International Journal on Robotics, Automation and Sciences

A Comprehensive Review on Machine Learning-Based Job Recommendation Systems

Rui-Ern Yap, Su-Cheng Haw* and Shaymaa Al-Juboori

***Abstract* – A dynamic, constantly shifting labor market creates enormous job postings, overwhelming candidates and making it difficult for businesses to find quality candidates. It is also hard for job seekers to find suitable jobs. Addressing these issues, machine learning-driven job recommender systems have recently become an essential tool using predictive models to improve the match between jobs and candidates. A hybrid design that combines collaborative filtering with content-based filtering and adds contextual information like geographic location, industry trends, and user behavioural data can enhance the accuracy and relevance of recommendations. This paper reviews and critically analyzes contemporary job recommender system techniques. The focus is on hybrid recommendation models and the integration of algorithmic approaches, indicating their strengths and weaknesses. This review also looks into the evaluation metrics like precision, recall, normalized discounted cumulative gain (NDCG), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE). To provide an overall perspective of the various approaches employed and the performance trade-offs inherent therein, this paper hopes to shed some light on the optimization of job recommendation systems for better effectiveness and user satisfaction.**

Keywords—Recommender System, Machine Learning, Hybrid-based, Job Recommender, Evaluation Metrics, Comprehensive Review.

I. INTRODUCTION

In this highly dynamic labor market, with many openings and the candidate profile being diverse, both the job seekers and the employers are confronted with enormous challenges. Traditional matching processes through manual screening or basic keyword searches are usually not effective in finding complex skill sets and, therefore lead to mismatches, inefficient hiring, and suboptimal employment outcomes [1]. Problems of sparsity, scalability, and cold start only to add to the complications of effectively personalizing the recommendation. Although approaches like collaborative filtering and content-based filtering have reduced some of the misery, they are afflicted by synonyms in skill descriptions, adaptive job positions, and the ability to widen existing biases [2].

Recent breakthroughs in Machine Learning (ML), deep learning, and Natural Language Processing (NLP), therefore, offer a range of thrilling possibilities to overcome those limitations. Modern recommender systems are now able to identify advanced patterns or connections between job seekers and job postings due to enormous mining datasets. It is now achievable to use NLP techniques to parse job postings and resumes into more accurate versions, thus creating more enhanced feature extractions and better skill-matching opportunities. The combination of these approaches has led to more robust, more sophisticated systems capable of coping with the

*Corresponding Author email: sucheng@mmu.edu.my, ORCID: 0000-0002-7190-0837
Rui-Ern Yap is with Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia (e-mail: 1211105182@student.mmu.edu.my).
Su-Cheng Haw is with Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia (e-mail: sucheng@mmu.edu.my).
Shaymaa Al-Juboori is with School of Computing, Engineering and Mathematics, University of Plymouth, UK (shaymaa.al-juboori@plymouth.ac.uk)

International Journal on Robotics, Automation and Sciences (2025) 7, 2:36-55
https://doi.org/10.33093/ijoras.2025.7.2.5
Manuscript received:19 Mar 2025 | Revised: 29 Apr 2025 | Accepted: 10 May 2025 | Published: 30 Jul 2025

Published by MMU PRESS. URL: http://journals.mmupress.com/ijoras

inherent dynamism and complexity of the contemporary labor market [3].

This review paper introduces the background of job recommendation systems, examines the shortcomings of traditional approaches, and discusses emerging ML-based methods for effective matchmaking. It emphasizes identifying key features for performance enhancement, the techniques of bias evasion and fairness, and methods for using user feedback for improved recommendation accuracy. The primary aim is to inform researchers and practitioners of cutting-edge practices that have the potential to transform the recruitment process so that it would be beneficial to both employers and job seekers.

II. Recommender System

A. Overview of Recommender System

Recommender systems are sophisticated tools for navigating and interacting with large, complex information environments. By using information filtering techniques, they generate personalized suggestions or predictions from historical data, user interactions, and individual preferences. This approach greatly enhances user satisfaction and experience, as it helps uncover products, services, or content that might otherwise go unnoticed. Through pattern and preference analysis, recommender systems offer tailored recommendations that save users both time and effort by aligning choices with their interests.

Shin [4] highlighted that user-item interactions are often captured in an interaction matrix, where rows represent users, columns represent items and matrix entries typically indicate ratings, clicks, or binary purchase signals. This matrix underpins many industry applications—such as Netflix, YouTube, Tinder, and Amazon—that analyze user behaviour to suggest related movies, videos, potential matches, or products. For instance, Amazon recommends products based on past purchase histories, while Netflix tailors series and film suggestions based on previously viewed content, thus enhancing user engagement.

Collaborative filtering, matrix factorization, and other situation-based algorithms Fayyaz et al. [5] commonly rely on this interaction matrix as a foundational element for identifying patterns in user behaviour and generating accurate, customized recommendations. Extending these concepts, Dhananjaya et al. [6] proposed a personalized recommendation system in education that integrates collaborative filtering, content-based methods, and hybrid techniques, supported by emerging ML and Artificial Intelligence (AI) solutions. Such advanced systems address issues like information overload, language barriers, and outdated content, holding promises for more inclusive, scalable, and adaptive learning environments in schools, higher education, and corporate training.

Latha & Rao [7] take a similar approach with a Convolutional Neural Network (CNN)-based product recommendation system for e-commerce, focusing on accuracy and efficiency. Their model overcomes long computation times and limited domain-learning capacities by leveraging Term Frequency-Inverse Document Frequency (TF-IDF) for feature extraction and applying various preprocessing steps—stemming, lemmatization, and stop word removal—to refine customer reviews for sentiment analysis, thereby enhancing product recommendations.

In the realm of job searching, recommender systems have transformed how job seekers find positions and how employers identify candidates. By narrowing down extensive job listings into a curated set of relevant opportunities, these systems save considerable time and effort for applicants. Gugnani et al. [8] described advanced algorithms that assess resumes, skills, and preferences to deliver accurate, personalized job recommendations. Meanwhile, Mhamdi et al. [9] show how platforms like LinkedIn employ hybrid recommender methodologies—combining content-based and collaborative filtering—to tailor suggestions based on a user's professional network and behavioural patterns. Services such as Glassdoor and Monster similarly integrate recommender systems to refine job matching and improve user experience.

The adoption of AI, ML, and context-aware data processing continues to push job recommendation systems toward more intelligent, inclusive, and efficient solutions. These advances account for dynamic factors such as location, timing, and industry trends, addressing many of the challenges in modern labor markets. Consequently, job recommendation systems are not only reshaping recruitment processes but also fostering an interactive, adaptable environment where both job seekers and employers can more readily meet their goals [10].

B. Recommender System Techniques

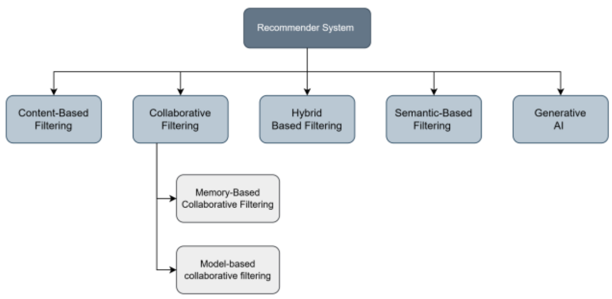

FIGURE 1. Recommender System Techniques.

1) Content-Based Filtering

Content-based filtering works by comparing active user preferences with similar items or features of those items. It builds a profile for the user based on explicit interactions such as purchases, ratings, and searches or implicit interactions derived from behaviour. Profiles are matched against characteristics, such as object colour, book author, or movie cast, to predict items a user will interact with. It is further strengthened in content-based filtering by various techniques, such as finding the important textual features by TF-IDF, finding the similarity based on vectors by Cosine Similarity, and grasping semantic relationships by Word Embeddings such as Word2Vec and FastText. Bayesian Classifiers predict preferences based on probabilistic models, whereas Ensemble Techniques merge several models for robustness and accuracy. Context-Aware Filtering: This is the process of refining recommendations about time, location, and even mood. All these put together contribute to the quality and adaptability of recommendations in manifold domains.

Negara et al. [11] used a TF-IDF methodology to represent the dataset, changing data from textual descriptions such as those in NFT naming and features to numerical vector forms that denote the importance of terms regarding the data set. For a given term, TF calculates about normalizing factors against document length, a measure of a term occurrence frequency in the document. However, IDF considers terms in higher weights where the appearance of the dataset is low. The combined TF-IDF score reflects a term’s overall relevance. After vectorizing the data using TF-IDF, they calculated cosine similarity to measure the similarity between items. This mathematical technique evaluates the similarity of two vectors by examining the cosine of the angle between them. It enables the system to discover NFT items with high similarity scores and provide recommendations based on user preferences or behaviours. The content-based filtering method worked effectively, producing accurate and important recommendations, making it particularly appropriate for NFT marketplaces with limited user rating information.

Sakti et al. [12] presented a music recommendation system that uses a content-based filtering approach and the Euclidean distance algorithm to suggest songs based on the user's mood. The system is based on James Russell's Circumplex Model, which positions emotions in a two-dimensional space where valence (positive or negative) is represented on the x-axis and energy (arousal or intensity) is shown on the y-axis. It collects the user's mood input, based on valence and energy statistics, and then calculates the Euclidean distance between the user's mood and the mood coordinates of the songs in the dataset. The Euclidean distance measures the direct distance between two places in n-dimensional space. Hence, this would find those songs whose mood comes closest to that of the user. Recommendation functionality will make sure the system effectively pairs songs with user moods. It uses the Normalized Discounted Cumulative Gain (NDCG) metric to test the quality of recommendations for relevance in ranked results, showing a highly relevant and satisfying approach. This effectively merges content-based filtering with the Euclidean distance algorithm, allowing for customized music suggestions.

2) Collaborative Filtering

A commonly used ML method, collaborative filtering, provides the foundation of many modern recommender systems. Collaborative filtering mainly relies on the prediction of user preferences or behaviours using group collective preferences. It operates under the theory that those with similar past tastes or preferences will most likely behave similarly going forward. That makes it an effective way to personalize recommendations for a user's unique interests, depending on patterns of collective user behaviour and not on the explicit content information.

In collaborative filtering, the matrix usually denotes the interaction of users with items in a system, such as ratings, purchases, or views. Each row of the matrix corresponds to a user, and each column corresponds to an item, with entries representing the interactions of the user with that item. This matrix forms the basis for analysing user behaviour and drawing insights about similarities and preferences Lara-Cabrera et al. [13].

This knowledge in the matrix generates a vector space where people and objects become points. Various similarity measures cosine similarity, Pearson correlation, and Euclidean distance help to define the space between these points. Such measures provide a numerical means for comparing products or users. These metrics provide quantitative ways to determine the likeness of users or items. For instance, when two users have similar preferences over a set of items, these two users are considered similar, enabling transferring the preferences of one user to recommend items for the other.

Memory-Based Collaborative Filtering*:* This is a fundamental technique for recommender systems, based on the concept that consumers with relevant prior preferences will likely make similar choices in the future. Similarly, items rated or interacted with in comparable ways by multiple users are likely to appeal to individuals with similar behaviours. This approach leverages historical user-item interaction data to learn the pattern and relationship among users or items for personalized recommendations.

