EXPLAINABLE ARTIFICIAL INTELEGIANT IN JOB RECOMMENDATION SYSTEM BASED ON OUTCOME-BASED EDUCATION USING EXPLAINABLE BOOSTING MACHINE

Jalu Muhammad Abror[1,*], Birru Muqdamien[2], Satrio Adi Priyambada[3], Siti Kania Kushadiani[4], Alfiyansyah Arrizqy Hidayat[5]

[1)] Department of Informatics, Sultan Maulana Hasanuddin State Islamic University of Banten, Serang 42171 Indonesia
[2)] Department of Informatics, Sultan Maulana Hasanuddin State Islamic University of Banten, Serang 42171 Indonesia
[3)] Department of National Research and Innovation Agency (BRIN), Bandung 40135 Indonesia
[4)] Department of National Research and Innovation Agency (BRIN), Bandung 40135 Indonesia
[5)] Department of information System, Telkom University, Surabaya 60231 Indonesia

ARTICLE INFO

Keywords:
Explainable AI, EBM, OBE Gap analysis, Job Recommendation System

** Corresponding author:*
jaluabrorr@gmail.com

Article history:
Received: xxx, xxx
Revised: xxx, xxx
Accepted: xxx, xxx
Available online: xxx, xxx

ABSTRACT

Job recommendation systems have been widely implemented in educational decision-making, but many still lack transparency and are not well-aligned with formal learning. This study presents an OBE-based job recommendation system that uses CLO as the primary measure of student competencies. This system consists of a two-component framework: a ranking component that ranks job openings using a composite score calculated based on similarity and the number of job keywords, and an explanation component in which EBM reconstructs the composite score in additive form and identifies the contribution of each feature, thereby providing transparent global and local explanations. This system was evaluated using curriculum data from 48 courses in the Information Systems program and 3,352 job vacancy records from JobStreet. EBM achieved the highest fidelity with R^2 = 0.9984 and RMSE = 0.0014, demonstrating its ability to generate a predefined evaluation function rather than independent predictive accuracy. An analysis of the curriculum-to-industry gap across 36 IT courses yielded an average job skill coverage score of 11.93% for database-related skills such as PostgreSQL, SQL, and MySQL. These findings provide informative evidence at the course level that supports the OBE curriculum, while the transparent breakdown of scores offers insights into ranking results and competency-matching analysis.

1. Introduction

The development of artificial intelligence in recent years has become a hot topic of discussion; with this boom, AI has made significant strides in the field of technology. One example is job recommendation systems, which utilize various predictive models such as neural networks and gradient boosting and are now capable of processing large-scale data to provide highly accurate job recommendations. However, despite these predictive results, some models remain "black boxes," meaning they lack transparency and cannot be fully trusted, which will eventually make it difficult for users to understand the basis for their recommendations [1]. Arguing that "although widely adopted, machine learning models remain largely a black box," while [2] stated "that an explainable recommendation system must provide intuitive explanations for why a particular assignment is recommended." A lack of transparency in this system not only undermines the trust of students and faculty but also limits the ability of educational institutions to conduct evidence-based curriculum evaluations.

Generally, higher education institutions in Indonesia are required to implement an Outcome-Based Education (OBE) approach in accordance with Regulation of the Minister of Education, Culture, Research, and Technology No. 53 of 2023. This regulation shifts curriculum evaluation from Program Learning Outcomes (PLOs) to Course Learning Outcomes (CLOs) at the individual course level as the primary measure of learning success throughout the course of study. However, the use of CLOs as the primary input for job recommendation systems has received limited attention in existing studies, and the mapping of competencies to industry needs is still conducted manually or relies on simplified approaches.

The need for a transparent and explainable CLO-based job recommendation system seems increasingly urgent

DOI: https://doi.org/10.12962/j24068535.v23i1.aXXXX

in this era of digital transformation. Approaches such as Explainable Artificial Intelligence (XAI) offer solutions to address the limitations of black-box models. [3] explains that "XAI proposes the development of a set of machine learning (ML) techniques using explainable models that maintain a high level of learning performance." By placing models such as the Explainable Boosting Machine (EBM) in a "glass box," the recommendation system can provide explanations for each feature's contribution to the recommendation. This approach can also aid the curriculum evaluation process by identifying competency gaps in an objective and measurable way.

