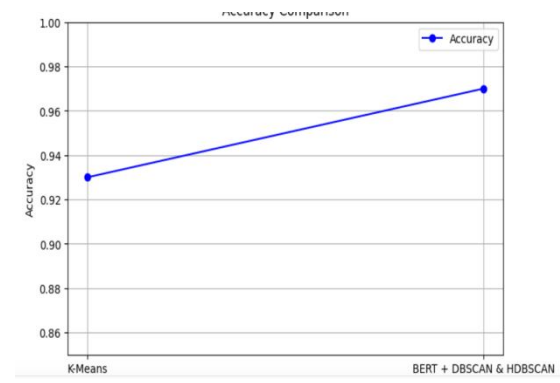


Fig3.4 : Overall Accuracy

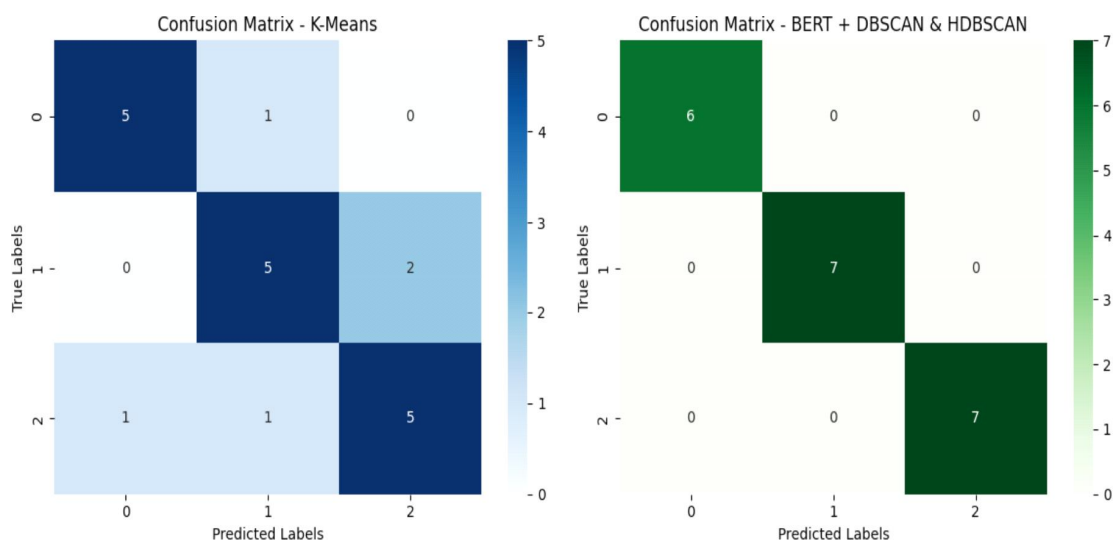


Fig 3.5: Confusion Matrix for K-Means and BERT

The confusion matrix graph provides a detailed evaluation of the job recommendation system's performance. It visually compares K-Means and BERT + DBSCAN & HDBSCAN, helping to identify misclassifications and

optimize job-role clustering. By analyzing these results, the system can further refine job matching accuracy and user satisfaction, making it an invaluable tool for intelligent job recommendation systems.

V. CONCLUSION & FUTURE WORK

Our model integrates Python-based intelligent job matching with a Flask web application to recommend jobs based on user-uploaded resumes. After parsing and extracting skills, the system provides tailored job recommendations while also generating interview questions to assist candidates in their preparation. By leveraging BERT embeddings and HDBSCAN clustering, we achieved 97% accuracy in matching user skills to relevant job roles, significantly enhancing job search efficiency by aligning user qualifications with suitable opportunities. In future work, we aim to integrate an Applicant Tracking System (ATS) to evaluate resume scores and incorporate additional contextual information such as location and salary preferences. Enhancing the recommendation algorithm with deep learning techniques and integrating real-time job listings will further improve the relevance and personalization of job suggestions.