This approach directly utilizes the raw user-item interaction data, like ratings, purchase records, or clicks, to calculate similarity scores without creating a separate predictive model or employing complex algorithms. Chen et al. [14] compute similarity scores among users or items utilizing metrics such as Pearson correlation, cosine similarity, or Euclidean distance. In user-user filtering, the suggestions are generated by finding users with the most similar preferences, or neighbours, and combining their ratings for items the target user has not yet interacted with.

A key advantage with memory-based collaborative filtering is the straightforwardness and ease with which it can be explained. Recommendations are produced directly from the interaction data, with no additional training or complex models required. This approach is light and easy to implement and suitable for real-time applications. At the same time, it has several very significant drawbacks: issues of data sparsity, especially when the interaction matrices are filled with unobserved entries such as potential issues for cold start in cases of new users or items; scalability when datasets get large.

However, memory-based collaborative filtering remains applicable to these issues when data is relatively dense and speedy, and simple recommendations are required instantly. The intuitive nature of it and the necessity for direct interaction data mean it will continue to be advantageous in many contexts, especially in those where computational ease and interpretability are central to substantial questions.

Model-based collaborative filtering: This is an advanced approach in recommender systems that allows the detection of hidden patterns and relationships in the data about user-item interaction using statistical and ML models. Unlike memory-based approaches that rely on raw data and pre-defined similarity metrics, model-based methods learn predictive models that generalize user preferences for items they have never interacted with. These models examine past user interactions by using reviews, purchases, or clicks to identify latent patterns in user behaviour and item properties. Calculating these trends helps the system forecast human interaction with identifying objects, so model-based collaborative filtering is an excellent tool for customised recommendations.

Horasan et al. [15] A unique aspect of this method is its capacity to uncover latent factors—unseen variables that clarify observed actions or preferences. For example, latent factors in a film suggestion system might signify abstract ideas such as genre inclinations, visual aesthetics, or thematic intricacy. Methods such as Matrix Factorization, Neural Networks, and Probabilistic Models are frequently utilized to derive these factors. Furthermore, advanced techniques such as Non-negative Matrix Factorization (NMF) and Truncated ULVD (T-ULVD) are recommended to improve scalability and address large datasets' sparsity. T-ULVD provides reduced computational complexity and greater dynamic adaptability than conventional SVD. Minimizing dimensions and refining the interaction matrix with these methods enhance the effectiveness and precision of suggestions.

Moreover, model-based collaborative filtering offers various benefits over memory-based approaches. To begin with, it is exceptionally scalable and offers efficient solutions for extensive, sparse datasets by emphasizing generalizable patterns instead of comparing entities. It is also easily adaptable for a system to provide precise and reliable recommendations in changing situations concerning users' tastes and the traits of items. These render it suitable for numerous real-world applications involving intricate datasets and personalization needs, like as e-commerce, streaming platforms, and social media.

It faces difficulties in model-based collaborative filtering. Model building and training require substantial computation and expertise in algorithm selection and fine-tuning. The cold start problem exists when items are recommended for new users or newly added items. A hybrid approach can be devised, combining model- and content-based techniques to overcome this. Nevertheless, the possibility of highly personalized and flexible recommendations made model-based collaborative filtering the cornerstone of modern recommender systems.

3) Hybrid Based Filtering

A hybrid recommendation system is an advanced structure that integrates a minimum of two recommendation methodologies, enhancing accuracy, diversity, and the efficacy of personalized suggestions. These systems exceed the constraints of individual approaches in handling issues such as data sparsity, scalability challenges, and cold-start problems. Consequently, hybrid systems provide more robust recommendations by leveraging multiple algorithms and diverse data sources, including user behaviour, item content, demographic information, and contextual data. Hybrid systems can be seen in many practical applications such as e-commerce, music streaming, and entertainment; therefore, hybrid recommendation platforms, like Netflix, depend on one of the most comprehensive hybrid methods to provide personalized recommendations for movies and TV series.

Two popular techniques widely used in hybrid systems are content-based and collaborative filtering. Content-based filtering considers the attributes of the items or the metadata with which a user has had previous interactions, recommending more similar items to them. On the other hand, collaborative filtering does this analysis on the pattern of interaction a user creates around items and finds similar users or similar items to produce recommendations for the active user. While each method has certain strengths, they also have a number of limitations: content-based filtering may lack diversity and novelty, while collaborative filtering faces cold-start problems and data sparsity. A hybrid approach balances these weaknesses by merging the best of both techniques.

Hybrid systems can be developed in various ways, depending on the application and the available data. Weighted hybrids assign different importance to various algorithms and then combine the outputs using weighted averages. Switching hybrids adaptively chooses the best algorithm depending on the user or context. Feature combination involves merging the features from several methods into one predictive model. In contrast, cascade hybrids refine the results sequentially, using the output of one method as the input for another. Meta-level hybrids include the model of one algorithm as an input to another, while mixed hybrids present the outputs from multiple algorithms side by side. Other more sophisticated forms include ensemble hybrids, which leverage ensemble learning techniques, whereas blended hybrids are domain-specific combinations tailored for needs.

Mazlan et al. [16] show that hybrid recommender systems, combining collaborative, content-based, and knowledge-based filtering, have greatly improved personalized recommendations on mental health. Such systems enhance the accuracy and relevance of suggestions by providing personalized needs and leveraging diversified data and algorithms. They assist mental health providers by providing suitable interventions and improving user satisfaction with digital therapy. Since these hybrid systems fill in the gaps in traditional methods, they are scalable, impactful solutions to personalized mental health care that improve treatment adherence and outcomes.

Hybrid systems provide several advantages, such as improved accuracy, more diverse recommendations, and handling of cold-start users and sparse data scenarios. They further enhance scalability by distributing the computational loads and incorporating richer data. Consequently, hybrid systems provide the foundation of personalized recommendation engines, allowing Amazon, Spotify, Netflix, and several other businesses to create enjoyable user experiences.

4) Semantic-Based Filtering Technique

Semantic-based filtering will use the meaning of text content to generate specific filters instead of precise text matches. Instead, for example, of blocking just the exact phrase "buy now," a semantic filter would also be able to find and block variants such as "purchase immediately" or "acquire right away." This method is commonly employed in information retrieval and recommendation systems as it relates to the significance of words, phrases, or ideas. Considering the context and interrelations among concepts, semantic-based filtering is anticipated to yield more precise and contextually relevant results than a mere keyword match. Semantic filtering could be divided further into various types depending on the fundamental methods and underlying applications: ontology-based, knowledge-graph-based, and rule-based, to name a few.

The ontology-based advertisement recommendation system introduces semantic-based filtering for enhanced ad personalization in social networks. Using the same ontological model, user data and advertisement content are shown as semantic vectors, containing significant linkages that extend beyond simple keyword matching. It dynamically updates user profiles based on interactions, avoiding problems such as data sparsity and cold-start issues while ensuring accurate and context-aware ad recommendations. This framework improves the relevance and diversity of recommendations by incorporating NLP tools for semantic annotation and similarity metrics such as cosine similarity, thus achieving very important metrics, such as high precision and F-measure scores in experimental evaluations [17].

Kartheek et al. [18] proposed a knowledge graph-based recommender system that applies the semantic-based filtering by embedding the entities and relationships into lower-dimensional feature space, which really mitigates the sparsity and cold-start issues. The system ensures accurate, explainable recommendations by harnessing scoring functions and optimization processes. It thus allows semantic filtering in the knowledge graph to provide contextually meaningful personalized suggestions that are proven by performance metrics in the case of a movie recommendation scenario.

Salama et al. [19] semantic filtering technique was applied under a rule-based IoT recommendation incorporated with social media, where enhancements in data reasonability and semantic contextual relevance involved semantic web techniques. Both approaches highlight the contribution of semantic-based recommendations toward personalization and adaptability in systems across different domains.

Such types of semantic filtering enable systems to deliver more precise, contextually relevant, and user-centred outcomes, making them suitable for diverse applications such as search engines, recommendation systems, and content moderation.

5) Generative AI Technique

Generative AI is a technology that completely revolutionizes how workflows, ranging from the creative arts, engineering, and research to scientific processes, will be enhanced across all industries and in each of us. Accordingly, generative AI with deep-learning models processes raw data inputs of many different types, including text, images, audio, video, code, and synthesizes new content within these same modes. This is to say, it may convert text into an image, transform an image into a song, or transcribe the video into text, allowing cross-modal content creation in innovative and seamless ways.

Loepp et al. [20] focuses on large language models (LLM) to improve choice-based preference elicitation in recommender systems. The work, therefore, generates textual summaries of item comparisons to enhance user comprehension and experience. Generative AI allows for better interactions by clarifying item sets' semantics, particularly when users are unfamiliar with certain items. It tries to combine traditional collaborative filtering methods with generative AI into an intuitive, user-centred recommendation process.

Deldjoo et al. [21] reviewed how generative AI models address the challenges of data sparsity, cold start, and diversity in recommender systems using GANs and VAEs. It is also demonstrated for generating synthetic training data that improves personalization and quality of suggestions. Efficiency with generative AI compared to traditional methods is presented and significant improvement toward the solution of core limitations like data sparsity is performed. Generative models represent a disruptive tool to work on for increasing user satisfaction and accuracy in a range of systems.

Generative AI techniques, and more importantly, Generative Adversarial Networks (GANs), have contributed significantly to recommender systems by allowing them to overcome serious issues such as data sparsity and noisy data. Unlike traditional systems, depending on user-item interactions modeled as either regression or classification tasks, GAN-based models take a generative approach and create synthetic data resembling real-world distributions. Conditional GANs (cGANs) are a variant that considers additional contextual inputs to enhance the relevance of recommendations. By referring to the recommendation problems as matching problems, the cGAN generates precise conditional rating vectors to improve the accuracy of personalized recommendations. This is very important for complex and sparse data sets to enable recommender systems to generalize better and create meaningful, diverse recommendations Dipak Mahajan et al. [22].

Table 1 shows the summary of the advantages and limitations of each group of recommender systems.

TABLE 1. Comparison of Recommender System Techniques

Advantages	Limitations
Content-Based:	
- Effective for generating personalized recommendations based on item attributes. - Does not require collaboration between users, making it suitable for systems with fewer interactions. - Easily adaptable for various domains using different similarity measures.	- Limited by content exhaustiveness, as recommendations are constrained to known attributes. - Requires detailed and structured data about items to build accurate recommendations.
Collaborative Filtering:	
- Does not require domain-specific knowledge or detailed content information about items. - Helps uncover hidden relationships between items and users, enabling serendipitous recommendations.	- Struggles with cold start issues for new users or items. - Memory-based approaches face scalability issues with growing datasets due to high computational demands.
Hybrid-Based:	
- Can recommend items beyond a user's immediate preferences, fostering discovery. - Automatically adapts to changing user preferences based on new interaction data. - Can generate accurate recommendations based on real user interactions.	- May focus on popular items, leading to reduced recommendation diversity. - Requires a large volume of historical interaction data to perform effectively.
Semantic-Based:	
- Adapts dynamically to evolving user preferences based on updated interaction data. - Generates highly personalized recommendations by analyzing similar user behaviours. - Does not require domain-specific knowledge or item attributes, making it adaptable to various domains.	- Heavily dependent on large volumes of user interaction data to perform effectively. - Experiences the cold-start issue for new users or products without previous interactions. - May lead to popularity bias, over suggesting widely favoured products while disregarding lesser-known alternatives.
Generative-AI::	
- Adapts dynamically to updated interaction data, capturing evolving user preferences. - Doesn't require domain-specific knowledge or item features, making it adaptable across domains.	- Can result in popularity bias, overly recommending well-known items while neglecting less popular ones. - Computational overhead increases for real-time systems, making scalability and responsiveness difficult.