This study aims to propose a job recommendation system that integrates the Outcomes-Based Education (OBE) framework with Explainable Artificial Intelligence (XAI) techniques through the EBM model. The novelty of this study lies in two stages: the first stage relies on manually curated competency profiles; this study uses CLOs (Clinical Learning Objects) as a representation of student competencies for job matching. Generally, previous recommendation studies have focused only on the alignment of CLOs with PLOs for accreditation purposes rather than on the alignment of CLOs with actual industry needs. The second stage of this study addresses specific methodological challenges related to CLO-based recommendations and the lack of an independent, relevant basis for course-job alignment. Instead of adopting a classification framework that creates a circular dependency between labels and predictive features, EBM (Evidence-Based Modeling) is reengineered as a transparent decomposition of similarity-driven ranking scores. It is hoped that this system can bridge the gap between academia and industry, thereby providing recommendations that are not only accurate but also transparent, methodologically sound, and accountable.

2. Related Work

2.1. Explainable Job Recommendation System

Job Recommendation Systems (JRS) have developed over time and evolved from a collaborative content-filtering approach to deep learning-based and hybrid models. However, most high-performing models remain black-box systems, making them difficult to explain to users, particularly students and curriculum providers [4]. [2] Explains that "an explainable job recommendation system aims to provide high-quality recommendations as well as explanations that are easy to understand and insightful."

Several recent studies have focused on the domain of work [5] We propose a framework that combines local and global SHapley Additive exPlanation (SHAP) with natural language justifications using a Large Language Model (LLM) for a job recommendation system based on ONET aptitude profiles. Local explanations will identify the features that have the most positive influence on individual recommendations, while global explanations will capture the domain characteristics of the recommended items. Expert evaluation results show that acceptance scores are quite high, ranging from 3.7 to 4.4 on a Likert scale. [6] This paper discusses the multi-stakeholder challenges in JRS—such as candidates, recruiters, and companies—and proposes a research direction focused on systems capable of providing explanations tailored to stakeholders, particularly through Graph Neural Networks and Knowledge Graphs. On the other hand, an approach was taken that combines the BERT cross-encoder with an ensemble (XGBoost) and SHAP or Local Interpretable Model-Agnostic Explanations (LIME) to improve transparency in text-based job matching.[7].

Although these studies enhance transparency through post-hoc methods, most of them still rely on black-box models as the core of their predictions. Furthermore, integration with the OBE framework and the use of CLO as a representation of competencies remain very limited.

2.2. Explainable Boosting Machine in Job Recommendation System

EBM is an interpretable, glass-box machine learning model; this model was successfully developed by [8] and implemented as the first state machine model in the InterpretML framework by [9]. EBM is a Generalized Additive Model (GAM) based on cyclic gradient boosting with automatic interaction detection. Its mathematical form can be written as follows.

$$g(E[y]) = \beta_0 + \sum_j f_j(x_j) + \sum_{ij} f_{i,j}(x_i, x_j)$$

Where:

- $g(\cdot)$: a link function to adapt the model to the type of problem (regression or classification)
- E[y] : the expected value (mean) of the target y.
- $\beta 0$: The baseline prediction value when all features are zero (constant).
- xj : individual input variables (Similarity, Keyword count).

$fj\ (xj)$: A function that describes the contribution of feature xj to the prediction.
$fij(xi, xj)$: A function that automatically calculates the interaction between two features (xi and xj).

Because of the additive nature of these automatically selected feature interactions, the contribution of each feature to the prediction can be visualized directly without the need for post-hoc methods [9]. Tran [4] is the only empirical study most relevant to job recommendation systems. In his thesis, Tran compared EBM with Factorization Machine as well as the post-hoc techniques LIME and kernel SHAP on the 2012 CareerBuilder dataset using the hit_rate@5 metric. As previously noted, Tran does not explicitly address the risk of circular dependency that can arise when training labels are derived from two prediction features, particularly in domains that lack independent ground truth regarding the suitability of CLO for a job [10].

3. Metodologi

This study employs a systematic methodology to develop an OBE-based job recommendation system using EBM. The system's architecture consists of a two-component framework comprising two main components.