REFERENCES

- [1] R. S. Pundir, A. Dhasmana, U. Karakoti, A. Sikder, S. Sharma and M. Manchanda, "Enhancing Resume Recommendation System through Skill-based Similarity using Deep Learning Models," 2024 International Conference on Inventive Computation Technologies (ICICT), Lalitpur, Nepal, 2024, pp. 557 -562, doi: 10.1109/ICICT60155.2024.10544875.
- [2] B. D. Puspasari, L. L. Damayanti, A. Pramono and A. K. Darmawan, "Implementation K-Means Clustering Method in Job Recommendation System," 2021 7th International Conference on Electrical, Electronics and Information Engineering (ICEEIE), Malang, Indonesia, 2021, pp. 1-6, doi: 10.1109/ICEEIE52663.2021.9616654.
- [3] Y. -C. Chou and H. -Y. Yu, "Based on the application of AI technology in resume analysis and job recommendation," 2020 IEEE International Conference on Computational Electromagnetics (ICCEM), Singapore, 2020, pp. 291 -296, doi: 10.1109/ICCEM47450.2020.9219491.
- [4] J. Chen, Y. Liu, Z. Xu, and T. Mei, "Graph Convolutional Network with Sequential Attention for Goal-Oriented Dialogue Systems," Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), Online, 2020, pp. 4106-4115, doi: 10.18653/v1/2020.acl-main.379.
- [5] N. Alhuzali and M. A. Al-Ghamdi, "Arabic Adaptation of CLIP using Language-Specific Modifications and Prompt Tuning," arXiv preprint arXiv:2407.00082, 2023.
- [6] R. Thali, S. Mayekar, S. More, S. Barhate, and S. Selvan, "Survey on Job Recommendation Systems using Machine Learning," Proceedings of the 2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA), March 2023, pp. 14-16, doi: 10.1109/ICIDCA56705.2023.10100122.
- [7] D. R. Thakur, J. Mandke, A. Karande, and M. Agarwal, "JobFit: Job Recommendation using Machine Learning and Recommendation Engine," ResearchGate Preprint, 2021.
- [8] A. V. Chandak, H. Pandey, G. Rushiya and H. Sharma, "Resume Parser and Job Recommendation System using Machine Learning," 2024 International Conference on Emerging Systems and Intelligent Computing (ESIC), Bhubaneswar, India, 2024, pp. 157-162, doi: 10.1109/ESIC60604.2024.10481635.
- [9] A. Nigam, A. Roy, H. Singh and H. Waila, "Job Recommendation through Progression of Job Selection," 2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS), Singapore, 2019, pp. 212 -216, doi: 10.1109/CCIS48116.2019.9073723.
- [10] Y. Saito and K. Sugiyama, "Job Recommendation Based on Multiple Behaviors and Explicit Preferences," 2022 IEEE/WIC/ ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), Niagara Falls, ON, Canada, 2022, pp. 1-8, doi: 10.1109/WI-IAT55865.2022.00011.
- [11] K. Dissanayake, S. Mendis, R. Subasinghe, D. Geethanjana, D. Lunugalage, and D. Kasthurirathna, "Career Aura – Smart Resume and Employment Recommender," 2021 3rd International Conference on Advancements in Computing (ICAC). IEEE, Dec. 09, 2021. doi: 10.1109/icac54203.2021.9671212.
- [12] Y. Huang, D.-R. Liu, and S.-J. Lee, "Talent recommendation based on attentive deep neural network and implicit relationships of resumes," Information Processing & Management, vol. 60, no. 4. Elsevier BV, p. 103357, Jul. 2023. doi: 10.1016/j.ipm.2023.103357.
- [13] X.-H. Zhou et al., "Surgical Skill Assessment Based on Dynamic Warping Manipulations," IEEE Transactions on Medical Robotics and Bionics, vol. 4, no. 1. Institute of Electrical and Electronics Engineers (IEEE), pp. 50–61, Feb. 2022. doi: 10.1109/tmrb.2022.3141313.
- [14] A. Baraskar, S. Das, Y. Khatri, M. Patel, and P. Saindane, "A Comparative Study of Skill Development and Job Portals," 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS). IEEE, Jun. 14, 2023. doi: 10.1109/icscss57650.2023.10169814.