III. MACHINE LEARNING

A. Overview of Machine Learning

ML, a branch of AI, uses data to generate flexible predictions over time. Unlike traditional programming, which requires straightforward code to follow the rules, ML finds patterns or relationships inside data to make conclusions. As the system's predictions and outputs improve with greater datasets, they become more accurate.

Many industries have been transformed and further integrated with ML into user experience and operational efficiency. According to Loukili et al. [23], recommender systems are among the most common applications of e-commerce to improve user experience and support decision-making. These systems analyze the history of purchases, browsing activities, and users' preferences to provide personalized suggestions based on their needs. Similarly, chatbots utilize ML to provide personalized advice, respond to frequently asked questions, and recommend products, improving customer service [23]. In cybersecurity, ML aids in detecting threats by analyzing user behaviour to identify anomalies, thus safeguarding systems and data. [24] classified behaviours as either malicious or normal, several ML algorithms were tested using both anomaly detection and classification methods; the experimental results revealed that the Support Vector Machine (SVM) classification method was superior to the other models. This methodology effectively recognised malicious users within the information system and obtained a predicting success rate of 100%. Another notable application is picture compression, which uses ML methods to minimize file sizes by clustering data into representative centroids while maintaining vital information and optimizing storage. [25] proposes a new medical X-ray image compression system that uses ML. It will consider DCT-based image compression with nine different compression ratios. ML algorithms will be applied to learn the relationship between grey intensities or pixel values representing X-ray images and their optimal compression ratios.

Supervised learning is one form of ML technique that trains an algorithm by using labelled data sets. It could aim at the classification of data or outcome prediction. Algorithms analyze big sets of inputs and their respective outputs, establishing a relationship between the input and its outcome. It is trained through a loss function that computes the inaccuracy of the prediction with the optimization of its parameters using algorithms like gradient descent. Then comes model validation to prevent overfitting and testing on new data for performance. If it is successful, then it is deployed into the application. Supervised learning finds applications in various domains, including spam detection, image recognition, and price prediction, where the goal is to map inputs to specific outputs based on labelled data. It broadly includes classification, which predicts discrete categories or labels, and regression, which predicts continuous values. Common algorithms used in supervised learning include Naive Bayes, Decision Trees, Linear Regression, Random Forest, Logistic Regression, K-Nearest Neighbors, and Support Vector Machines.

Zhu et al. [24] proposed that the model, developed using Logistic Regression, can predict with 90% accuracy whether a patient is at high risk of having a heart attack. Considering that it has been highly accurate and has good performance on test cases, this model will generalize well when new patients run against it. It will also provide valuable insights to healthcare professionals in making early diagnoses and taking necessary preventative care. This capability underlines the potential contribution of the model towards clinically valid decision-making and improved patient outcomes. According to [25], manual detection of fake news is challenging and time-consuming since misinformation is mostly subtle and sophisticated. Naive Bayes, Random Forest, Logistic Regression, and Support Vector Machines (SVM) are some of the different ML techniques used for comparative analysis in handling such difficulty. Each of these methods was compared based on the performance regarding tasks of fake review classification and sentiment analysis. Results of the study identified that, among all methods, SVM was superior in accuracy and reliability in finding the trends that distinguish real from fake reviews. This highlights the possibility SVM has become a strong tool for removing false news and maintaining online information integrity.

Unsupervised learning depends on unlabelled data, hence, the algorithms discover trends and connections. Organising vast amounts of data and finding unknown trends depend especially on this ability. Fundamental methods comprise association, dimensionality reduction, and clustering. Applied in consumer segmentation and genetic data analysis, clustering techniques including K-Means and DBSCAN group data points depending on similarities. Techniques for dimensionality reduction that minimize feature count while preserving the salient patterns are PCA and t-SNE. Using the Apriori algorithm among other association methods, transactional data reveals the linkages between several transactions—things bought together. Other key methods, including Autoencoders, Singular Value Decomposition, and Anomaly Detection, cover extensive unsupervised learning.

According to Iparraguirre-Villanueva et al. [26] comments can damage the reputation of individuals or companies, potentially causing social and economic harm. After using unsupervised learning with K-Means, it was discovered that negative sentiment was the most common, followed by positive sentiment, fear, confidence, and other factors. For this reason, managing user feedback on social networks is important to build social capital. Trogh et al. [27] proposed an unsupervised learning approach to construct and optimize radio maps for indoor localization, trying to maximize accuracy while reducing the need for large-scale data collecting efforts, device calibration, or inertial measurement units. Achieved median localization accuracy of 2.07 meters (28.6% improvement) using WHIPP with only 15 minutes of training data.

Semi-supervised learning is a strategy that combines the powers of supervised and unsupervised learning [28]. It considers a small amount of labelled data with many other unlabelled data. This method works well when obtaining labelled data is expensive or labour-intensive. Semi-supervised learning begins with labelled data training, after which the model uses unlabelled data to improve its understanding of the data distribution using methods such as clustering and pseudo-labelling. Applications include speech recognition, text categorization, and medical imaging. Common techniques include self-training, co-training, graph-based approaches, and teacher-student frameworks that utilize labelled and unlabelled data to enhance model precision.

Most of these classifiers are based on semi-supervised learning, requiring expensive manual labelling by medical specialists for training using labour-intensive tools. On the other hand, semi-supervised learning extracts information from unlabelled samples and uses only a small amount of labelled data. This approach Eckardt et al. [29] effectively bridges the gap between the limited labelled data and the abundance of available data in cancer diagnostics. According to Ramírez-Sanz et al. [30], Fault Detection and Diagnosis (FDD) has turned into a fundamental element for maintaining the effective functioning of intricate industrial processes and machinery. It is among multiple processes that may be automated or semi-automated in industrial environments. This segment continues the study of semi-supervised learning for FDD by highlighting several best practices that have been useful in this field over the last decade. Furthermore, the subject's future possibilities are stressed, as are new research topics that are expected to gain significant interest in the next years.

Reinforcement Learning is a category of ML in which an agent can operate within an environment to optimize a cumulative reward. Unlike supervised or unsupervised learning, reinforcement learning depends on trial and error. An agent acts in various states of the environment, receives rewards or penalties, and refines its policy- a strategy that dictates what to do in each state. The reinforcement learning methods include value-based approaches, such as Q-learning [31], on optimizing the value of actions; policy-based methods like Policy Gradient, which optimize the policy directly, and actor-critic methods, which merge the two to make better decisions. Some very important areas where reinforcement learning can be gainfully employed include robotics, gaming, autonomous vehicles, and dynamic recommendation systems.

Liu et al. [32] reviewed the recent progress in deep reinforcement learning for robotic manipulation, showing how it can handle unstructured environments and optimize tasks by training with rewards. Some of the main challenges are improving sample efficiency, generalization, safety, and scalability for real-world applications. Techniques like imitation learning, Hindsight Experience Replay, and meta-learning help solve these issues and make deep reinforcement learning applicable in warehousing, production, and medical industries.

The different approaches and applications show the huge effect of ML on the industry. Whether through specific suggestions, threat alerts, or self-operating systems, ML continues transforming technology's impact on society. However, as discipline changes and its ability to increase creativity and efficiency improves, it will be more important in designing the technological future.

B. Machine Learning Techniques

ML techniques are a broad family of algorithms and methods that confer capabilities on computers to learn from data for making predictions or decisions. These are some of the fundamental techniques used in teaching a computer to carry out some tasks without actually programming for every step by finding patterns and drawing inferences from the available data. These techniques enable systems to do better in complex tasks with time, improving their precision and effectiveness through experience. This capability is so important for a wide range of industries and applications that it's thus driving autonomous vehicles, facial recognition, personalized recommendations, and predictive analytics.

1) SVD

Singular Value Decomposition (SVD) is a very fundamental factorization in the field of matrices in linear algebra, finding much use in its subfields, including data science, engineering, and other practical mathematics areas. The decomposition is represented as Equation (1):

$$A = U\sum V^T \qquad (1)$$

Where:

- ***U***: an ***m* × *m*** orthogonal matrix containing left singular vectors
- ∑: an ***m* × *n*** diagonal matrix containing singular values in a descending order
- $\boldsymbol{V^T}$: the transpose of an n×n orthogonal matrix containing right singular vectors

In applications such as job recommender systems, to enhance computational efficiency, the matrix ∑ is often truncated to retain only the top singular values. This dimensionality reduction retains the most impactful features, thus enabling the model to make predictions of user-job preferences for unseen job postings by reconstructing the matrix from the reduced U, ∑, and V^T. This will give results in personalized job recommendations, using the latent preference of the user, derived from his or her historical data. In this way, SVD applies to scale up the accuracy and scalability of recommendations that match job seekers with their highly relevant job roles, keeping in consideration their skill set and career aspirations.

Beyond job recommendations, SVD is important in data compression, noise reduction, and feature extraction for ML which helps improve the algorithms for classification and clustering. It underpins recommendation systems and NLP for semantic analysis, crucial in improving search engines and information retrieval systems. By its versatility with complex data, SVD becomes an indispensable tool in modern computational applications that give deep insight into the structure and patterns within the data.

2) TF-IDF

TF-IDF is a statistical measure used in information retrieval and text mining to assess the importance of a word to a document within a collection or corpus. The TF-IDF value increases proportionally with the frequency of a word in a specific document but is offset by the frequency of the word across the entire corpus. This adjustment helps to manage the bias that might occur due to some words appearing more frequently in general.

The formula for TF-IDF includes two components: **Term Frequency (TF)** and **Inverse Document Frequency (IDF)**:

Term Frequency (TF) measures how frequently a term appears in a specific document. This count is normalized to prevent a bias towards longer documents, calculated as in Equation (2):

$$\mathrm{TF}(t,d) = \frac{\text{Number of times } \boldsymbol{t} \text{ appears in document } \boldsymbol{d}}{\text{Total number of terms in documrnt } \boldsymbol{d}} \qquad (2)$$

Where:

- ***t***: A specific term or word whose frequency within a document is being calculated
- ***d***: Represents the document in which the term t appears

Inverse Document Frequency (IDF) gauges the importance of the term across the document set. The more commonly a term appears across documents, the lower its IDF. It is computed as the logarithm of the ratio of the total number of documents to the number of documents containing the term, adjusted to avoid division by zero, as shown in Equation (3):

$$IDF(t,D) = log\frac{N}{1+df} \qquad (3)$$

Where:

- ***D***: Represents the entire corpus or document set that is being considered for analysis
- ***N***: Total number of documents in the corpus D
- ***df***: the number of documents within the corpus D that contains the term t

The overall TF-IDF score is then calculated by multiplying these two figures for each term in each document, which highlights words that are relevant in a document while being rare across the document set, represented as Equation (4):

$$\mathrm{TF-IDF} = \mathrm{TF}(t,d) * \mathrm{IDF}\ (t,D) \qquad (4)$$

These weights are extremely useful in various applications such as search engines, where they enhance the relevance of search results by weighing words more heavily when they are a distinguishing factor in the documents they appear in. TF-IDF also plays a crucial role in document clustering and text classification by allowing algorithms to prioritize words with high document-specific weights, thereby improving classification accuracy.