1. The retrieval component, which ranks job openings based on a composite relevance score
2. The explanatory component, in which EBM breaks down the contribution of each feature to the composite score of each ranked job.

EBM does not alter the rankings, but it serves as an alternative decomposition model that provides transparent attribution regarding the composition of the scores. The overall process consists of six main stages: data collection, preprocessing and NLP, feature engineering, model selection and justification, EBM modeling, and system implementation. The complete workflow is illustrated in Figure 1.

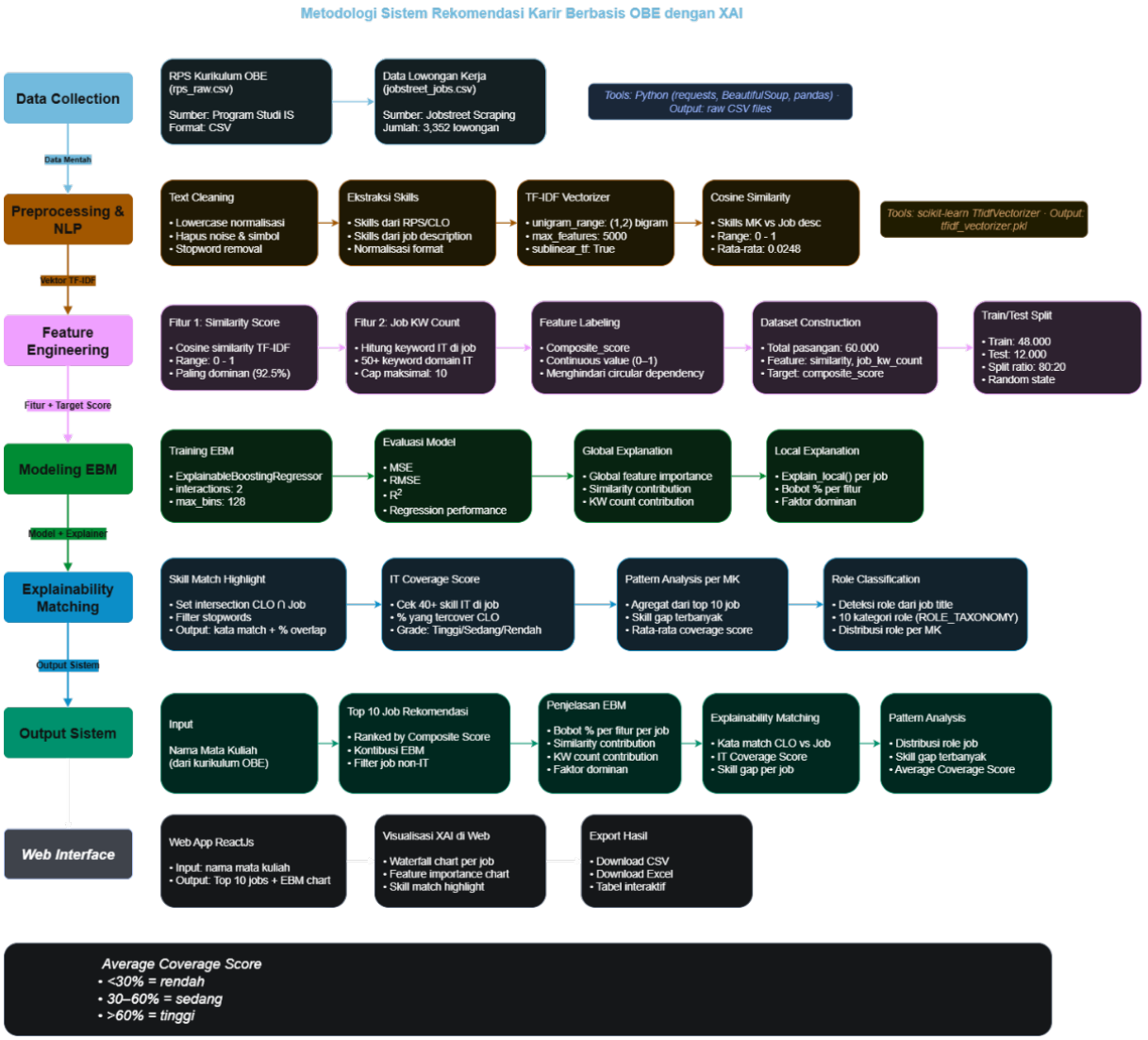

Fig. 1. Research methodology for an explainable OBE-based job recommendation system