Essentially, TF-IDF is one of the most significant features in content-based recommender systems, for certain applications like job matching, TF-IDF will analyze and compare textual contents with the goal of retrieving or recommending similar documents, items, or job openings. By extracting the most important words from texts, TF-IDF enhances huge volume text processing over different platforms, including information retrieval systems, search engines, content filters, and job recommendation systems. It inspects the text of resumes and job descriptions, especially the TF-IDF in job recommender systems, for the importance of terms in both job requirements and candidate profiles. This therefore allows employers to match suitable candidates effectively by scoring the importance of each term across individual documents and the whole set of documents.

The application initiates text pre-processing and vectorization, meaning every document, either a candidate's resume or a vacancy description, would be initially represented as a numeric vector regarding the TF-IDF score of each term. Each vector corresponds to unique terms present in the corpus; hence, computational analysis-such as cosine similarity computation for similarity score determination between a candidate's resume and job description vectors is facilitated. This will define how good the closeness of a match is by this cosine similarity score, aiding in ranking the job ads through relevance against candidate skills and experiences. The result is that TF-IDF can improve the relevance of results in search and document classification on any platform, but in a job recommender system, it makes huge boosts in the effectiveness and satisfaction outcomes of job recommender systems; it is, therefore, an invaluable tool for deeper insights and effective data-driven decisions in recruitment.

3) KNN

KNN is a popular non-parametric algorithm in ML that can be used for classification and regression; by far the most common usage, though, has to do with classification problems. Basically, KNN works in the way that a new data point gets classified by the majority vote of its 'k' nearest neighbours, where 'k' is a user-defined constant, and the neighbours are decided based on their distance to the new data point [33]. Usually, this proximity is calculated with Euclidean distance, but Manhattan or Minkowski metrics work just fine too. Because KNN may be computationally intensive to implement because, for every prediction, computation of the distance to every point in the training set is needed, its simplicity and ease of implementation make it one of the most popular choices for many practical applications.

Some steps are required in the implementation of KNN on the job recommender system, which can help find a match between seekers of jobs and suitable postings. First, the description of the vacancy should be changed into feature vectors, as well as the profiles of candidates. Such vectors could represent features like skills, experience level, educational qualifications, and other relevant metrics. These are the vectors, once estimated, to which the KNN algorithm can be applied to determine the 'k' nearest profiles for a given job description, or vice-versa, by the similarity in their feature vectors.

KNN in a job recommender system enables the dynamic and flexible matching process of jobs. For example, KNN will compute for the best 'k' candidates matching with a job, which can be performed by measuring the distances between a job posting vector and the candidates in a system [34]. Merely adding a candidate profile, this system would offer job recommendations that are most fitting for this candidate and allow for the improvement in user experience via timely and very relevant opportunities.

More generally, KNN can operate on many different types of data; this will prove quite useful within a job recommender system. It would function with numerical data, but even more importantly, it also handles categorical data; thus, giving a multidimensional richness toward the recommendation of jobs. Therefore, this kind of flexibility would also allow for different industries, jobs, or different levels of job complexity and candidate expertise within the KNN job recommender system. However, for the best performance and accuracy, the value of 'k' must be carefully chosen. In contrast, data pre-processing steps such as normalization must be considered, especially when features are of a different scale and dimensionality. In that respect, KNN provides a simple way to implement job recommendations, its efficiency and effectiveness may depend significantly on these preparatory steps and parameters.

4) XG-Boost

XGBoost is an enhanced version of the gradient boosting concept, which has been highly utilized in ML due to its efficiency and performance. It extends the basic ideas of boosting by adding models that predict the residuals of the previous models to cumulatively add up to the final prediction. XGBoost adds in the inclusion of a regularization term, which helps to reduce overfitting in models. The XGBoost method contains many system optimizations, ranging from parallel processing and cache optimization to make XGBoost fast and scalable. XGBoost also supports multiple custom objective functions and different metrics for better support on various hard prediction tasks at large.

In the case of a job recommendation system, for instance, the system would entail training a predictive model for a job match with XGBoost, using past information on job postings and successful candidates.

Examples of features can be the categorization of the position, their required and preferable skills, previous job titles, educational background, and all other relevant factors extracted from both job descriptions and candidate profiles. Based on these features, XGBoost will analyze them to identify patterns that lead to successful job placements. These are vital data points that, after proper training in the XGBoost model, could rank job postings effectively for any given candidate based on the predicted probability of successful applications. It may result in customized job recommendations to job seekers that will dramatically raise the match of job seekers and positions. This predictive capacity makes XGBoost a compelling tool to augment the efficiency of job recommendation, optimizing the recruiting process by fitting candidate qualifications into the needs arising within job openings.

IV. EVALUATION METRICS

Evaluation metrics refer to the quantitative measures that normally take place in performance evaluation related to any model or algorithm of ML, data science, or statistical methods. Generally, these provide a sense of how well a model will make predictions classify data, or match patterns on the task it is designed for. It helps stakeholders contextualize strengths and weaknesses in one's models as a roadmap to further improvements or optimizations by informing them of which metric of evaluation may be appropriate again, depending on the scope and goals of a project.

A. Precision

Precision is a critical metric for evaluating the accuracy of classification models, emphasizing the proportion of positive identifications that were accurate. It is calculated with the Equation (5):

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

Where:

- **TP (True Positives):** Represents the instances where the model correctly predicts the positive class
- **FP (False Positives):** Instances where the model incorrectly labels a negative instance as positive

Precision is especially useful in domains like medical diagnosis and spam detection, where the consequences of false positives are grave. Precision is only concerned with the correctness of the positive predictions, which makes it a measure of how well the model identifies relevant instances correctly. It makes precision especially important in cases where the cost of a false positive is high, not only to ensure that the model predictions are frequent but also truly relevant.

B. Recall

Recall, sensitivity, or true positive rate, is an important metric while assessing the different classification models when the model performance focuses on the capture of as many positives as possible. It defines the ratio between actual positives, which are correctly estimated by the model. Thus, it is of special importance when missing a positive instance, a false negative would entail grave consequences, for example, medical diagnostics or fraud detection.

Recall is given by the Equation (6):

$$Precision = \frac{TP}{TP + FN} \tag{6}$$

Where:

- **TP (True Positives)**: Represents the instances where the model correctly predicts the positive class
- **FN (False Negatives):** Instances where the model fails to detect the positive class, erroneously categorizing them as negatives

C. RMSE

The RMSE is a useful metric to quantify the accuracy of a model about regression-type problems, where it must predict against observed values. It is a weighted average, measured by the square root of the mean of the squared differences between predictions and observations, mathematically represented as Equation (7):

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{7}$$

Where:

- $\boldsymbol{n}$: The number of observations
- $\boldsymbol{y_i}$: The actual value for the i-th observation.
- $\boldsymbol{\hat{y}_i}$: The predicted value for the i-th observation.

The RMSE gives the standard deviation of the residuals or prediction errors. It conveys the extent to which the data points fall away from the regression line and, hence, how closely packed the data is around the best-fit line. A high value of RMSE means large magnitudes of errors, implying poor generalization of the model; a low RMSE implies good performance, meaning a closer range between the predictions and the actual values observed. This metric is very informative because it's a measure of the size of error; thus, it helps one estimate how many errors a model typically makes in its predictions.

D. MAE

MAE is the average magnitude of the prediction errors made over a set of predictions for the instances in a regression model. It represents the average value of the absolute of errors in a set of predictions. It is a linear measure of the size of the errors, and it doesn't take the direction of the errors into consideration; hence, all the errors weigh equally. Calculation in the MAE is done by averaging the sum of absolute differences between the predicted values and the values observed, mathematically expressed as Equation (8):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{8}$$

Where:

- $\boldsymbol{n}$: The number of observations

- y_i: The actual value for the i-th observation.
- $\hat{y}_i$: The predicted value for the i-th observation.

Because this calculation is just the simple average of absolute errors from all predictions, the MAE becomes easy to interpret. Thus, a lesser MAE would denote a more accurate model by giving the number where the error that the model is making with a particular value on average will be smaller. Contrarily, higher MAE reflects greater deviation of predicted from actual values and can point toward an area for model improvement. This metric is useful, mainly because of its resistance to outliers, since it does not square the errors; hence, one huge error will not drive the metric astray.

Offline performance measures for model testing before deployment in industrial job suggestion systems are imperative[35]. Precision and recall are typically utilized to determine the relevance and coverage of recommendations, while RMSE and MAE evaluate the size of prediction errors. RMSE has a worse punishment for huge errors, compared to MAE which provides an unbiased average error measurement. Companies like LinkedIn and Indeed employ offline testing with historical interaction data to benchmark algorithms and eliminate candidate models before live testing. These metrics capture strengths and weaknesses under controlled environments while primarily logging predictive accuracy and not user satisfaction or engagement, and caution is necessary before real-world use. Table 2 presents a comparison of different evaluation metrics [36].

TABLE 2. Comparison of Evaluation Metrics

Metrics	Purpose	When to use it
Precision	To measure the correctness of positive predictions.	When false positives are costly
Recall	To measure the completeness of positive predictions.	When missing positives are costly
RMSE	To quantify the average magnitude of the error, emphasizing larger errors.	In regression tasks where large errors must be penalized more heavily.
MAE	To measure the average absolute difference between predictions and true values.	In regression tasks where all errors should be treated equally, regardless of size.

V. FAIRNESS CHALLENGES AND DEBIASING TECHNIQUES IN JOB RECOMMENDATION SYSTEMS

Despite advances in machine learning-based job recommendation systems, fairness challenges remain a significant concern. Historical biases embedded in training datasets can result in unequal exposure to job opportunities across gender, ethnicity, and other demographic groups. These biases risk reinforcing systemic inequalities rather than promoting inclusive hiring practices [37].

Several notable cases highlight the presence of bias in industrial recommendation systems. For example, LinkedIn was found to recommend high-paying leadership roles less frequently to women than to men, due to the influence of historically biased hiring patterns captured in its training data. Similarly, Amazon discontinued the use of its AI recruiting tool after discovering that it downgraded resumes containing words like "women's," as the model had learned from male-dominated historical hiring data. These examples illustrate the potential risks when machine learning models inherit societal biases without explicit correction mechanisms [38],[51], [52].