3.1. Data Collection

In this study, the data used comes from two main sources: first, a curriculum dataset—specifically, the Semester Learning Plans (RPS) from the information studies program, which detail the PLOs and CLOs for a total of 48 courses; and second, a dataset of real job postings obtained from JobStreet, comprising 3,352 job listings. Both datasets are stored in CSV format for further processing. The 48 courses were then categorized into three groups based on an interpretation of the 2010 IS curriculum guidelines[11] IT-specific (15 courses), general IT (21 courses), and non-IT (12 courses). This categorization was used to limit the analysis of aggregate gaps to the 36 IT-related courses.

3.2. Pre-Processing and NLP

The text data in the CLO and job descriptions undergo several preprocessing steps, including lowercase normalization, noise and symbol removal, and the removal of stopwords not defined by IT. This is done to bridge the vocabulary gap between academic CLO terminology and industry terminology (e.g., the term “data dasar” in CLO versus “basis data” in job postings). This is a curated translation (CLO_TO_INDUSTRY) consisting of more than 100 term mappings applied prior to skill matching. This dictionary translates Indonesian academic phrases into standard industry technical terms across various domains such as programming, databases, networking, cloud computing, and enterprise architecture [12].

Skill entities were then extracted from the CLO data and job descriptions, which were normalized using curated IT skill terms (IT_SKILL_TERMS). Next, TF-IDF vectorization was applied to convert the text into numerical vectors using unigrams and bigrams, with a maximum of 5,000 features, and the TF-IDF formula was defined as follows [13][14].

$$\omega_{t,d} = tf_{t,d} \times \log(\frac{N}{df_t})$$

Where:

$\omega_{t,d}$: the weight of term t in data d.

$tf_{t,d}$: term frequency, the number of times the word t appears in d.

N : total number of data entries in the collection (all CLOs and job descriptions).

df_t : frequency data, the number of data points containing the word t

$\log(\frac{N}{df_t})$: inverse data frequency: the less frequently a word appears in a large dataset, the higher its weight.

Simply put, TF-IDF assigns a high score to words that appear frequently in one dataset but rarely in other documents—words that are too general, such as “system.”

Next, to calculate the cosine similarity between the specialized learning outcomes and the job descriptions in order to measure textual relevance, the formula is defined as follows[14].

$$\text{sim}(A, B) = \frac{A \cdot B}{||A||||B||}$$

Where:

A : TF-IDF vector from CLO

B : TF-IDF vector from the job description

$A \cdot B$: dot product of two vectors

$||A||$: norm of vector A

$||B||$: norm of vector V

A simple way to measure the angle between these two vectors is using a value of 1 or 0: a value of 1 means the two texts are very similar, and a value of 0 means the two texts are not similar.

3.3. Feature Engineering

Based on the TF-IDF vector stage and the results of the skill extraction, two main features were constructed using similarity and IT keywords, which can be explained as follows:

1. Similarity score, which is a measure of the similarity between the CLO and the job description, expressed as a number on a scale from 0 to 1.
2. Number of job keywords, which is the number of IT keywords found in the job description, with a maximum limit of 10.

Job postings are then filtered to ensure they contain at least four keywords, to ensure that low-relevance postings are eliminated before ranking takes place. These two features are then combined into a single score—a composite score—using a limited formula such as:

$$Composite\_score = Similarity \times \left(1 + \frac{job\_kw\_count}{10}\right)$$

Where:

$Similarity$: Similarity scores between CLO and job descriptions (range 0 to 1).

Job_kw_count : Number of IT keywords detected in the job description (maximum of 10).

$Composite\_score$: the scores that will be used to rank job openings

1 : for the multiplier (auxiliary) factor when the result is 0 or nothing at all

This formula is known as a limited multiplicative heuristic, so it is derived entirely from BM25, but it adopts the idea of relevance boosting without using IDF, term frequency saturation, or document length normalization as in BM25 [15] This similarity serves as the primary signal, while the number of keywords acts as an additional reinforcing factor. The final ranking is then determined by the composite score; 60,000 data pairs were constructed and divided into an 80% training set and a 20% test set.

3.4. Modelling

The model used in this study is the Explainable Boosting Machine (EBM), implemented as the `ExplainableBoostingRegressor` from the interpretML library [9]. Thus, EBM is a GAM-based glass-box model that offers high accuracy while maintaining inherent interpretability; furthermore, EBM also produces additive and precise feature attributions as an intrinsic property of its architecture [8]. The main parameters used are:

1. Feature name: similarity, number of job keywords
2. Interactions: 1
3. Maximum Bins: 128
4. Minimum number of leaf samples: 20
5. learning rate: 0.01
6. Maximum spins: 30

Next, the performance of the devaluation model was evaluated using Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the R^2 score. These metrics measure how accurately the EBM reproduces the composting evaluation function, not the empirical accuracy of the work recommendations.

3.5. Explainability and Matching

After the model is trained, an interpretive analysis is then conducted at two levels:

1. Global Explanation: an explanation of the importance of features across all trained models, obtained via `explain_global`
2. Local Explanation, which is an explanation of the contribution of each feature to individual job recommendations, obtained through `explain_lokal` [9].

Following decomposition by EBM, an analysis of the curriculum's alignment with industry was conducted, and work skill coverage scores were calculated according to the Precision and Information Retrieval formula, as follows:

$$Job\ Skill\ Coverage\ Score = \frac{|CLO \cap Job\ IT\ Skills|}{|Job\ IT\ Skills|} \times 100\%$$

Here, "Job-Related IT Skills" refers to a set of specific IT terms identified through IT_SKILL_TERMS, while its intersection represents terms that also appear in the relevant CLO following the vocabulary in Section 3.2. This metric operates independently of the EBM model and provides a post-hoc view of the alignment between the curriculum and industry requirements, including the identification of skill alignment and skill gaps [14].

3.6. System Implementation and Outputs

The implementation and output of this system is a web-based application built with React.JS and a FastAPI backend. Users can select a course name from the OBE curriculum, and the system will display:

1. Top 10 job recommendations, ranked based on composite scores

2. An explanation of local and global EBM for details on the contributions of the two features (similarity and number of job keywords).
3. Skill-matching highlights and skill identification
4. Visualization of the aggregate job-skill coverage scores for each course across all IT-specific courses

4. Results

In this section, the researchers present the results of experiments conducted on the job recommendation system they have developed; the evaluation focuses on model fidelity, feature contribution patterns, recommendation outputs and explanations, as well as an analysis of the gap between the curriculum and industry needs and the system's implementation.

4.1. Model fidelity

The EBM regression model was trained using 60,000 data pairs, consisting of 48,000 training pairs and 12,000 separate test pairs, with an 80:20 split. The model's evaluation metrics are presented in Table 1.

Table 1. Metrics Evaluation Results

Metrics	Value
MSE	0.000002
RMSE	0.0014
R^2	0.9984

The R^2 value indicates that EBM accurately generates the composite score used as its training target. It is hoped that these results are not an indication of predictive accuracy, since the composite score is constructed deterministically from the same two input features (similarity and number of job keywords) using the formula in Section 3.3. EBM functions as a surrogate decomposition model that reconstructs a transparent function on both features, rather than as a predictive model that ranks jobs independently from scratch.

4.2. The Importance of Global Features

Global explanation generated by EBM's explain_global() [9], calculating the average absolute contribution across all training instances for both features (similarity and number of job keywords) and with a single interaction term, resulting in the configuration `interactions=1` described in Section 3.4; the interpretability scores are normalized so that their sum equals 100%. Using the following calculation formula:

$$Importance\ Precentage_i = \frac{importance_i}{importance_{Similarity} + importance_{Keyword}} \times 100\%$$

Thus, in this paragraph, the empirical results on global feature importance show that the similarity score contributes 92.5%, while the number of job keywords contributes 7.5%. These results indicate that the similarity between students' competencies and job descriptions is the dominant factor in determining the recommendation score. Meanwhile, the number of job keywords acts as a supplementary feature that reinforces the recommendation through a composite scoring mechanism.

4.3. Examples of recommendations and explanations

Table 2 presents an illustrative example of the top three job recommendations, with an explanation based on one example from the Advanced UI Development course. Job rankings are determined by a composite score, not by the proportion of EBM contributions.