Debiasing techniques can generally be categorized into three main approaches: pre-processing, in-processing, and post-processing [39], [40]. Pre-processing methods aim to reduce bias before model training by modifying or rebalancing the input data. For instance, by augmenting underrepresented candidate profiles or reweighting features, models can learn from a more balanced dataset. This approach is model agnostic but may inadvertently discard useful information. In-processing techniques introduce fairness constraints during model training, optimizing not only for accuracy but also for equitable treatment across demographic groups. While more integrated and potentially powerful, in-processing methods are complex to design and often model-specific. Post-processing methods adjust model outputs after training to ensure fairness, such as re-ranking job recommendations to balance exposure between groups. Although simpler to implement, post-processing can sometimes degrade the overall quality of recommendations. Table 3 shows the comparison of debiasing techniques in Job Recommendation Systems. Therefore, the choice of debiasing strategy should be informed by the specific goals, data characteristics, and fairness requirements of the job recommendation system under development.

TABLE 3. Comparison of Debiasing Techniques in Job Recommendation Systems

Advantages	Limitations
Pre-processing:	
- Simple to implement - Model-agnostic - No change to model design needed	- May lose useful information - Risk of underfitting - Hard to correct complex biases .
In-processing:	
- Can directly optimize fairness and accuracy - Deep control over fairness trade-offs	- High implementation complexity - Often model-specific - Difficult to generalize

Post-processing:	
- Easy to apply - Works with any pre-trained model - Flexible	- May reduce predictive accuracy - Only treats symptoms - Less principled fairness

VI. RELATED WORK

Appadoo et al. [41] proposed a job recommendation system called JobFit, which estimates the suitability of applicants for jobs by using multiple ML models together with a collaborative filtering recommendation engine. Each applicant will be given a JobFit score based on qualifications, skills, experience, personality, job satisfaction, and retention probabilities. It would be a system that tries to make the recruitment process as easy as possible by supporting human resources personnel in making the best hiring decisions by filtering those candidates who best suit the bill. The different functions are integrated into various ML models, such as qualification matches to check if applicants meet the minimum educational requirements, and skill and experience matches are performed using NLP) and similarity measures. A personality model is one that, based on collaborative filtering, estimates personality-job fit. Satisfaction and retention models, on the other hand, estimate applicant potential of satisfaction and probability in the role. These model outputs are combined using a regression model to compute the final JobFit score. Publicly available datasets such as IBM HR Analytics, the National Longitudinal Survey of Youth 1997, and Similar Skills were supplemented with survey data to power the system. Class balancing and augmentation techniques were done to make preprocessed data more diverse. JobFit has many strong points, including its capability of assembling several models to determine different dimensions of suitability regarding candidates, such as personality and retention, which are overlooked in the traditional approaches. Besides that, cold starts and other problems like data scantiness are tackled with related skills and high-end NLP techniques. However, it relies on structured and comprehensive datasets, which reduces its applicability in a sparse or unstructured data scenario. While it outperforms conventional approaches, further enhancement in the assessment of cognitive abilities and personal strengths would be required for a better fit to the job requirements, though that would need huge time and effort.

Besides, de Ruijt & Bhulai [42] reviewed the literature on JRS between 2011 and 2021, with a special emphasis on relatively neglected aspects: temporal and reciprocal job recommendations and algorithmic fairness. The paper systematizes hybrid recommendation methods, providing the reasons for their distinctiveness and discussing how different availability guides the validation methods. The study underlines that, for the recommender systems, both the recruiter and job seekers' preferences must be approached from the modern light, including advanced techniques such as deep learning. The methodologies involved in JRS include content-based systems, collaborative filtering, and hybrids. Content-based systems use semantic similarities through techniques such as TF-IDF or word embeddings that match candidate profiles with job descriptions. On the other hand, collaborative filtering relies on user-item interactions, and hybrids combine them for better performance. Recently, many studies have extracted high-level features from resumes and job descriptions using deep neural networks. The validation is done using competition datasets, expert reviews, or interaction data to establish the effectiveness of recommendations. Notable datasets include those from the RecSys 2016 and 2017 competitions, which featured data from the job board Xing and the CareerBuilder 2012 dataset from Kaggle. These datasets provide candidate profiles, job postings, and interaction data such as clicks or applications. Precision, recall, accuracy, and F1-score are standard metrics to gauge performance, whereas recent approaches include temporal features for recency of job posts, and reciprocal measures have also been used to balance the applications across vacancies. Hybrid systems and deep learning models increase personalization, besides overcoming cold-start problems and discrepancies in the language used by the job seeker and the recruiter. However, limitations remain, including dataset dependency that raises concerns about model generalizability, insufficient attention to fairness and discrimination in algorithms, and a lack of access to real-world interaction data. Scalability challenges persist, especially for large-scale applications, necessitating further optimization to address these issues comprehensively.

Moreover, Freire & de Castro [43] conducted a systematic review of recommender systems (RS) in the e-recruitment domain, analyzing methodologies, data sources, and evaluation methods. It highlights the challenges of information overload, cold-start issues, and the need for context-aware recommendations. The study identifies a shift toward hybrid approaches and novel techniques that integrate multiple methods, such as content-based, collaborative filtering, and knowledge-based systems, to enhance recommendation quality and address traditional RS limitations. The datasets used are professional social networks, resumes, job posts, and interaction data. Public datasets, such as those from ACM RecSys challenges, are often used. Evaluation metrics include precision, recall, F1 score, and utility measures like click-through rates that assess system relevance and efficiency. Advanced AI and deep learning models include artificial neural networks (ANNs) and long short-term memory (LSTM) networks that are being highlighted for their ability to handle complex recommendation tasks, while hybrid systems combine strengths to improve scalability and cold-start handling. While these systems can provide high performance and scalability, several challenges persist. Addressing insufficient or sparse data, minimizing cold-start problems, and lowering the computational complexity of AI models remain major challenges. Moreover, relevance is a matter of perspective and recruitment procedures are temporary. Furthermore, investigations and innovative approaches are necessary to address concerns like equity and minimizing biases in suggestions.

In 2022, Parida et al. [44] analyzed the job recommender system using ML techniques to match suitable employment opportunities for candidates using their candidate and job analysis descriptions. The system automates the recommendation of employment. Hence, it's much easier than matching possible workers and employers. The framework cleans the data to remove redundancies, extracts key features, and uses a range of ML algorithms, including Logistic Regression, K-Nearest Neighbors, Naïve Bayes, and the Random Forest Classifier (RFC). Of these, RFC performed best, with the highest accuracy. The system uses a variety of data visualization techniques, such as heatmaps and scatter plots, for feature distribution analysis and further optimization of recommendations. Stratified K-fold cross-validation was applied to optimize the classifiers for further improvements in the predictive accuracy. The system will also include a geo-area-based recommendation framework that could enable job seekers to look for job opportunities around themselves using geographical mapping tools and enhance accessibility, ensuring ease of locating employers by the candidates, hence improving the pragmatic viability of the recommendation system. The study used the datasets from LinkedIn and Facebook, which were pre-processed to retain only relevant job-related features. The dataset from LinkedIn contained 39,538 entries and eight features, while the one from Facebook had fewer features but was used to prove the system's effectiveness on different platforms. Metrics for evaluation included accuracy, precision, recall, and F1 score, with RFC giving the maximum optimized results with an accuracy of 99.78% for LinkedIn and 99.03% for Facebook. The Advantages of the system are high prediction accuracy, efficient data processing, and information overload reduction via suggesting jobs that fit users' profiles. However, limitations were noted regarding data sparsity in less visited geographical areas and potential biases in data representation. The authors mentioned further expanding the geo-area framework for more localized and dynamic recommendations and optimising the recommendation algorithms toward better handling diverse datasets.

In addition, Simanjuntak & Wibowo [45] proposed a hybrid job recommendation system that will elevate candidate-job matching. This design combines content-based filtering and collaborative filtering with Named Entity Recognition (NER). Combining these methods allows it to handle the typical challenges of the cold-start problem and sparsity found in traditional recommendation systems. TF-IDF and NER techniques will be applied to extract meaningful features for job descriptions and candidate resumes to train ML models using methods such as Support Vector Machines, and Naive Bayes algorithms. This is because bringing both elements into one allows the best way for the system to tune its capability of analyzing text data and finding patterns while trying to make recommendations. The proposed methodology for hybrid job recommendation is systematic and exhaustive, including punctuation removal, case folding, tokenization, stopword removal, stemming, and vectorization using TF-IDF. NER categorizes names, skills, and locations, enhancing data representation. The ML model will use 70% of the data for training and 30% for testing, while content-based and collaborative filtering will handle the cold-start issue. The dataset used is from Kaggle, containing 24,475 job postings and 200 resumes. The performance evaluation by metrics such as accuracy, precision, recall, and F1-score showed that KNN achieved 71% accuracy with combined data, while SVM excelled at 83.7% accuracy with resumes alone. NER performed well for names and email addresses but struggled with locations and years of experience. The hybrid system improves feature extraction through TF-IDF and NER, enhancing ML model performance while addressing cold-start limitations. The flexibility of multiple classifiers enables optimization, but limitations persist, including a relatively small dataset, poor NER precision for some categories, and scalability concerns with larger datasets. Additionally, novel users or jobs could still challenge the system's adaptability. It therefore sums up that a hybrid system would have tremendous promises for improving job and candidate matching processes, even as the latter area of optimization and scalability needs further enhancement.

Singh et al. [46] proposed a novel employment recommendation system incorporating ML and Deep Learning (DL) to better serve job candidates. This system combines the ability of ML to handle diverse datasets with the strength of DL in discerning intricate patterns aiming to enhance recommendation accuracy and reduce the "cold start" issue. Salient features are extracting relevant information from job descriptions and resumes using TF-IDF, while CNNs will find a pattern in the data for exact recommendations. The model also focuses on fairness to guarantee that suggestions are given equitably across demographics for better job market transparency. This study applies several ML and DL models. Feature extraction applies TF-IDF to represent textual data in numerical format. Logistic regression, decision trees, random forests, and convolutional neural networks were applied to establish an effective match between job positions and candidates. CNNs adopt convolution and pooling layers to examine data patterns. Moreover, ensembles of decision trees used in random forests enhance the dependability of a prediction. It involves pre-processing to clean and normalize the information, and feature engineering to improve the data so that the models built will be more accurate and adaptable to a wide range of user needs. The system proposed here is trained and evaluated on Kaggle, LinkedIn, and the Recruit Challenge 2020 datasets. These datasets have been comprehensive, with job titles, descriptions, required skills, qualifications, and user profiles. The accuracy, precision, recall, and F1 score evaluate the performance of the models in this study. Among the methods, CNN has the best results with an accuracy of 97% and an F1 score of 0.97, proving outstanding among all job recommendations. Other models, such as random forests and logistic regression, have also been developed for their validity. The advantages of this approach are high accuracy in job matching and candidate matching, and the ability to use DL techniques to handle complex datasets that reduce human effort in the recruitment process. It reduces hiring costs while improving the quality of the matches and is fair and inclusive in terms of equitable recommendations. Yet, there are a few limitations in the system. The efficiency of such a system greatly depends on the quality and comprehensiveness of the input data., while a "cold start" problem about new users is not resolved. Furthermore, the computational cost for training deep learning models, such as CNNs, may further limit scalability for real-time applications in a wide context.

Beyond that, Mao et al. [47] proposed a job recommendation model that integrates user attention levels and tensor decomposition for improved recommendation accuracy. The approach takes full advantage of browsing time as an important indicator of user interest, establishing a three-dimensional tensor covering "job seeker-user, position, and attention levels." It is then decomposed by the Bayesian Probability Tensor Factorization (BPTF) method to predict ratings that users would give to each position that they have not rated so far, thus providing a personalized recommendation. This approach involves two steps: it first derives user attention levels from browsing behaviours, categorizes the same using threshold-based segmentation, and constructs a score tensor. These interaction sequences comprise browsing, submitting resumes, and offers that are scored to form this tensor. Second, BPTF predicts job ratings by factoring in user preferences, job attributes, and attention levels. It considers the data from the online recruitment platform "rezhao", which contains 335 job seekers, 2796 job positions, and 6582 ratings, where low-activity users and bots were filtered out. Accuracy metrics predicted rating precision, binary classification task Precision and Recall, and F1-score were used for the task. The proposed model demonstrated various advantages, such as using multisource heterogeneous data, eliminating some cold-start and sparsity problems, and outperforming traditional and hybrid recommendation techniques. Some of the limitations include difficulty distinguishing genuine browsing behaviour from the anomaly and the fact that the population of college students restricts the representativeness, reliance on historical data and sparsity with high granularity segmentations.

Apart from that, Sankarasetty et al. [48] proposed a job recommendation system that leverages ML classification algorithms to match jobseekers with relevant job opportunities based on their skill sets, certifications, and interests. The system also incorporates peer ratings and recommendations to highlight top-rated jobs, simplifying the job search process. It employs five ML algorithms—Logistic Regression, Support Vector Machine, Naive Bayes, Decision Tree, and Random Forest—to determine a jobseeker's eligibility for specific jobs. Logistic Regression, which demonstrated the highest accuracy, forms the basis of the proposed model. The system comprises three modules: an Admin Module for managing user data and job postings, a Job Recruiter Module for adding and managing job applications, and a Job Seeker Module for profile creation, job applications, and rating jobs. The authors used a dynamic dataset, jobdetails.csv, which includes job seeker information such as grades and abilities and job posting details. The best possible outcome was achieved with Logistic Regression, which presented an accuracy of 88.89%, beating the other algorithms tested, including Random Forest, with 83.34%; Decision Tree, 66.67%; Naive Bayes, 66.67%; and Support Vector Machine, 61.12%. The proposed system will find job seeker eligibility using profile attributes to recommend jobs to him/her. The inclusion of peer ratings and recommendations enhances the reliability of the suggestions, while ease of use ensures accessibility in terms of navigation. While it is a good system, it does have its limitations. If users do not actively rate, the system might have biases. The model may depend on skill sets and grades, which may not fully encapsulate other critical factors influencing job fitness. This system does not explore advanced methods, such as deep learning or NLP, which could improve suggestion accuracy. The suggested method effectively enhances job-matching processes, significantly conserving time for both job seekers and recruiters.

Additionally, Huamán et al. [49] proposed a hybrid job recommendation model, inclusive of both collaborative filtering techniques and content-based recommendation techniques to leverage job matching for undergraduate software engineering students. The system tries to overcome difficulties in the search for employment by graduates that match their professional profiles by employing ML to compute similarity percentages for job offers with students' profiles. It involves administering professional orientation tests, which are psychologists-validated, through platforms like Testlify and TestGorilla, to create detailed profiles, including skills, competencies, work interests, and entrepreneurship data. Web scraping also fetches job descriptions and related information from LinkedIn. These datasets were pre-processed, and a hybrid recommendation model built on the Surprise library in Python used similarity scores derived through TF-IDF-like tools or normalized ratings calculation and ranking. Figure 2 shows the Job Recommender flow.

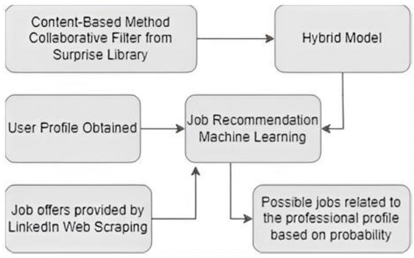

FIGURE 2. Job Recommender flow by Huamán et al. [33].

The data used includes two major ones: student professional profiles categorized by their test results and job description data from LinkedIn. Performance metrics adopted to assess the system included accuracy, relevance, and processing time. This model has achieved a 40% accuracy rate, better than several alternatives from 10 to 22%, with faster output of recommendations at 16.23 seconds, much quicker than Kumar et al. at 20.87 seconds. The key advantages of this model include higher recommendation accuracy and diversity, personal recommendations at similarity percentages, scientifically consolidated profiling. It easily fetches job details from LinkedIn and does not require the user to create an account.

However, the model has certain limitations, such as the scarcity of matching job offers, user confusion due to technical job descriptions, and the temporal availability of job postings. Though it outperforms some models in terms of accuracy, further refinement is needed to come up with results that better match user profiles. Despite difficulties, it is encouraging to integrate psychology and technology to assist software engineering students in securing suitable employment, thereby enhancing their professional and emotional well-being.

Next, K et al. [50] proposes a job recommendation system for LinkedIn user profiles that adopts a hybrid approach between content-based filtering and collaborative filtering in recommending jobs based on personal preferences. The system considers the users' attributes, including skills, work experiences, and preferences, to match them to relevant job opportunities. While collaborative filtering relies on the matrix of user-job ratings and parameter vectors to predict user preferences, content-based filtering does so by an attribute-based analysis of job descriptions and skills, thus providing feature vectors. The neural network extends this content-based approach even further by processing more information for users and jobs, generating more precise recommendations. The dataset used for evaluation in this work is from LinkedIn; it contains an enormous variety of positions and a wide range of user preferences. It contains elaborate user profiles, preferences, and features engineered for average user ratings and job content descriptors. By doing so, these features complete the process of training and testing and allow extensive experimentation with various recommendation approaches. These would include the proportion of relevant recommendations, referred to as precision; recall, referring to completeness; the F1 score, which provides a balanced measure for both precision and recall; and training time, regarding computational efficiency. The system has several advantages, such as personalized and adaptive recommendations that update when user profiles change, efficient real-time application by using dimensionality reduction techniques, and improved job matching. However, it also has some limitations: the overspecialization problem, or the filter bubble, reliance on complete and accurate user profiles, and the hybrid model may produce biased results. Additional strategies must be included to address these difficulties and deliver diverse and equitable recommendations without compromising the system's efficacy.

VII. DISCUSSION AND WAY FORWARD

This section gives the related works described above. Table 4 provides an overview of these studies, their most important findings, the datasets employed, and the measures used to test each system. The review indicates that hybrid-based recommender systems hold immense potential for optimum performance by compensating for one method's deficiency with the strengths of another.

The reviews highlight the rapid advancement and complexity of job recommendation systems driven by recent advances in ML, deep learning, and NLP. By systematically comparison among traditional methods—content-based and collaborative filtering—and other advanced hybrid methods, the paper demonstrates that using multiple algorithms enhances candidate-job matching accuracy. Techniques like TF-IDF feature extraction and semantic filtering have also enhanced job posting analysis and resume analysis, while ML techniques like SVD, KNN, and XGBoost have helped to unveil latent patterns from high-dimensional data. Deep models like CNNs and GANs also help solve problems like noisy data and cold-start problems. With these innovations, though, come also certain significant challenges, like dependence on good-quality, structured data, filter bubbles due to overspecialization, computational expense, and risk of algorithmic bias.

Despite significant advancements in hybrid-based job recommendation systems, several challenges remain unresolved. Hybrid models, while effective in mitigating cold-start and sparsity issues, can still amplify underlying biases when trained on imbalanced datasets, leading to fairness concerns. Moreover, hybrid approaches often struggle with balancing recommendation diversity against precision, risking overspecialization or filter bubbles. Scalability also remains a concern when integrating multiple algorithms, especially with large, real-time labor market data. Furthermore, the dynamic nature of user preferences and job requirements demands continual model retraining and adaptation, which current hybrid systems are not fully optimized to handle. Addressing these limitations is crucial for the next generation of job recommendation systems that are not only accurate but also fair, dynamic, and scalable.

In addition to technical issues such as sparsity and scalability, bias and fairness are an important issue in the development of job recommendation systems. Even newer hybrid models are prone to propagation and amplification of existing biases in training data, leading to disproportionate exposure to job opportunities among different demographic groups. For example, algorithms that primarily learn from historical patterns of employment will unintentionally discriminate against majority groups and underrepresent minority applicants, women, or lower-economic-status applicants. These biases not only erode the moral credibility of recommender systems but also have the potential to replicate systemic bias in hiring. Also, standard measures like precision and recall hardly ever incorporate fairness dimensions, necessitating the application of fairness-sensitive measures and debiasing during training as well as testing.

TABLE 4. Summary of prior related research works.

References & Titles	Findings & Datasets	Evaluation Metrics
Appadoo et al., 2020 [41] JobFit: Job Recommendation using Machine Learning and Recommendation Engine	Findings: Proposed "JobFit," a multi-model JRS combining ML techniques with collaborative filtering; integrates models for qualifications, skills, experience, personality, satisfaction, and retention to compute a JobFit score. Tackles cold-start issues with high-end NLP techniques. Challenges include dependence on structured data. Datasets: IBM HR Analytics, National Longitudinal Survey of Youth 1997 Source: https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset	Regression-based JobFit score, accuracy, and performance. The specific numeric results are not given, but the authors state that JobFit surpasses conventional techniques by being able to merge various models into a single score. They state better candidate–job matching and reduced manual screening. Other enhancements needed for unstructured data scenarios.
de Ruijt & Bhulai, 2021 [42] Job Recommender Systems: A Review	Findings: Reviewed JRS literature (2011–2021), highlighting hybrid systems with deep learning and the inclusion of temporal and reciprocal recommendations. Discussed fairness in algorithms. Identified the shift toward advanced personalization methods and challenges with generalizability and real-world scalability. Datasets: RecSys 2016 & 2017 (Xing), CareerBuilder 2012 (Kaggle) Source: https://www.kaggle.com/c/job-recommendation	Covers basic metrics in the context of Precision, Recall, Accuracy, F1-score, and time-based measures. The paper is not giving raw numerical performance figures but referring to hybrid systems and deep learning leading to better performance, personalization, and cold-start control. It also stresses fairness and the issue of large-scale real-world data.
Freire & de Castro, 2021 [43] e-Recruitment recommender systems: a systematic review	Findings: Systematic review of RS in e-recruitment; identified the shift toward hybrid approaches integrating content-based, collaborative filtering, and AI techniques. Challenges include handling incomplete data, biases, and the transient nature of recruitment processes. Datasets: Public datasets from ACM RecSys 2016 challenges, resumes, job postings, and interaction data. Sources: Not Available	Uses Precision, Recall, F1, and utility metrics. The authors note that many studies have high performance but have issues with data sparsity, fairness, and scalability. No single number outcome is generated, being a review, but they conclude that advanced or hybrid AI methods can significantly boost match quality if data are sufficient and appropriately structured.
Parida et al., 2022 [44] Prediction of recommendations for employment utilizing machine learning procedures and geo-area based recommender framework	Findings: Created a system matching candidates and jobs using various ML models, with Random Forest Classifier achieving the best results. Includes geo-area-based recommendations for proximity-based job matching. Datasets: LinkedIn dataset (39,538 entries with 8 features), Facebook dataset with fewer features but used for cross-platform validation. Sources: Not Available	Accuracy, Precision, Recall, F1-score applied. RFC gained 99.78% on LinkedIn and 99.03% on Facebook—best results among classifiers used. High accuracy and effective processing of data, though with reported limitations in poorly populated areas and possible data prejudices, are underscored by the authors.