Table 2. Examples of recommendations for one of the top 3 courses

Rank	Job tittle	Similarity	Job keywords	Composite Score	EBM Contribution similarity
1	UI/UX Designer	0.2458	5/10	0.3687	96.12%
2	Frontend Engineer	0.1749	9/10	0.3324	98.58%
3	Full Stack Developer	0.1676	9/10	0.3184	98.58%

For rankings 2 and 3, the similarity contribution of the EBM is identical at 98.58% because both works have the same number of keywords (9/10). Based on the composite score formula, the contribution percentage depends on the relative weight of the similarity and the number of keywords in the work, not on the absolute value of the similarity. The ranking between these works is determined by the composite score. When composite scores are tied, similarity is used as a secondary ranking factor.

The difference in scale between similarity values of 0.17–0.25 and EBM contribution percentages of 96–99% indicates two distinct quantities. Similarity is a limited cosine score that measures CLO and job descriptions, whereas the EBM contribution percentage is a decomposition of keyword counts limited to the multiplication range (1.0x–2.0x); similarity remains the dominant component or feature of the composite score even when its absolute value is low.

4.4. Gap analysis

Curriculum alignment with the industry uses the work skill scope score defined in Section 3.5, where these work skills are a specific, unweighted set of IT skills identified in job descriptions via IT_SKILL_TERMS, and the numerator counts the terms in the corresponding CLO set after vocabulary normalization in Section 3.2, Each skill is assigned the same weight—that is, no skill is considered more important than another in this formulation, which is acknowledged as a simplification in Section 6—for each of the 36 IT-specific courses, across the top 10 recommended jobs that have been ranked by composite score. The coverage for each observation is calculated independently using the formula 36 × 10 = 360 job observations, with an unweighted (aggregated) average for all 360 observations presented in Table 3.

Table 3. Coverage score

Job mean coverage score	**Courses with coverage > 0%**	**Courses with = 0%**
11.93%	31 Courses	5 courses

The skills that most frequently do not appear in the CLO relative to job postings can be analyzed as follows: PostgreSQL does not appear in 33 out of 36 IT courses; SQL does not appear in 32 out of 36 IT courses; MySQL does not appear in 32 out of 36 IT courses; and Python does not appear in 30 out of 36 IT courses. Furthermore, the 5 courses with a coverage of 0 consist of conceptual learning; therefore, a coverage of 0 requires interpretation at the course level.

4.5. System Implementation

This system is implemented as a web-based application with a React.js front end and a FastAPI back end. The web application interface provides three main pages: Dashboard, Gap Analysis, and Explanation.

1. The dashboard displays course selections and the top 10 job recommendations, along with features that contribute to job rankings
2. Gap analysis: This page includes a distribution table for skill coverage, a table showing coverage by course with a filter for courses that have already been categorized, and a table showing the frequency of skill gaps across all IT courses
3. The explanation shows the feature's global and local contributions, as well as the skills that have been matched.

5. Conclusion

The results of this study present an explainable job recommendation system based on the OBE principle that integrates CLO with job descriptions. This system has a two-component framework: a retrieval component that ranks job openings based on a composite score derived from cosine similarity and the number of job keywords, and an explanation component in which EBM functions as a surrogate decomposition model. In this role, EBM does not generate or determine rankings; rather, it reconstructs the composite score in additive form and attributes the contribution of each feature, thereby providing transparent global and local explanations.

Thus, by using EBM as a decomposer of scores rather than an independent predictive classifier, this study avoids treating model performance metrics as evidence of the accuracy of empirical recommendations. The observed R^2 value of 0.9984 and RMSE of 0.0014 demonstrate that EBM successfully reproduces the specified composite value function. A substantial finding emerged from the analysis of the gap between the curriculum and industry needs: there are 36 IT-specific courses, and the average composite score for job-related skills coverage stands at 11.93%. These skills are related to databases such as PostgreSQL, SQL, and MySQL. These results provide evidence regarding the level of coursework that can support an OBE curriculum review.