Simanjuntak & Wibowo, 2023 [45] Recommendation System for Online Job Vacancy Using Machine Learning	Findings: Proposed a hybrid job recommendation system that integrates content-based and collaborative filtering with NER. It enhances feature extraction through TF-IDF and NER, addressing cold-start and sparsity challenges. Datasets: Kaggle dataset with 24,475 job postings and 200 resumes. Sources: Not Available	Accuracy, Precision, Recall, F1-score. SVM was 83.7% accurate on resume-only data, and KNN was 71% accurate on combined data. NER was good for names/emailed but was weak on locations and experience years. The hybrid system in general performed better than a single-method model, though the size of the dataset and the NER limitations are still issues.
Singh et al., 2023 [46] Method for Job Recommendation based on Machine Learning and Deep Learning Model	Findings: Developed a system combining ML and DL (e.g., CNNs) for job recommendations. It handles complex patterns, focuses on fairness, and reduces human effort in recruitment while offering high accuracy and adaptability. Datasets: Kaggle, LinkedIn, and Recruit Challenge 2020 datasets with job titles, descriptions, and profiles. Sources: Not Available	Accuracy, Precision, Recall, F1-score. CNN performed best with 97% accuracy and F1=0.97, outperforming other ML models. The paper also indicates reduced human effort in recruitment and improved candidate–job matching. Deep learning, however, has a computational costs and continues to suffer from real new (cold start) users or unstructured data.
Mao et al., 2023 [47] A Job Recommendation Method Based on Attention Layer Scoring Characteristics and Tensor Decomposition	Findings: Introduced a job recommendation system using user attention levels and Bayesian Probability Tensor Factorization for personalized recommendations. It resolves cold-start issues and sparsity using multisource heterogeneous data. Datasets: Rezhao online recruitment platform data with 335 job seekers, 2796 jobs, and 6582 ratings. Sources: Not Available	Uses rating prediction accuracy, Precision, Recall, and F1-score. The new method beats traditional/hybrid baselines by utilizing attention signals. Numerical improvements are not elaborated, but improvements to rating accuracy and classification metric are reported. The study is limited by potential noise in browsing data and a bounded user population (primarily college students).
Sankarasetty et al., 2023 [48] A Comparative Study on Job Recommendation using Classification Algorithms	Findings: Created a system using ML classifiers (e.g., Logistic Regression, Random Forest) for job recommendations. It includes peer ratings and simplifies job searches with personalized and reliable suggestions. Datasets: Dynamic dataset (jobdetails.csv) containing jobseeker attributes and job postings. Sources: Not Available	Accuracy is the most important measure. Logistic Regression leads the pack with 88.89%, trailed by RF (83.34%), DT (66.67%), NB (66.67%), and SVM (61.12%). The study highlights the mechanism through which peer ratings increase dependability of recommendation, although system performance depends on user involvement in rating.
Huamán et al., 2024 [49] Hybrid job recommendation model based on professional profile using data from job boards and Machine Learning libraries	Finding: Hybrid JRS for software engineering students using collaborative filtering and content-based methods; incorporates professional orientation tests and web scraping to build profiles and match jobs. Achieved better accuracy and faster output than alternatives. Limitations include scarcity of matching jobs and user confusion with technical terms. Datasets: Not Available	Achieved 40% accuracy, superior to prior approaches at 10–22%. Time taken to process was 16.23s, quicker than 20.87s in a study. Although 40% accuracy may seem low, it is an increase in a specialist environment. The approach still suffers from limitations in job availability, user misunderstanding through jargon, and the dynamic nature of adverts.
K et al., 2024 [50] Job Recommendation System using LinkedIn User	Findings: Proposed a LinkedIn-based hybrid system combining content-based filtering, collaborative filtering, and neural networks to provide adaptive and personalized job recommendations. Datasets: LinkedIn dataset with user profiles, preferences, ratings, and job content descriptors. Sources: Not Available	Employs Precision, Recall, F1-score, and training time. No values are provided, but the authors note improved recommendation relevance for real-time use, with a focus on agility regarding user profile updates. Negative features noted are the potential "filter bubbles" and reliance on total, exact user information. Further steps are suggested to introduce diversity and fairness.

VIII. CONCLUSION AND FUTURE WORK

In this review, evidences the massive transformation job suggestion systems have gained through developments in ML, deep learning, and NLP. The traditional algorithms like content-based filtering and collaborative filtering, effective to some extent, are less adept when problems regarding data sparseness, cold-start difficulty, and over-specialization need to be resolved. Hybrid-based recommender systems, which combine the strengths of multiple methodologies, have been viewed as a potential solution that can eliminate these limitations by fusing multiple data sources, context data, and user feedback. This integrative strategy not only enhances the accuracy and personalization of the recommendations but also the efficiency of the overall recruitment process.

Future research must involve developing more adaptive hybrid models that dynamically adapt to real-time data quality and evolving user needs. Bias-aware algorithms need to be developed urgently to achieve fairness and explainability in candidate-job matching and minimize the computational cost to enable scalability in real-world large-scale deployments. In addition, future work would be aided by more comprehensive evaluation metrics that not only quantify accuracy but also quantify user satisfaction and diversity of recommendations. Improving these areas will pave the way for more robust, inclusive, and effective job recommendation systems that better serve employers and job seekers alike.

Furthermore, research needs to focus on enriching hybrid-based methods through adaptive system development to dynamically optimize hybrids of models based on real-time user activity and changing conditions of the labor market. Mitigation strategies of bias used in hybrid approaches will be important for ensuring increased fairness and diversity of recommendations. With advances in recommendation systems, the incorporation of Large Language Models such as BERT and GPT has encouraging potential to create intelligent career guides. Such types of systems would move beyond simple job matching by being able to provide personalized career guidance, skill training paths, and training recommendations according to individual needs. In the longer term, privacy-preserving mechanisms, i.e., federated learning, will become increasingly important. Federated learning makes it possible to train models to be decentralized and, thus, user data can be kept securely locally on devices and still receive benefits from collective learning. The roadmap envisions a future where job recommendation systems not only are accurate and personalized but also ethical, fair, adaptive, and user-privacy-aware.

ACKNOWLEDGEMENT

We thank the reviewers and editor of the journal for their guidance in improving the quality of our article.

FUNDING STATEMENT

There are no funding agencies supporting the research work

AUTHOR CONTRIBUTIONS

Rui-Ern Yap: Conceptualization, Data Curation, Methodology, Validation, Writing – Original Draft Preparation;

Su-Cheng Haw: Project Administration, Supervision, Writing – Review & Editing;

Shaymaa Al-Juboori: Writing - Review & Editing.

CONFLICT OF INTERESTS

No conflict of interests were disclosed.

ETHICS STATEMENTS

This research did not involve human participants, animal subjects, or sensitive personal data, and therefore did not require ethical approval.

REFERENCES

[1] X. Gu, L. Jian, C. Rao, Z. Bu and X. Cheng, "Together is better: knowledge-aware model with resume fusion for online job recommendation," *ACM Transactions on Knowledge Discovery from Data, vol. 19, no. 3, pp. 1-15, 2025.* DOI: https://doi.org/10.1145/3716503

[2] T. Anwar, V. Uma, M.I. Hussain and M. Pantula, "Collaborative filtering and kNN based recommendation to overcome cold start and sparsity issues: A comparative analysis," *Multimedia Tools and Applications, vol. 81, no. 25, pp. 35693-35711, 2022.* DOI: https://doi.org/10.1007/s11042-021-11883-z

[3] 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)*, pp. 157–162, 2024. DOI: https://doi.org/10.1109/ESIC60604.2024.10481635

[4] D. Shin, "How do users interact with algorithm recommender systems? The interaction of users, algorithms, and performance," *Computers in Human Behavior, vol. 109, pp. 106344, 2020.* DOI: https://doi.org/10.1016/j.chb.2020.106344

[5] Z. Fayyaz, M. Ebrahimian, D. Nawara, A. Ibrahim and R. Kashef, "Recommendation systems: Algorithms, challenges, metrics, and business opportunities," *Applied Sciences (Switzerland)*, vol. 10, no. 21, 2020. DOI: https://doi.org/10.3390/app10217748

[6] G.M. Dhananjaya, R.H. Goudar, A.A. Kulkarni, V.N. Rathod and G.S. Hukkeri, "A Digital Recommendation System for Personalized Learning to Enhance Online Education: A Review," *IEEE Access*, vol. 12, 2024. DOI: https://doi.org/10.1109/ACCESS.2024.3369901

[7] Y.M. Latha and B.S. Rao, "Product recommendation using enhanced convolutional neural network for e-commerce platform," *Cluster Computing,* vol. 27, no. 2, pp. 1639-1653, 2024. DOI: https://doi.org/10.1007/s10586-023-04053-3

[8] A. Gugnani and H. Misra, "Implicit skills extraction using document embedding and its use in job recommendation," *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, 2020. DOI: https://doi.org/10.1609/aaai.v34i08.7038

[9] D. Mhamdi, R. Moulouki, M.Y. El Ghoumari, M. Azzouazi and L. Moussaid, "Job recommendation based on job profile clustering and job seeker behavior," *Procedia Computer Science*, 2020. DOI: https://doi.org/10.1016/j.procs.2020.07.102

[10] Y. Mao, Y. Cheng and C. Shi, "A Job Recommendation Method Based on Attention Layer Scoring Characteristics and

Tensor Decomposition," *Applied Sciences*, vol. 13, no. 16, p. 9464, 2023.
DOI: https://doi.org/10.3390/app13169464
[11] E.S. Negara, Sulaiman, R. Andryani, P.H. Saksono, and Y. Widyanti, "Recommendation System with Content-Based Filtering in NFT Marketplace," *Journal of Advances in Information Technology*, vol. 14, no. 3, 2023.
DOI: https://doi.org/10.12720/jait.14.3.518-522
[12] S.M. Sakti, A.D. Laksito, B.W. Sari and D. Prabowo, "Music Recommendation System Using Content-based Filtering Method with Euclidean Distance Algorithm," *Proceeding - 6th International Conference on Information Technology, Information Systems and Electrical Engineering: Applying Data Sciences and Artificial Intelligence Technologies for Environmental Sustainability, ICITISEE 2022*, 2022.
DOI: https://doi.org/10.1109/ICITISEE57756.2022.10057753
[13] R. Lara-Cabrera, Á. González-Prieto, F. Ortega and J. Bobadilla, "Evolving Matrix-Factorization-Based Collaborative Filtering Using Genetic Programming," *Applied Sciences*, vol. 10, no. 2, pp. 675, 2020.
DOI: https://doi.org/10.3390/app10020675
[14] L. Chen, Y. Yuan, J. Yang and A. Zahir, "Improving the prediction quality in memory-based collaborative filtering using categorical features," *Electronics*, vol. 10, no. 2, 2021.
DOI: https://doi.org/10.3390/electronics10020214
[15] F. Horasan, A.H. Yurttakal and S. Gündüz, "A novel model based collaborative filtering recommender system via truncated ULV decomposition," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 8, 2023
DOI: https://doi.org/10.1016/j.jksuci.2023.101724
[16] I. Mazlan, N. Abdullah and N. Ahmad, "Exploring the Impact of Hybrid Recommender Systems on Personalized Mental Health Recommendations," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 6, 2023.
DOI: https://doi.org/10.14569/IJACSA.2023.0140699
[17] F. García-Sánchez, R. Colomo-Palacios and R. Valencia-García, "A social-semantic recommender system for advertisements," *Information Processing & Management, vol. 57, no. 2, pp. 102153, 2020.*
DOI: https://doi.org/10.1016/j.ipm.2019.102153
[18] M. Kartheek and G.P. Sajeev, "Building Semantic Based Recommender System Using Knowledge Graph Embedding," *Proceedings of the IEEE International Conference Image Information Processing*, 2021.
DOI: https://doi.org/10.1109/ICIIP53038.2021.9702632
[19] A. Salama, M.E. Shaheen and H. Alfeel, "Rule-based Recommendation System based on Semantic Web of Things," *International Journal of Engineering Research and Technology*, vol. 13, no. 6, 2020.
DOI: https://doi.org/10.37624/ijert/13.6.2020.1455-1465
[20] B. Loepp and J. Ziegler, "Exploring the Potential of Generative AI for Augmenting Choice-Based Preference Elicitation in Recommender Systems," *Adjunct Proceedings of the 32nd ACM Conference on User Modeling, Adaptation and Personalization*, pp. 114–119, 2024.
DOI: https://doi.org/10.1145/3631700.3664873
[21] Y. Deldjoo, Z. He, J. McAuley, A. Korikov, S. Sanner, A. Ramisa, R. Vidal, M. Sathiamoorthy, A. Kasirzadeh and S. Milano, "A Review of Modern Recommender Systems Using Generative Models (Gen-RecSys)," *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6448-6458, 2024.
DOI: https://doi.org/10.1145/3637528.3671474
[22] A.D. Mahajan, A. Mahale, A.S. Deshmukh, A. Vidyadharan, V.S. Hegde and K. Vijayaraghavan, "Generative AI-Powered Spark Cluster Recommendation Engine," *Proceedings of the 2023 2nd International Conference on Augmented Intelligence and Sustainable Systems,* 2023.
DOI: https://doi.org/10.1109/ICAISS58487.2023.10250700
[23] M. Loukili, F. Messaoudi, and M. El Ghazi, "Machine learning based recommender system for e-commerce," *IAES International Journal of Artificial Intelligence*, vol. 12, no. 4, 2023.
DOI: https://doi.org/10.11591/ijai.v12.i4.pp1803-1811
[24] L. Zhu and G. Goldsztein, "Predicting the Chance of Heart Attack with a Machine Learning Approach – Supervised Learning," *Journal of Student Research*, vol. 11, no. 3, 2022.
DOI: https://doi.org/10.47611/jsrhs.v11i3.3380
[25] M. Tabany and M. Gueffal, "Sentiment Analysis and Fake Amazon Reviews Classification Using SVM Supervised Machine Learning Model," *Journal of Advances in Information Technology*, vol. 15, no. 1, 2024.
DOI: https://doi.org/10.12720/jait.15.1.49-58
[26] O. Iparraguirre-Villanueva, V. Guevara-Ponce, F. Sierra-Liñan, S. Beltozar-Clemente and M. Cabanillas-Carbonell, "Sentiment Analysis of Tweets using Unsupervised Learning Techniques and the K-Means Algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 6, 2022.
DOI: https://doi.org/10.14569/IJACSA.2022.0130669
[27] J. Trogh, W. Joseph, L. Martens and D. Plets, "An unsupervised learning technique to optimize radio maps for indoor localization," *Sensors*, vol. 19, no. 4, 2019.
DOI: https://doi.org/10.3390/s19040752
[28] X. Yang, Z. Song, I. King and Z. Xu, "A Survey on Deep Semi-Supervised Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 9, pp. 8934-8954, 2023.
DOI: https://doi.org/10.1109/TKDE.2022.3220219
[29] J. Eckardt, M. Bornhäuser, K. Wendt and J.M. Middeke, "Semi-supervised learning in cancer diagnostics," *Frontiers in Oncology*, vol. 12, 2022.
DOI: https://doi.org/10.3389/fonc.2022.960984
[30] J.M. Ramírez-Sanz, J. Maestro-Prieto, Á. Arnaiz-González and A. Bustillo, "Semi-supervised learning for industrial fault detection and diagnosis: A systemic review," *ISA Transactions*, vol. 143, pp. 255-270, 2023.
DOI: https://doi.org/10.1016/j.isatra.2023.09.027
[31] X. Chen, L. Yao, J. McAuley, G. Zhou and X. Wang, "Deep reinforcement learning in recommender systems: A survey and new perspectives," *Knowledge-Based Systems*, vol. 264, pp. 110335, 2023.
DOI: https://doi.org/10.1016/j.knosys.2023.110335
[32] R. Liu, F. Nageotte, P. Zanne, M. de Mathelin, and B. Dresp-Langley, "Deep reinforcement learning for the control of robotic manipulation: A focussed mini-review," *Robotics*, vol. 10, no. 1, 2021.
DOI: https://doi.org/10.3390/robotics10010022
[33] R.K. Halder, M.N. Uddin, M.A. Uddin, S. Aryal and A. Khraisat, "Enhancing K-nearest neighbor algorithm: a comprehensive review and performance analysis of modifications," *Journal of Big Data*, vol. 11, no. 1, 2024.
DOI: https://doi.org/10.1186/s40537-024-00973-y
[34] B. S, P. Majumdar and D. Hemanth, "Career Compass: Predictive Analytics and Personalized Job Recommendation System," *SSRN Electronic Journal*, 2025.
DOI: https://doi.org/10.2139/ssrn.5086025
[35] D. Jannach, A. Manzoor, W. Cai and L. Chen, "A Survey on Conversational Recommender Systems," *ACM Computing Surveys*, vol. 54, no. 5, pp. 1-36, 2022.
DOI: https://doi.org/10.1145/3453154
[36] P. Castells and A. Moffat, "Offline recommender system evaluation: Challenges and new directions," *AI Magazine*, vol. 43, no. 2, pp. 225-238, 2022.
DOI: https://doi.org/10.1002/aaai.12051
[37] D. Jin, L. Wang, H. Zhang, Y. Zheng, W. Ding, F. Xia and S. Pan, "A survey on fairness-aware recommender systems," *Information Fusion*, vol. 100, pp. 101906, 2023.
DOI: https://doi.org/10.1016/j.inffus.2023.101906
[38] I.D. Raji and J. Buolamwini, "Actionable Auditing Revisited," *Communications of the ACM*, vol. 66, no. 1, pp. 101-108, 2023.
DOI: https://doi.org/10.1145/3571151
[39] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman and A. Galstyan, "A Survey on Bias and Fairness in Machine Learning," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1-35, 2022.
DOI: https://doi.org/10.1145/3457607
[40] Y. Zhao, Y. Wang, Y. Liu, X. Cheng, C.C. Aggarwal and T. Derr, "Fairness and Diversity in Recommender Systems: A Survey," *ACM Transactions on Intelligent Systems and Technology*, vol. 16, no. 1, pp. 1-28, 2025.
DOI: https://doi.org/10.1145/3664928
[41] K. Appadoo, M. B. Soonnoo, and Z. Mungloo-Dilmohamud, "JobFit: Job Recommendation using Machine Learning and Recommendation Engine," *2020 IEEE Asia-Pacific*

Conference on Computer Science and Data Engineering (CSDE), 2020.
DOI: https://doi.org/10.1109/CSDE50874.2020.9411584

[42] C.D. Ruijt and S. Bhulai, "Job Recommender Systems: A Review," *arXiv*, 2021.
DOI: https://doi.org/10.48550/arXiv.2111.13576

[43] M.N. Freire and L.N.d. Castro, "e-Recruitment recommender systems: a systematic review," *Knowledge and Information Systems*, vol. 63, no. 1, pp. 1-20, 2021.
DOI: https://doi.org/10.1007/s10115-020-01522-8

[44] B. Parida, P. KumarPatra, and S. Mohanty, "Prediction of recommendations for employment utilizing machine learning procedures and geo-area based recommender framework," *Sustainable Operations and Computers*, vol. 3, pp. 83–92, 2022.
DOI: https://doi.org/10.1016/j.susoc.2021.11.001

[45] Y.F. Simanjuntak and A. Wibowo, "Recommendation System for Online Job Vacancy Using Machine Learning Models," *Journal of Theoretical Applied Information Technology*, vol. 101, no. 2, 2023.
URL: https://www.jatit.org/volumes/Vol101No2/28Vol101No2.pdf (Accessed 5 June, 2025)

[46] D. Singh, N. Patel and U. Singh, "Method for Job Recommendation based on Machine Learning and Deep Learning Model," *2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS)*, pp. 875–883, 2023.
DOI: https://doi.org/10.1109/ICACRS58579.2023.10405321

[47] Y. Mao, Y. Cheng and C. Shi, "A Job Recommendation Method Based on Attention Layer Scoring Characteristics and Tensor Decomposition," *Applied Sciences*, vol. 13, no. 16, p. 9464, 2023.
DOI: https://doi.org/10.3390/app13169464

[48] S. Sankarasetty, S. Kakaraparti, M. Saiful Haq, and S. K. Jonnalagadda, "A Comparative Study on Job Recommendation using Classification Algorithms," *Proceedings - 7th International Conference on Computing Methodologies and Communication,* 2023.
DOI: https://doi.org/10.1109/ICCMC56507.2023.10083728

[49] A. Huamán, G. Rebaza and D. Subauste, "Hybrid Job Recommendation Model Based on Professional Profile Using Data from Job Boards and Machine Learning Libraries," *Proceedings of the International Multi-Conference on Society, Cybernetics and Informatics*, pp. 72-79, 2024.
DOI: https://doi.org/10.54808/IMSCI2024.01.72

[50] G. K, M.R. K and A. Haldar, "Job Recommendation System using LinkedIn User Profiles," *2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE)*, pp. 1-6, 2024.
DOI: https://doi.org/10.1109/ic-ETITE58242.2024.10493325

[51] J.W.T. Lim, C. Tee and M.K.O. Goh, "Exploring Activities of Daily Living Among the Elderly through Machine Learning Techniques," *International Journal on Robotics, Automation and Sciences*, vol. 7, no. 1, pp. 35-46, 2025.
DOI: https://doi.org/10.33093/ijoras.2025.7.1.5

[52] J.P. Cheng and S.C. Haw, "Mental Health Problems Prediction Using Machine Learning Techniques," *International Journal on Robotics, Automation and Sciences*, vol. 5, no. 2, pp. 59-72, 2023.
DOI: https://doi.org/10.33093/ijoras.2023.5.2.7