6. Limitations and the Future

6.1. Limitations

In the research that has been conducted, there are several limitations that must be acknowledged. The first limitation is the absence of independent reference data, which prevents the absolute validation of the quality of the recommendations; the system can only be evaluated relatively by determining which jobs are ranked higher relative to others for a specific degree program, rather than confirming whether the recommendations are objectively correct. The second limitation is that the formulation of the composite score restricts the explanation of local EBM to a narrow range, as similarity is designed to dominate the score; by design, this limits the extent to which the contribution of local features can differ meaningfully across individual recommendations. The third limitation is that the mapping of CLO vocabulary to industry terms is still done manually, and although it has been substantially improved through iterative refinements, it remains incomplete. The fourth limitation is that the current TF-IDF pipeline does not explicitly perform cross-language alignment, so Indonesian and English equivalents are treated as different tokens unless they are linked through a mapping dictionary. The fifth limitation is that job vacancy data was collected from a single platform (Jobstreet Indonesia) over a limited time period; as a result, the findings are susceptible to platform-specific bias, temporal shifts in labor demand, and noise inherent in online job postings. The sixth limitation is that this dataset is limited to a single information studies program, and no formal usability evaluation was conducted with faculty or students.

6.2. *The Future*

Future work should focus on integrating semantic embedding models such as IndoBERT and SentenceBERT to comprehensively address the vocabulary gap between academic CLO language and industry terminology. Although manual vocabulary mapping has already been performed, further work could also explore a comparative analysis between EBM and ad hoc explanation methods -hoc explanation methods such as SHAP to evaluate the consistency of feature attributions. Additionally, it is necessary to expand the dataset to include various academic programs and job posting platforms, regularly update job posting data to reflect market changes, and conduct usability studies with faculty and students to strengthen the generalizability of this system's practical application.

References

[1] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should i trust you?' Explaining the predictions of any classifier," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, Aug. 2016, pp. 1135–1144. doi: 10.1145/2939672.2939778.

[2] Y. Zhang and X. Chen, "Explainable recommendation: A survey and new perspectives," Mar. 11, 2020, *Now Publishers Inc*. doi: 10.1561/1500000066.

[3] A. B. Arrieta *et al.*, "Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI," Dec. 2019, [Online]. Available: http://arxiv.org/abs/1910.10045

[4] T. Hoang and A. Tran, "MSc Business Information Technology Final Project Explainable Artificial Intelligence in Job Recommendation Systems," 2023.

[5] N. R. M and M. B. K.P, "A hybrid explainability framework for recommender systems using SHAP with natural language interpretations," *International Journal of Innovative Research and Scientific Studies*, vol. 8, no. 6, pp. 687–703, Sep. 2025, doi: 10.53894/ijirss.v8i6.9669.

[6] R. Schellingerhout, "Explainable Multi-Stakeholder Job Recommender Systems," in *RecSys 2024 - Proceedings of the 18th ACM Conference on Recommender Systems*, Association for Computing Machinery, Inc, Oct. 2024, pp. 1318–1322. doi: 10.1145/3640457.3688014.

[7] "Explainable and Scalable Job Recommendation System using".

[8] Qiang. Yang, *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012.

[9] H. Nori, S. Jenkins, P. Koch, and R. Caruana, "InterpretML: A Unified Framework for Machine Learning Interpretability," Sep. 2019, [Online]. Available: http://arxiv.org/abs/1909.09223

[10] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman, "Leakage in data mining: Formulation, detection, and avoidance," in *ACM Transactions on Knowledge Discovery from Data*, Dec. 2012. doi: 10.1145/2382577.2382579.
[11] H. Topi *et al.*, "IS 2010: Curriculum guidelines for undergraduate degree programs in information systems," *Communications of the Association for Information Systems*, vol. 26, no. 1, pp. 359–428, 2010, doi: 10.17705/1cais.02618.
[12] F. Gurcan and N. E. Cagiltay, "Big Data Software Engineering: Analysis of Knowledge Domains and Skill Sets Using LDA-Based Topic Modeling," *IEEE Access*, vol. 7, pp. 82541–82552, 2019, doi: 10.1109/ACCESS.2019.2924075.
[13] C. B. Gerard Salton, "Term Weighting Approaches in Automatic Text Retrieval," 1987.
[14] C. D. . Manning, Prabhakar. Raghavan, and Hinrich. Schütze, *Introduction to information retrieval*. Cambridge University Press, 2009.
[15] S. E. Robertson and S. Walker, "Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval."