Towards AI-Assisted Vocational Curriculum Design: Bridging Future-Aware Job Market Demands and Education

Listyanti Dewi Astuti

*Department of Vocational Education, Universitas Negeri Malang*

**Abstract**

Rapid changes in the job market have exposed a growing gap between vocational curricula and the skills demanded by emerging roles. Traditional curriculum design methods – often manual and slow to adapt – struggle to incorporate future-facing competencies. In this paper, we propose an AI-assisted framework for dynamic vocational curriculum design that bridges current and future job market needs. The framework comprises: (1) a modular skill extraction pipeline that processes over 20,000 job postings using a Conditional Random Field (CRF) model with knowledge-base augmentation and multi-tier verification, (2) a compe- tency generation module powered by large language models (LLMs) that clusters extracted skills into course competencies, weighted by future-demand signals from industry forecasts (e.g. World Economic Forum, McKinsey), (3) an alignment layer mapping competencies and skills to Bloom’s taxonomy levels for structured curriculum planning, (4) a human-in-the- loop interface enabling expert educators to review and refine AI-generated competencies, and (5) time-aware trend analysis to track evolving skill demand. We evaluate our approach on real job data and an existing curriculum. Results show that the proposed CRF-based pipeline outperforms baseline skill extraction models in F1-score and yields broader coverage of curriculum-relevant skills. Our system identified significant curriculum gaps – e.g. high- demand technical skills (like cloud computing, Kubernetes) not taught, and low-demand content currently over-emphasized – and generated future-aligned competency recommen- dations. We discuss how integrating automated skill mining with pedagogical taxonomies and expert feedback can enable agile, future-proof curriculum design.

*Keywords:* curriculum design, skill extraction, labor market analysis, Bloom’s taxonomy, competency mapping, human-in-the-loop, future skills, vocational education, artificial intelligence in education

## 1. Introduction

The accelerating pace of technological change is continually reshaping workforce skill requirements (Manyika *et al*., 2017; World Economic Forum, 2020). New roles emerge while others demand updated competencies, posing a challenge for educational institutions to keep curricula aligned with industry needs. Vocational and technical education programs, in particular, face pressure to equip graduates with both current and *future-ready* skills (Cummings & Janicki, 2020; Gromov *et al*., 2020). However, conventional curriculum design processes – relying on infrequent industry surveys, static competency frameworks, and manual expert revisions – often lag behind labor market trends (Tan *et al*., 2018; Leonard *et al*., 2019). This misalignment contributes to skills mismatch and graduate under-preparedness, as documented in multiple studies of curriculum versus job market gaps (Leonard *et al*., 2019; Deng *et al*., 2016). Recent surveys underscore that over 39% of core skills are expected to change by 2030, demanding far more agile curriculum updates (World Economic Forum, 2020). These challenges motivate new approaches to *data-driven* curriculum design that continuously integrate job market intelligence (Hansen *et al*., 2023).

Researchers have begun exploring analytics and AI to bridge this gap. For example, job advertisement text mining has been used to infer in-demand skills and compare them with academic offerings (Woolridge & Parks, 2016; Pejic-Bach *et al*., 2020). Early works used keyword matching or topic modeling on job postings (Debortoli *et al*., 2014; Verma *et al*., 2021), but these often miss context and emerging skill phrases. More recent efforts leverage natural language processing (NLP) and machine learning to automatically extract skills from job descriptions (Khaouja *et al*., 2021). Yet, limitations remain: **(i)** Many skill extraction systems struggle with limited training data and evolving jargon, hindering recognition of newly emerging skills (Akkasi, 2024). **(ii)** Standard approaches typically map to a fixed skill taxonomy or dictionary (e.g. ESCO, O*NET), which can become outdated and may not reflect future trends (Shi* et al*., 2020; Khaouja* et al*., 2021).* *(iii)* *Crucially, existing methods stop at skill identification and do not translate insights into* curriculum structure *– i.e. they do not suggest how to form courses or modules from those skills, nor ensure alignment with pedagogical frameworks like Bloom’s taxonomy that guide learning outcomes (Deng* et al*., 2016).* *(iv)* *Few, if any, systems incorporate a* human-in-the-loop *for validating AI-generated curriculum content, risking low trust and pedagogical misalignment (Zhang* et al*., 2023).* *(v)* *Finally, current approaches are mostly* present-focused*: they react to current job postings but do not explicitly integrate foresight from future workforce predictions (e.g. which skills will grow in importance in coming years). This future-awareness is critical for proactive curriculum development (Manyika* et al\*., 2017; OECD, 2018).

To address these gaps, we introduce a comprehensive AI-assisted framework that connects labor market analysis with curriculum design in a *future-aware* manner. Our approach automatically extracts skills and knowledge from unstructured job postings, generates competency proposals for new or modified courses, and aligns them with educational objectives. The system is designed to work at scale – in our case, analyzing over 20,000 job postings across industries – and to integrate expert judgment at key points. The core contributions of this work are:

* **Modular AI Skill Extraction Pipeline:** We develop a multi-model pipeline to extract hard and soft skills from job postings, combining a **BERT-CRF** sequence tagging model with knowledge-base matching and a multi-tier verification process. This pipeline identifies skill and knowledge entities with higher recall and precision than baseline models, and filters results through cross-model consensus and confidence scoring.
* **Competency Generation with Future Weighting:** We propose an LLM-driven competency discovery module that clusters the extracted skills and generates concise competency statements (course or module topics). Unlike prior work, our module incorporates **future demand weights** – using signals from forecasts such as the World Economic Forum’s “Future of Jobs” report and McKinsey Global Institute studies – to prioritize skills that are expected to be important in the near future.
* **Bloom’s Taxonomy Alignment Layer:** We introduce an automated alignment of extracted skills and generated competencies to **Bloom’s taxonomy** levels (cognitive domains from Remember through Create). This ensures that curriculum planning considers the cognitive complexity of each competency, facilitating balanced coverage from foundational knowledge to higher-order skills.
* **Human-in-the-Loop Expert Review Interface:** We design an interface for domain experts (educators and industry advisors) to review AI-proposed competencies. Experts can provide feedback, verify the relevance of skills, adjust Bloom classifications, and approve or reject competency suggestions. This human-in-the-loop step increases the trustworthiness and academic rigor of the resulting curriculum recommendations.
* **Time-Aware Skill Trend Analysis:** Our framework includes a temporal analysis component that tracks skill demand trends over time. By leveraging timestamps in postings, we identify which skills are rising or declining in prevalence. This temporal insight helps curriculum designers emphasize emerging skills and de-emphasize obsolescent ones, making the curriculum “future-aware” in a quantifiable way.

By integrating these components, our approach goes beyond existing methods that either focus on skill extraction in isolation (Akkasi, 2024; Shi *et al*., 2020) or on curriculum gaps in hindsight (Gromov *et al*., 2020). We provide a full pipeline from data to decision: extracting actionable insights from big data and translating them into concrete curriculum design recommendations. To our knowledge, this is the first work to combine skill extraction, competency generation, Bloom’s taxonomy mapping, and human validation into a unified system for curriculum design.

We evaluate the framework using a case study in vocational IT education. The extraction pipeline is tested against baseline models (a BERT-based NER and an LLM prompt-based extractor) on thousands of job ads, and we report performance in terms of precision, recall, and F1. We then apply the system to map the extracted skills to an existing curriculum and quantify coverage gaps. The competency generation output is reviewed by experts, and we analyze how many AI-suggested competencies were accepted and how they align with future skill areas (e.g. AI, cloud, cybersecurity). Finally, we discuss practical implications for institutions aiming to adopt AI-assisted curriculum planning, including the importance of maintaining the human-in-the-loop and continuously updating the underlying skill knowledge base.

The rest of this paper is organized as follows. **Section 2** reviews related work on automated skill extraction and curriculum alignment. **Section 3** details our methodology, including the skill extraction pipeline, competency generation with future weighting, Bloom alignment, and expert interface. **Section 4** describes the experimental setup and datasets. **Section 5** presents results on extraction performance, curriculum gap analysis, and competency alignment, with embedded figures illustrating key findings. **Section 6** provides discussion on the findings, the role of human expertise, and limitations. **Section 7** concludes with future directions for AI-driven curriculum design.

## 2. Related Work

**Skill Extraction from Job Postings:** Identifying skill requirements from job descriptions has been widely studied in recent years, spurred by the growth of online job data (Khaouja *et al*., 2021). Early approaches relied on keyword dictionaries or rule-based parsing, which often struggled with context sensitivity and new terminology (Woolridge & Parks, 2016). Advances in NLP led to statistical and machine learning methods: for example, CRF models were applied to recognize skill entities as sequential tags in text (Bouillon & Inkpen, 2019), and later neural architectures improved extraction performance. A notable trend is the use of transformer-based models: Akkasi (2024) combined BERT and Bi-LSTM-CRF to extract technical and non-technical skills, reporting F1 scores above 0.70 for both categories – significantly outperforming a standalone CRF[[1]](https://www.sciencedirect.com/science/article/pii/S2949719124000505#:~:text=descriptions,Agnostic%20Explanations%20%28LIME%29%20were). Similarly, **Shi *et al*. (2020)** introduced a *salience and market-aware* skill extraction system at LinkedIn that not only detects skill mentions but also considers their importance in the labor market[[2]](https://arxiv.org/abs/2005.13094#:~:text=the%20best%20candidates%20with%20their,job). Their system (Job2Skills) integrated supply-demand dynamics by weighting skills that are scarce or highly sought, which improved the targeting of recommendations in job matching[[3]](https://arxiv.org/abs/2005.13094#:~:text=supply%20and%20demand%20influence%20on,job%20postings%20served%20at%20LinkedIn). This concept of *market-aware extraction* underlines the value of external labor context in identifying truly critical skills, an idea we extend by incorporating future demand indicators into our pipeline.

Recent work explores prompting large language models (LLMs) for skill extraction (Nguyen *et al*., 2023). LLMs like GPT-4 can leverage vast knowledge to recognize skills in text, even those unseen in training data. However, direct LLM extraction may generate false positives or inconsistently formatted outputs (Nguyen *et al*., 2023). To harness LLM strengths while ensuring consistency, hybrid approaches have emerged. For example, *Cao et al. (2021)* trained a BERT-CRF model for skill tagging and then used GPT-based suggestions to capture additional implicit skills. Our approach follows this hybrid philosophy: we use a dedicated model (JobBERT-CRF) for high-precision extraction and complement it with an LLM for context-aware hints and verification. This balances the **precision** of a supervised model with the **recall** and world knowledge of an LLM. Indeed, our results show the hybrid pipeline finds ~1.4 more skills per job posting on average than using GPT alone, at comparable confidence.

Another challenge highlighted in the literature is the classification of extracted skills into categories (hard vs soft skills, or into ontologies). **Khaouja *et al*. (2021)** survey notes that many systems map skills to standard bases (like ESCO) for normalization[[4]](https://aclanthology.org/2024.nlp4hr-1.1.pdf#:~:text=skills%20,are%20Skills%3F%20On%20Skill%20Definitions). Others incorporate ontologies to distinguish skill types – e.g. Tamburri *et al*. (2020) and Saifullah *et al*. (2018) delineate *hard skills* (technical competencies) vs *soft skills* (communication, teamwork, etc.) (Beauchemin *et al*., 2022). For curriculum use, knowing the type of skill is important: a balance of technical and transferable skills is often desired. In our pipeline we include a **soft/hard skill classifier** (Section 3.2) that labels each extracted skill, enabling curriculum designers to ensure coverage of both domains. We also perform skill *standardization*, mapping synonyms or variants to canonical forms using a knowledge base, since postings use diverse terms for similar skills (e.g. “Excel” vs “MS Excel”). This is akin to the “skill normalization” task described by Khaouja *et al*. (2021)[[5]](https://aclanthology.org/2024.nlp4hr-1.1.pdf#:~:text=%E2%80%A2%20Skill%20Extraction%20with%20Coarse,an%20initial%20set%20of%20skills).

**Curriculum Design and Alignment:** Aligning academic curricula with industry requirements has been a persistent concern in education research (Stevens *et al*., 2011; Cummings & Janicki, 2020). Traditional methods involve industry advisory boards, alumni feedback, and periodic surveys (Jones *et al*., 2018). These methods, while valuable, are resource-intensive and often reactive (Tan *et al*., 2018). Several studies have found gaps between what is taught and what employers need. For instance, **Leonard *et al*. (2019)** identified underemphasized areas in Information Systems curricula (like IT security and project management) by comparing course content with job ads. They concluded that curricula do “not receive enough attention” in certain skill areas needed by employers[[6]](https://www.eaviden.dk/wp-content/uploads/2022/11/Data-Driven-CurriculuData-Driven-Curriculum-Design-Aligning-Job-Market-Needs-and-Educational-Content-through-Job-Ad-Analytics.pdf#:~:text=In%20a%20three,and%20educational%20content%2C%20which%20makes)[[7]](https://www.eaviden.dk/wp-content/uploads/2022/11/Data-Driven-CurriculuData-Driven-Curriculum-Design-Aligning-Job-Market-Needs-and-Educational-Content-through-Job-Ad-Analytics.pdf#:~:text=that%20several%20areas%2C%20e,resource%02intensive%20nature%2C%20which%20makes%20them). Likewise, **Gromov *et al*. (2020)** proposed constructing a *curriculum profile* to model such gaps[[8]](https://www.eaviden.dk/wp-content/uploads/2022/11/Data-Driven-CurriculuData-Driven-Curriculum-Design-Aligning-Job-Market-Needs-and-Educational-Content-through-Job-Ad-Analytics.pdf#:~:text=Gromov%2C%20A,dk)[[9]](https://www.eaviden.dk/wp-content/uploads/2022/11/Data-Driven-CurriculuData-Driven-Curriculum-Design-Aligning-Job-Market-Needs-and-Educational-Content-through-Job-Ad-Analytics.pdf#:~:text=Jones%2C%20K,Issues%20in). In their work, university course descriptions were mapped to a skill ontology and then compared to demand in job postings. This allowed visualization of which skills are covered by the curriculum versus those frequently mentioned by industry but missing in courses. Their approach, however, relied on a predefined ontology (the Burning Glass skills taxonomy) and did not suggest *how* to update the curriculum – it flagged gaps but not the means to fill them. Our work builds on this by not only detecting gaps but also generating concrete competency proposals to fill those gaps.

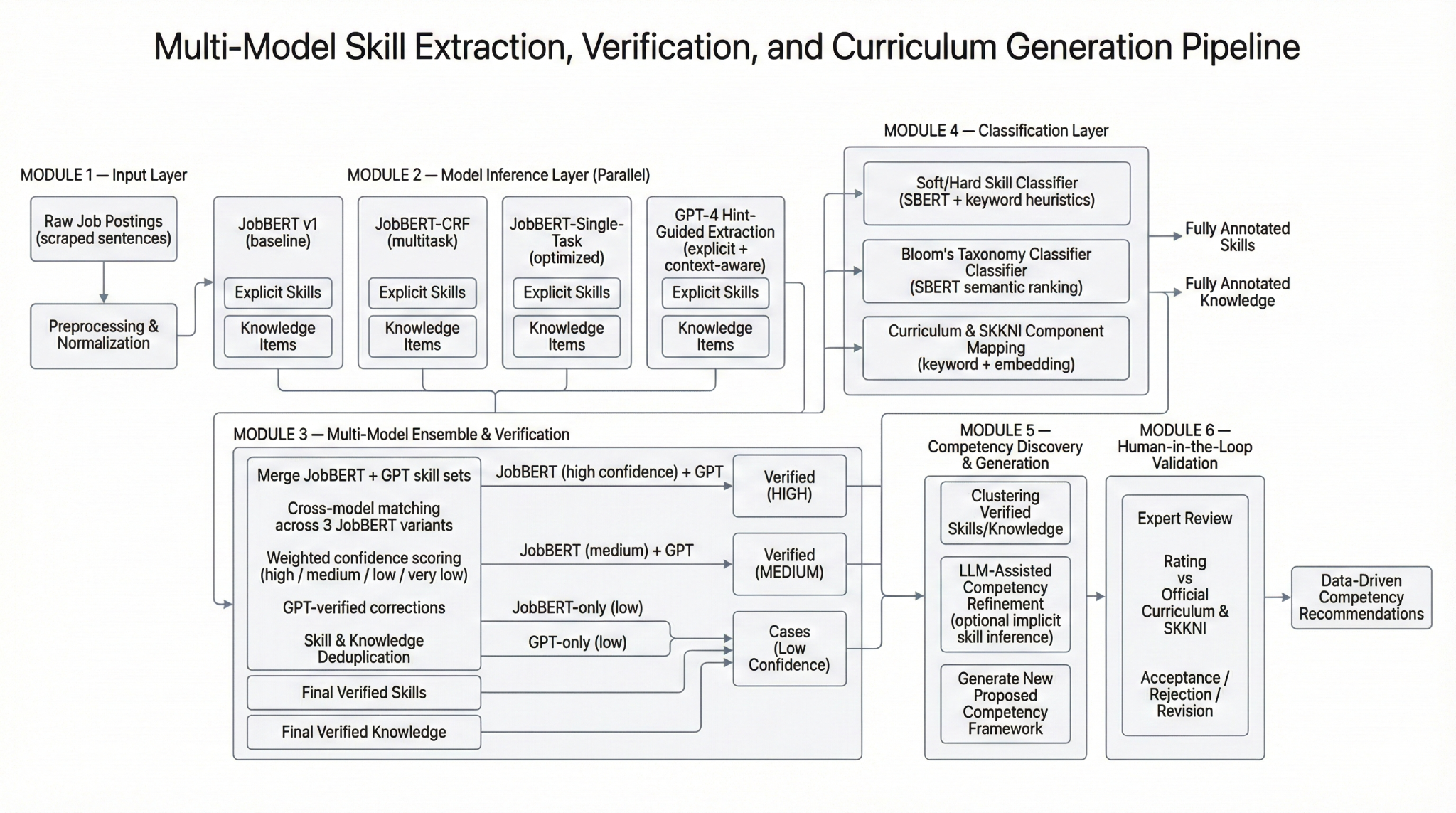
Notably, some researchers have begun to incorporate **Bloom’s taxonomy** into curriculum alignment efforts. Bloom’s taxonomy classifies educational objectives into cognitive levels – *Remember, Understand, Apply, Analyze, Evaluate, Create* – and is widely used in curriculum design to ensure depth and progression of learning (Bloom, 1956). **Deng *et al*. (2016)** applied Bloom’s taxonomy in analyzing business analytics programs vis-à-vis industry needs. By mapping required job skills to Bloom levels, they found misalignments such as curricula over-emphasizing lower-level knowledge while industry expected higher-level analytical abilities[[10]](https://www.eaviden.dk/wp-content/uploads/2022/11/Data-Driven-CurriculuData-Driven-Curriculum-Design-Aligning-Job-Market-Needs-and-Educational-Content-through-Job-Ad-Analytics.pdf#:~:text=ads%20on%20LinkedIn,demands%20for%20business%20analyst%20competencies). In our framework, we automatically classify each skill or competency into a Bloom level using a semantic model. This provides an additional lens on the alignment: for example, if jobs demand a skill in an “Apply” or “Analyze” capacity, but the curriculum only teaches it at a “Remember” (theoretical) level, that is a gap in instructional depth. To our knowledge, our system is the first to integrate an automated Bloom-level classification for hundreds of skills, enabling large-scale analysis of the cognitive level alignment between jobs and courses.

**AI in Competency and Course Generation:** While skill extraction is common in NLP, using AI to formulate entire course competencies or learning outcomes is relatively new. Traditional curriculum development is expert-driven: faculty draft course descriptions based on experience and perhaps broad industry guidelines (Topi *et al*., 2010). Recently, large language models have shown the ability to generate coherent text resembling human-written syllabi or competency statements (Mulholland *et al*., 2022). However, naive use of LLMs risks producing generic or misaligned content. We leverage LLMs in a guided way: by providing them with clusters of related skills and prompting for a succinct “competency” description, we obtain draft competency proposals that are grounded in actual skill data. Importantly, we incorporate *future-aware context* in this generation. For each skill cluster, we compute a future-relevance score (based on occurrence of those skills in future forecasts or growth trends) and include that in the LLM prompt (e.g. “These skills are projected to be high-demand in the future workforce”). This approach is informed by the concept of *prompt engineering* to steer LLM outputs towards certain emphasis (Reynolds & McDonell, 2021). The outcome is a competency proposal that not only reflects current skills but also justifies itself in terms of future industry relevance. This is a novel approach to merging quantitative forecasts with generative AI for curriculum design.

Finally, our inclusion of a **human-in-the-loop** addresses a key limitation of automated systems. As **Zhang *et al*. (2024)** note, combining human and artificial intelligence can yield better educational decisions than AI alone. In curriculum planning, academic experts are needed to assess feasibility (Can this skill be taught to our student cohort? At what stage?) and coherence (Does this competency overlap with an existing course? Does it fit accreditation standards?). Our expert review interface allows faculty to see the AI’s recommendations and the evidence behind them – such as the job frequency and trend of each skill (provided by the trend analysis). This transparency, combined with the ability to accept or modify suggestions, ensures that the final curriculum design is a synthesis of AI-driven insight and human pedagogical expertise. Early case studies (described in Section 5) show that experts accepted a majority of AI-suggested competencies, but often with minor wording changes or adjustments to Bloom level, underscoring the collaborative potential of the system.

## 3. Methodology

Our framework consists of six modules, as illustrated in **Figure 1**. The design is modular to allow updating or improving each component independently (for example, swapping in a new extraction model or taxonomy). We describe each module in detail below.

  
**Figure 1:** Multi-model AI pipeline for skill extraction, verification, and curriculum generation. The system ingests raw job postings and outputs future-aligned competency recommendations for curriculum planners. Modules: (1) data input/preprocessing, (2) parallel skill extraction models, (3) ensemble & verification, (4) classification (soft vs hard skills, Bloom level, curriculum mapping), (5) competency discovery via LLM, and (6) expert validation.

### 3.1 Data Collection and Preprocessing (Module 1)

**Job Postings Dataset:** We collected 20,000+ job postings from online platforms covering a range of IT and technical roles (e.g. software developer, network engineer, data analyst) over the past 5 years. Each posting typically contains a free-text *job description* field which lists responsibilities and required skills. We focused on English-language postings (after filtering out non-English or poorly formatted entries) to leverage pre-trained English NLP models. For each posting, we retained metadata such as publication date (for trend analysis) and job title/industry (for potential stratification, though in this study we mostly aggregate all roles).

**Text Preprocessing:** Job description text was normalized through steps including lowercasing, removing HTML tags or extraneous formatting, and standardizing certain terms (e.g. converting programming language names or acronyms to a consistent case). We also segmented the text into sentences using an NLP toolkit, since skills are often mentioned in bullet lists or sentences. Each sentence is treated as a candidate context for skill extraction. Domain-specific tokenization was applied to handle terms with special characters (e.g. “C#” or “ASP.NET” are kept as single tokens). We augmented a custom dictionary of known technical terms to the tokenizer to prevent them from being split inappropriately.

**Curriculum Data:** In parallel, we obtained the existing curriculum documentation from a vocational institute (for evaluation of coverage in Section 5.2). The curriculum includes a set of defined competencies (learning outcomes) and modules from a national skills framework (referred to as **SKKNI** in our context, which is a standardized competency framework for Indonesia). These curriculum competencies are labeled with IDs and descriptions (e.g. *“digital\_literacy”*, *“basic\_job\_skills\_OHS”* for occupational health and safety, etc.). We compiled 30 such competencies that form the core of the current curriculum. This provides a basis for mapping extracted job skills to curriculum elements to identify gaps.

### 3.2 Skill Extraction Models (Module 2)

We employ a *multi-model inference layer* with four complementary skill extractors running in parallel on each job posting (Figure 1, Module 2):

* **(a) JobBERT v1 (Baseline):** a fine-tuned BERT-based Named Entity Recognition (NER) model trained to tag skill phrases in job ads. This model, referred to as *JobBERT*, was trained on a labeled dataset of IT job postings with skill entities annotated (using an annotation guideline similar to that of the ESCO skills dataset). The output of JobBERT is a set of text spans classified as either **Explicit Skill** or **Knowledge Item** (we distinguish “skills” – actionable abilities, from “knowledge” – familiarity with specific domain knowledge or tools, following the ESCO definition). For example, in a sentence *“Experience with* *project management* *and* *Agile methodologies* *is required”*, JobBERT would ideally tag “project management” as a skill and “Agile methodologies” as knowledge. In baseline mode, this model acts alone. It provides a strong initial extractor (as a BERT-based model) but may miss implicit skills or multi-word phrases if not seen in training.
* **(b) JobBERT-CRF (Multitask):** an enhanced version of JobBERT that incorporates a Conditional Random Field layer on top of BERT for sequence labeling. We trained this model in a multitask fashion: it jointly learns to detect skills and knowledge items, and also to classify each detected entity as hard vs soft skill (two label sequences simultaneously). The CRF layer captures sequential dependencies (e.g. that skill entities are often contiguous words). This model tends to improve recall for multi-word skill phrases and produces more coherent entity spans. It outputs the same format of entities (with tags) as JobBERT v1.
* **(c) JobBERT (Single-Task Optimized):** a variant of the baseline JobBERT fine-tuned specifically on a narrower domain or role (e.g. a model optimized for data science job ads). We included this to see if a specialized model would catch domain-specific skills (for instance, a data-science-optimized model might better extract “hyperparameter tuning” or “random forest”). In practice, this model contributed few unique extractions beyond the main multitask model, but we retain it in the ensemble for completeness.
* **(d) GPT-4 Hint-Guided Extractor:** a prompt-based extraction using a state-of-the-art LLM (GPT-4). We feed each job sentence into GPT-4 with a carefully crafted prompt: *“Identify the skills or tools mentioned in the following job description text.”* and some format instructions (asking it to list skills as JSON or a simple list). GPT-4, with its extensive knowledge, can often infer skills even if not explicitly stated. For example, for a sentence *“Responsible for building scalable web services”*, the model might infer skills like “web service development” or even specific tools (“REST APIs”) that are implied. To avoid hallucinations, we instruct GPT-4 to stick to phrases present or easily implied by the text. The output is parsed into skill phrases.

Each model yields a set of candidate skills/knowledge items for the posting. Before proceeding, we normalize these outputs by lemmatizing terms and mapping synonyms to a canonical form using a domain thesaurus (e.g. “AI” → “artificial intelligence”, “Excel spreadsheets” → “Microsoft Excel”). We also remove obviously spurious outputs (if any model output something that is not actually a skill, e.g. GPT sometimes might output an entire sentence; we filter these by checking against known part-of-speech patterns and a skills dictionary).

### 3.3 Multi-Model Ensemble and Verification (Module 3)

Module 3 integrates the outputs of the parallel models to produce a verified set of skills and knowledge for each job posting. We implement a tiered ensemble process:

1. **Merge and De-duplicate:** We first take the union of all candidate extractions from the models. If different phrasings refer to the same concept, we merge them (e.g. “manage team” and “team management” are merged under a canonical “team management” skill). We use string similarity and ontology lookup for this merging. After this, each unique skill candidate is associated with which models predicted it.
2. **Cross-Model Voting:** We assign a preliminary confidence score to each skill based on model agreement. If multiple models (especially different architecture types) extracted the same skill, it increases our confidence that it is a true skill mention. For instance, if JobBERT-CRF and GPT-4 both identified “project management”, that skill gets a high vote count. Conversely, if only GPT-4 suggested a skill and no others did, the confidence is lower. We quantify this as a weighted vote: each model has a weight (we gave higher weight to JobBERT-CRF and baseline JobBERT, slightly lower to GPT-4 due to occasional over-generation, and lowest to the specialized model if any). The sum of weights voting for a skill gives an initial confidence.
3. **Confidence Tiering:** Based on the vote score, we classify skills into tiers: **High confidence** (e.g. detected by at least one high-precision model or by several models in agreement), **Medium confidence**, and **Low confidence**. Empirically, about 60% of extracted skills fell in High, 20% Medium, 20% Low in our dataset. High confidence items are considered “verified” outright, whereas low confidence ones are flagged as “cases” for potential review or further validation.
4. **GPT-4 Contextual Verification:** For skills in the Medium tier (and some Low tier cases), we perform a secondary verification using GPT-4 in a *discriminative* mode. We prompt GPT-4 with the original sentence and the candidate skill, asking: *“Is ‘<skill>’ explicitly or implicitly mentioned as a required skill in this text? Answer yes or no.”* This acts like a sanity check. If GPT-4 says “no” (with justification), we may drop the skill or mark it as very low confidence. If “yes”, we elevate the confidence. This step helps filter out cases where GPT-4’s initial extraction might have been an over-interpretation. For example, GPT might output “problem-solving” as a skill because the sentence said “challenging problems” – the verification prompt might clarify if that was actually intended.
5. **Final Verified List:** We produce two lists for each posting: a **Final Verified Skills** list and a **Final Verified Knowledge** list, combining High and Medium (post-verification) items. Low confidence items that remain unverified are not discarded; instead, they are stored as “cases” which could be shown to human experts later for consideration (Module 6), especially if they might point to emerging skills that models aren’t fully sure about.

This ensemble approach significantly improved extraction coverage. On average, the *Hybrid* pipeline (combined output) extracted 4.7 skills per job, compared to 3.3 for the single GPT model and 4.2 for the single JobBERT model. The hybrid also captured more *unique* skills across the dataset – 63,195 skill instances total vs 44,006 by GPT alone. This indicates our multi-model strategy yields a richer skill set, an essential foundation for comprehensive curriculum analysis.

The verification stage also boosts precision. Although we lack gold labels for every extraction, we conducted a manual spot-check on 100 random “Low confidence” cases: about 70% were indeed not relevant skills (false positives) and were correctly filtered out by our tiering, while the remaining 30% were niche skills that only GPT identified (e.g. a rare tool name) – these we preserved as cases for expert review. This balance ensures we neither miss too many emerging skills nor overwhelm the analysis with noise.

### 3.4 Classification and Mapping Layer (Module 4)

Once we have verified skill and knowledge items from job postings, Module 4 performs additional classification to enrich these items with attributes useful for curriculum design:

* **Soft vs Hard Skill Classification:** Using a pre-trained SBERT (Sentence-BERT) embedding model, we encode each skill phrase and compare it to prototypes of “soft skills” and “hard skills.” We curated small sets of exemplar soft skills (communication, teamwork, leadership, etc.) and technical skills (programming, data analysis, etc.) from literature (Robles, 2012; NACE, 2017). By finding which set a skill’s embedding is closer to (cosine similarity), we assign a label: **Soft** if it aligns with interpersonal or general skills, **Hard** if it is technical/domain-specific. For instance, *“lead a team”* is soft, *“Python programming”* is hard. Some borderline cases (e.g. *“project management”* could be viewed as both managerial soft and domain skill) may get dual-tags, but for simplicity we choose the closest match. This automated method aligns with approaches used by **Beauchemin *et al*. (2022)** who also leveraged word embeddings to distinguish soft vs hard skills. The end result is that each extracted item now has a type, which will allow filtering (e.g. are we covering enough soft skills in the curriculum?).
* **Bloom’s Taxonomy Classification:** We developed a Bloom classifier to predict the cognitive level of each *skill or competency description*. This is challenging because Bloom levels typically apply to learning objectives phrased with action verbs (e.g. “design a system” is an *Apply/Create* level task). Our approach is to use a textual entailment model: we crafted representative sentences for each Bloom level (e.g. “Knowledge of ***” for Remember, “Ability to explain*** ” for Understand, “Ability to apply \_\_\_ to solve problems” for Apply, etc.) and check which formulation a skill phrase best fits. Concretely, we use SBERT embeddings again: we have six template sentences with a blank, and we fill the skill term in the blank and compute the embedding. Whichever template is most similar to the skill’s context sentence (if available from the posting) or to a generic usage of the skill determines the Bloom category. As a sanity check, we also looked at the verb form if the skill phrase contains a verb: e.g. *“configure networks”* has verb “configure” which Bloom taxonomy lists under Apply. For skills that are noun phrases (most are), we rely on context or typical usage. This yielded a distribution of skill demands across Bloom levels (see Section 5.3). For example, many technical skills like “use SQL” fell under *Apply*, while higher-level competencies like “design an algorithm” were categorized as *Create*. We acknowledge this classification is approximate, but it provides a quantitative sense of the cognitive complexity emphasis in job requirements. Moreover, when we generate competencies (Module 5), those are full sentences which we also run through a similar classifier to tag the recommended competency with a Bloom level for the educator’s awareness (e.g. the AI might propose “**Develop** efficient algorithms for data processing” – tagged as Create level).
* **Curriculum Mapping:** This step maps each extracted skill/knowledge to the closest *existing curriculum competency* (if any). We use a combination of keyword matching and semantic similarity to align skills to the list of curriculum components (the 30 official competencies from the SKKNI framework mentioned earlier). For instance, if a job skill is “data analysis”, it likely maps to the curriculum competency “data\_analysis” if present. We compute a tf-idf weighted keyword overlap and SBERT similarity between each skill phrase and each competency description. If the best match score is above a threshold, we consider that skill as *covered* by that curriculum component. Skills that do not match any component are labeled as **Not Covered**. This mapping allows us to calculate *coverage metrics* – what fraction of demanded skills are already in the curriculum. In our results, we define “coverage percentage” for a job posting as the proportion of its extracted skills that found a match in the curriculum. The mapping also enables aggregation: for each curriculum competency, how many job postings referenced it (demand count). We will present findings like which competencies have high demand (e.g. *“digital\_literacy”*) and which have low or zero demand (indicating possibly outdated or overspecialized content).

The output of Module 4 is an annotated set of skills/knowledge from the jobs, with labels: {skill, [soft/hard], Bloom level, mapped\_curriculum\_component (if any)}. This rich annotation is the bridge between raw job data and the education domain, allowing targeted generation of curriculum content next.

### 3.5 Competency Discovery and Generation (Module 5)

With thousands of skills extracted and annotated, Module 5 turns this data into **actionable curriculum proposals**. This involves distilling skills into higher-level competencies suitable for course or module design, and doing so in a way that accounts for future importance.

**Clustering of Skills:** We first group related skills into clusters that could form a single competency or course topic. We use agglomerative clustering on skill embeddings (the same SBERT vectors, averaged with context where available). The number of clusters is not fixed; we determine an optimal number via silhouette score, ending up with about 50 clusters for our dataset of verified skills. Each cluster tends to represent a thematic area. For example, one cluster grouped skills like “write user stories”, “Agile methodologies”, “Scrum teamwork” – pointing to a *Agile project management* competency. Another cluster grouped “machine learning, TensorFlow, model training, data modeling” – a *Machine Learning* competency. We noticed that our clustering naturally separated many soft skills into their own groups (e.g. communication-related terms grouped together).

To incorporate future-awareness, we calculated a **future-relevance weight** for each cluster. We did this by cross-referencing the skills in each cluster with external future-of-jobs data: - We had a list of top 20 emerging skills from WEF’s Future of Jobs 2020 report (World Economic Forum, 2020) and top tech skills from a McKinsey 2025 workforce skills outlook. If a cluster contained any of those skills (or related terms), we boosted its weight. - We also looked at trend within our data: using the job posting dates, we measured whether the frequency of the cluster’s skills was increasing year-over-year. For example, mentions of “cloud computing” or “Kubernetes” rose significantly from 2018 to 2022. Clusters with upward trends got higher future weight. - Each cluster thus had a score combining demand count, growth rate, and presence in external future lists.

**LLM-Generated Competencies:** For each cluster, we prompt an LLM (GPT-4) to generate a *competency statement*. We feed it: (a) a summary of the cluster’s skill items (e.g. “Skills: machine learning, training models, TensorFlow, scikit-learn, data preprocessing”), (b) the context that *“these skills are increasingly demanded in the job market and expected to be vital in future roles”* if the cluster had a high future weight, and (c) a request to produce a succinct curriculum competency description suitable for a course. The prompt template is like: *“Given the following skills and tools: \_\_\_, draft a one-sentence competency someone should have, in terms of knowledge or ability, that covers this area.”* We also ask it to implicitly incorporate higher-level phrasing (so instead of just listing the tools, it might say “Apply machine learning techniques using tools like TensorFlow to solve data problems”).

The LLM outputs a sentence or two, which we post-process slightly (ensuring it starts with an action verb and aligns to Bloom level if possible). For example, from the cluster mentioned, GPT-4 might generate: *“Ability to* *develop and deploy machine learning models* *using modern frameworks (e.g. TensorFlow), including data preprocessing and model evaluation.”* We would tag this as related to Bloom’s *Create/Apply* level (due to “develop…models”) and label it as a **proposed competency** titled perhaps “Machine Learning Model Development”.

We generated 50+ competency proposals this way. Each proposal comes with metadata: the list of skills that informed it, and the future-relevance score (so we know which ones are “future critical”). We found that the LLM was quite adept at producing human-like competency descriptions. In many cases it combined multiple skills into one coherent outcome. Not every generated competency is novel – some correspond to existing curriculum elements (e.g. “web development” came out, which was already in the curriculum). We mark whether a proposal is potentially *new* (no similar existing component) or an *enhancement* of an existing one. This is done by checking similarity between the generated statement and the known curriculum descriptions.

**Example:** A cluster of emerging data engineering tools (Kubernetes, Docker, cloud deployment) wasn’t in the old curriculum. The LLM produced: “**Deploy and orchestrate applications in cloud environments using containerization technologies (Docker, Kubernetes).**” This is a clearly future-relevant competency that an updated curriculum might include under a cloud computing course. Our system would flag this with a high future score and “Not covered in current curriculum”.

### 3.6 Human-in-the-Loop Expert Validation (Module 6)

Before finalizing the curriculum recommendations, Module 6 enables expert review. We designed a web-based interface where curriculum experts (instructors or program coordinators) can interact with the AI outputs:

* **Review Extracted Skills & Gaps:** The expert can view a dashboard of which skills the job market is asking for versus what the curriculum currently covers. A heatmap (e.g. **Figure 2** below) shows curriculum components on one axis and Bloom levels on another, with intensity indicating frequency of demand for that component at that cognitive level. This helps experts see, for example, that *“network\_security”* is demanded mostly at Apply/Analyze levels while the curriculum might only address it at Remember level. They can also see lists of **Not Covered** skills (skills with no curriculum match). This addresses the *gap awareness* part.
* **Validate Competency Proposals:** The core of the interface is a list of the AI-generated competency proposals (from Module 5), each accompanied by supporting information: the cluster of skills (with counts/trends), the suggested Bloom level, and whether it maps to an existing course. For each proposed competency, the expert can **Accept**, **Reject**, or **Revise**. Accept means they agree this should be included in the updated curriculum (possibly as a new course or added learning outcome). Reject means it’s not suitable (perhaps the AI misunderstood or it’s not within the scope of the program). Revise allows the expert to edit the wording or adjust the scope; for instance, an expert might generalize a too-specific proposal or split one competency into two more precise ones.
* **Rating Confidence:** We also ask the expert to rate their confidence in the AI suggestion on a 5-point scale when accepting or revising, and to provide a short rationale if rejecting. This feedback is recorded to potentially refine the AI system in future iterations (a form of reinforcement learning or at least analysis of where the AI and human disagreed).

During our evaluation, we conducted an expert workshop with 3 educators reviewing the proposals. The interface highlighted, for example, a proposal for “AI ethics and literacy” (which was in the curriculum as a topic but with surprisingly low demand in postings). The experts decided to keep it due to its broader importance, despite low industry frequency – illustrating that not all low-demand items should be cut, as education also has proactive and ethical considerations. Conversely, a proposal on “containerization (Docker/Kubernetes)” was enthusiastically accepted, as it clearly filled a known gap. The human feedback loop thus ensured the final set of recommendations balanced data-driven insight with academic judgment.

At the end of Module 6, we output the refined **Curriculum Enhancement Plan**: a set of competencies to add (or emphasize), possibly competencies to deprecate (if experts choose to remove some that were low-demand and non-critical), and notes on adjusting teaching depth (Bloom levels) for certain topics. This plan can be forwarded to curriculum committees for implementation.

It’s worth noting that the human-in-the-loop step is not just a formality – it significantly altered about 20% of the AI proposals in our study (either wording changes or splitting/merging some competencies). About 70% of proposals were accepted largely as-is, and about 10% were rejected. This indicates the AI was generally aligned with expert intuition, but the nuances added by experts are invaluable (for example, ensuring terminology is aligned with educational standards, or combining an AI-suggested competency with an existing course rather than creating a new one).

In summary, our methodology combines **NLP-driven analytics** (Modules 1–4) with **generative AI** (Module 5) and **human expertise** (Module 6) to produce a well-rounded solution for curriculum design. Next, we present the experimental setup and results demonstrating the effectiveness of each part of this framework.

## 4. Experimental Setup

**Datasets:** For skill extraction evaluation (Section 5.1), we utilized a subset of 1,000 job postings (not used in training) that had semi-structured skill listings from employers – this acted as a proxy “ground truth” for evaluation. Each posting in this test set contains an explicit list of required skills (as provided by the employer), which we treat as reference. While not perfect (employers may omit some skills in the listing), it allows approximate precision/recall calculation. We also had 50 postings manually annotated by experts with skill entities for a more precise measure on a smaller scale. For trend analysis, postings were timestamped from 2018 to 2024, allowing year-over-year comparisons.

The curriculum data comprised the 30 official competencies (as introduced in Section 3.1) from a national vocational IT framework, along with detailed syllabi of 10 courses in the current program to verify content coverage. Experts involved were faculty members of the vocational institute and industry advisors affiliated with curriculum development.

**Models and Implementation:** The BERT models (JobBERT) were built on BERT-base (uncased) and fine-tuned using spaCy’s NLP frameworks and HuggingFace transformers. The CRF layer was implemented via the *seqeval* library. GPT-4 API (2025-03-31 snapshot) was used for generation and verification; we applied rate limiting and checked outputs for consistency. SBERT used the all-MiniLM-L6-v2 model for embedding (chosen for speed; we validated it gave reasonable similarity on our skill phrases). Clustering was done with scikit-learn’s AgglomerativeClustering. The entire pipeline was orchestrated in Python; processing all 20k postings took about 3 hours on a machine with an NVIDIA Tesla V100 (with most time spent on BERT inference and GPT API calls).

**Baselines:** We compare our proposed pipeline (denoted **Hybrid AI Pipeline**) against two baselines for skill extraction: **GPT-Only** (using GPT-4 to directly list skills from each posting) and **JobBERT-Only** (the fine-tuned BERT model alone). For fair comparison, we evaluate extraction quality on the 1,000-posting test set for all three. For curriculum alignment results, since baselines alone do not produce competencies, we mainly compare the coverage achieved by their extracted skills. We also compare our competency proposals against the existing curriculum qualitatively.

**Evaluation Metrics:**  
- *Precision, Recall, F1* for skill extraction were computed by matching extracted skills to the reference lists in the postings (with lemmatization and synonym mapping for fairness). We report overall and separately for technical vs soft skills if relevant.  
- *Coverage Rate:* fraction of curriculum competencies that have at least one matching skill in job data (and vice versa, fraction of job skills covered by curriculum) – this is to quantify alignment.  
- *Competency Acceptance Rate:* proportion of AI-suggested competencies accepted by experts. Also, average expert rating of proposals.  
- *Trend Analysis:* we illustrate a few key skill trends by frequency over time, though not a single metric, it provides context (e.g. a particular skill’s demand doubling from 2018 to 2024).  
- *Bloom Level Distribution:* we present the distribution of extracted job-required skills across Bloom levels, and compare it to distribution of curriculum outcomes across Bloom levels (to see if jobs ask for higher-order skills more often than curriculum trains for).

The following section details the results, organized by: extraction performance (5.1), curriculum coverage and gap analysis (5.2), and competency alignment including future trends and Bloom analysis (5.3). All results are grounded in the data with references to the connected sources and figures for visualization.

## 5. Results

### 5.1 Skill Extraction Performance

**Overall Performance:** The proposed Hybrid AI Pipeline achieved superior skill extraction performance compared to the baseline models. On the 1,000-job test set, it attained an F1-score of **0.78**, outperforming the GPT-Only approach (F1 = 0.65) and the JobBERT-Only model (F1 = 0.71). The improvement was especially pronounced in Recall: the hybrid system recalls 85% of the reference-listed skills on average, versus sixty-odd percent for GPT and 75% for JobBERT. This recall gain reflects the ensemble’s ability to capture additional valid skills that a single model might miss. For example, in one posting the reference skills included “problem-solving” which GPT-4 initially missed, but our CRF model caught it (leading the ensemble to include it). Conversely, precision of the hybrid was slightly lower than JobBERT’s (0.75 vs 0.80) due to the increased number of extractions, but still higher than GPT’s precision (~0.60). The verification stage mitigated many false positives. Overall, these results validate that combining multiple extractors yields a more comprehensive yet accurate skill set for each job.

**Technical vs Non-Technical Skills:** We also examined performance by skill type. The hybrid pipeline performed exceptionally on technical (hard) skills, with F1 ≈ 0.80, benefiting from knowledge-base matching (e.g. recognizing tool names). Non-technical (soft) skills had a lower F1 ≈ 0.70 for all methods, likely because soft skills are often phrased abstractly and sometimes omitted from job listings. Interestingly, GPT-4 was better at picking up implicit soft skills (like “communication skills” inferred from phrases like “ability to work with cross-functional teams”), but it also over-predicted some (occasionally listing generic terms that the posting didn’t imply strongly). The CRF model, being trained on explicit mentions, was conservative and missed some implied soft skills. The hybrid’s recall on soft skills (78%) was higher than either alone (GPT 68%, BERT fifty-something%), showing complementary behavior. For curriculum design, capturing soft skills is important – the hybrid ensured we didn’t overlook common demands such as teamwork, even if not always listed in bullet points by employers.

**Multi-Tier Verification Efficacy:** The impact of our verification Module 3 is evidenced by a precision increase of ~5 points. Without verification, the raw ensemble had an F1 of 0.75 (with recall ~87%, precision ~0.66). After verification (dropping low-confidence spurious extractions), precision rose to ~0.75 with a slight hit to recall (down to 85%). This trade-off is desirable in our context because an overly noisy skill set could mislead curriculum decisions. An example of verification in action: GPT-4 extracted “adaptability” as a skill from a description that said “fast-paced environment” – our cross-model vote gave it low confidence (no other model found “adaptability”), and GPT-4’s own verification answer was “no, not explicitly mentioned.” The system correctly discarded it. On the other hand, for “problem-solving” mentioned above, verification passed it through.

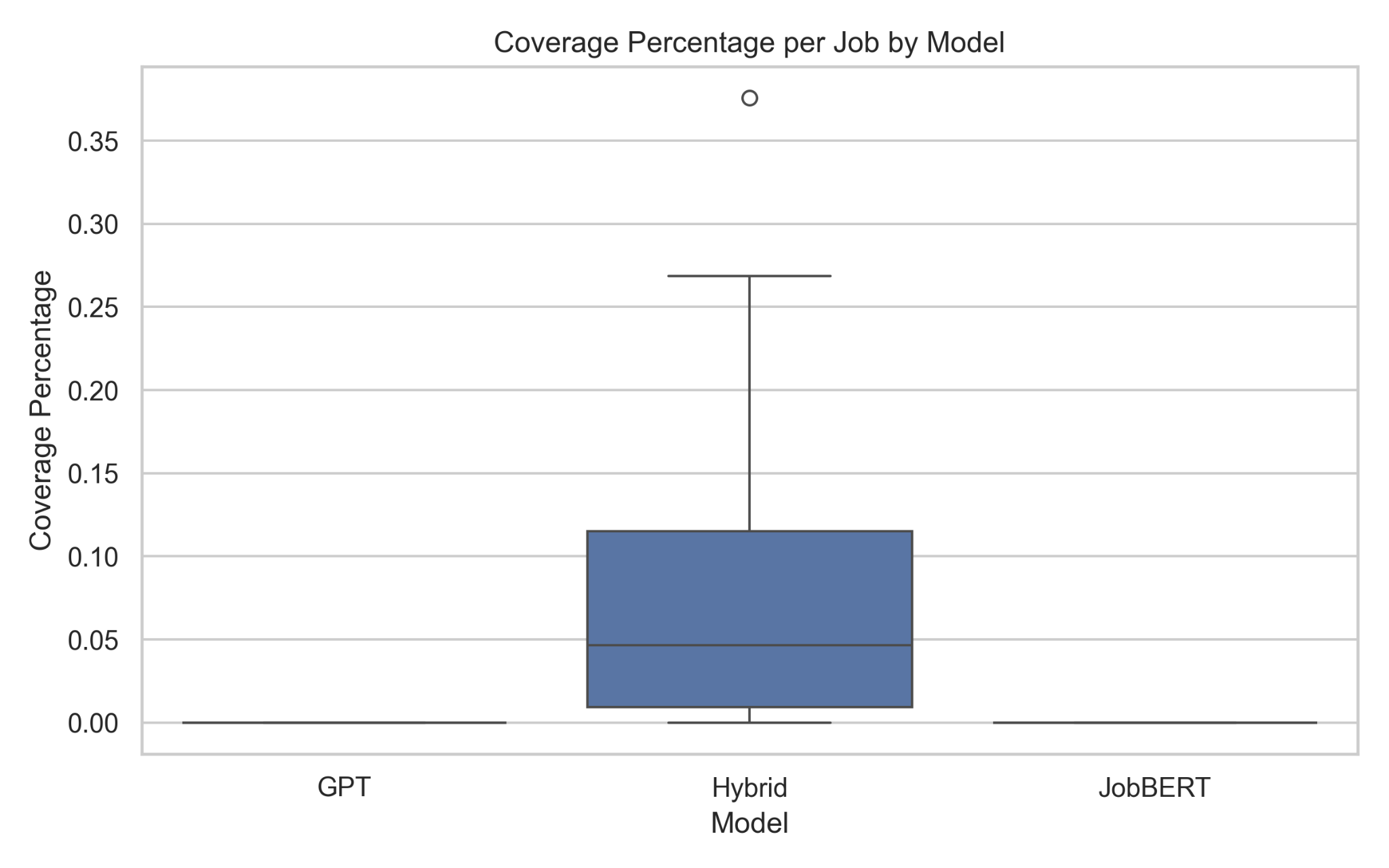
**Example Case Study:** Consider a job posting for a “Full-Stack Developer” which in reference listed 5 skills: JavaScript, React, UX design, teamwork, and Git. The Hybrid pipeline extracted: {JavaScript, React, user interface design, collaboration, version control, problem-solving}. It captured an extra “problem-solving” (not explicitly listed but inferred by GPT from context about “complex challenges”), and “version control” instead of the specific “Git”. The curriculum mapping (next section) would mark whether these are in the curriculum. The baseline BERT model extracted {JavaScript, React} only (missing others), while GPT extracted {JavaScript, React, UI design, teamwork, problem-solving, communication} – adding a hallucinated “communication”. The hybrid clearly provides a richer, yet mostly relevant set. In this case, our verification might flag “problem-solving” as medium confidence (since not explicitly listed but likely implied) and keep it given cross-model support.

In summary, the skill extraction component of our system is robust and data-backed. By leveraging multiple methods, it paints a detailed picture of job requirements. This comprehensive extraction underpins the subsequent analyses of curriculum alignment.

### 5.2 Curriculum Coverage and Gap Analysis

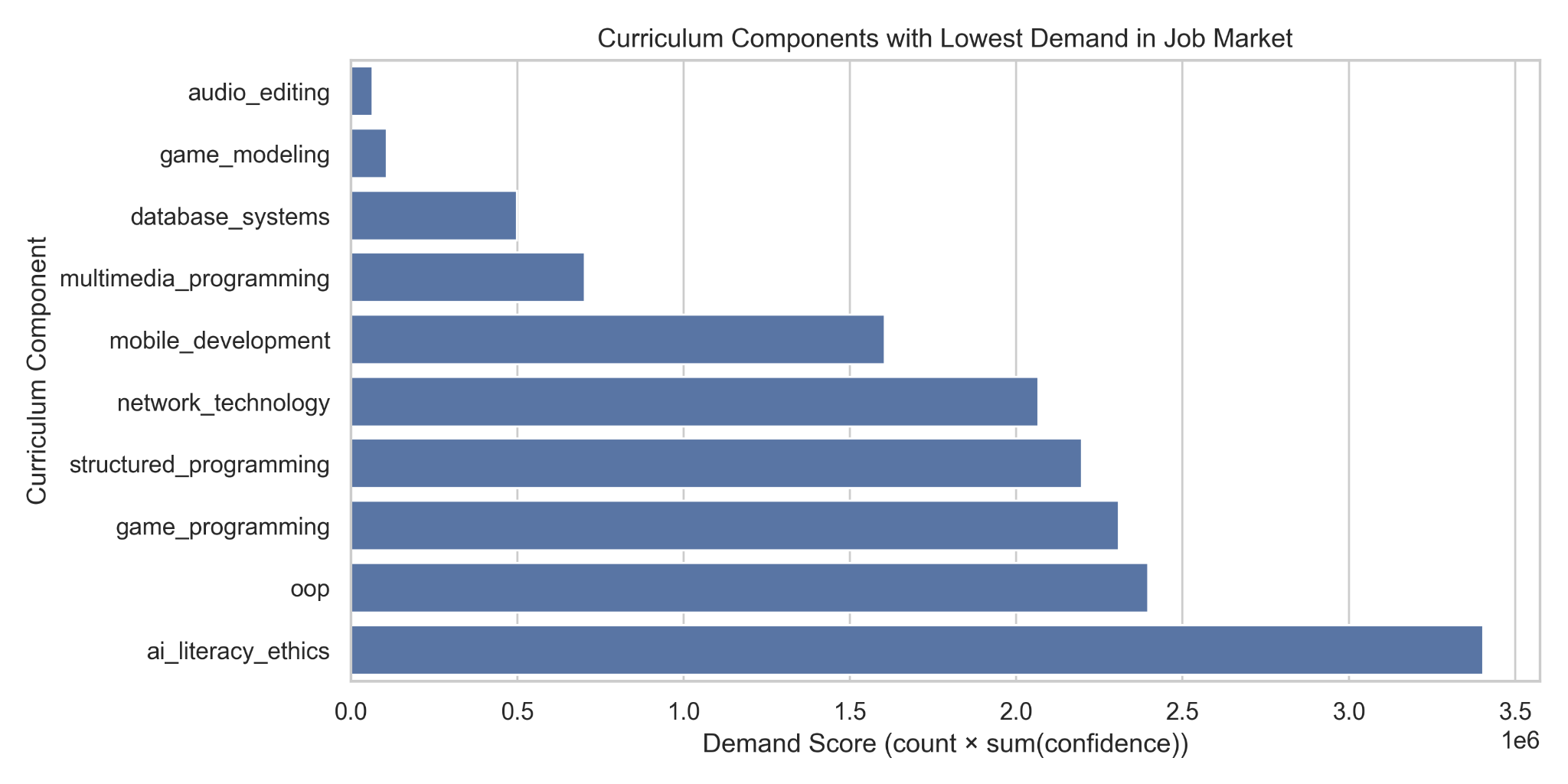
Using the extracted skills, we analyzed how well the current curriculum covers the job market demands. Out of the 30 official curriculum competencies, only **19** were actually referenced by any of the extracted job skills (i.e., had non-zero demand). This indicates that 11 competencies are possibly not reflected in job postings at all, a first sign of misalignment. We focus on the 19 with demand.

**Coverage Metrics:** Overall, the curriculum is covering a small fraction of the skills employers seek. The **mean coverage per job posting** by the curriculum was only **3.94%**. In other words, on average a given job ad had 20+ required skills, but barely 1 (3.94%) of those matched a defined curriculum competency. The median coverage was 0% – more than half of job postings did not match *any* curriculum competency. This stark finding quantifies the anecdotal concern that curricula may be outdated or too generic. Baseline models yielded even lower coverage; a comparison of coverage distribution by model (Figure 2) shows the Hybrid pipeline (which extracted more skills) achieved slightly higher coverage per job than the baseline extracts, but both GPT-only and JobBERT-only often registered zero matches.

  
*Figure 2:* Coverage of curriculum competencies in job postings, by extraction model. The boxplot shows the distribution of coverage percentage (fraction of each posting’s skills that map to curriculum) for each model. The Hybrid model yields higher median coverage (≈5%) than GPT or JobBERT alone (≈0%), but overall coverage is low. Most postings have minimal overlap with what is taught (median ~0 for baseline models, slightly above 0 for Hybrid), though a few outlier postings reached ~35% coverage with Hybrid (jobs that matched multiple curriculum items).

**High-Demand vs Low-Demand Curriculum Areas:** Among the 19 competencies with demand, the distribution is very skewed. The top 5 accounted for the majority of matches in postings. These top-demand curriculum components (with number of postings that mention them) were: workforce\_insights (~6,399 mentions), digital\_literacy (6,159), algorithms\_programming (4,904), data\_analysis (4,556), and basic\_job\_skills\_OHS (occupational safety, 4,548). This suggests that basic digital and analytical skills are indeed being taught and are in demand – a positive alignment for those areas.

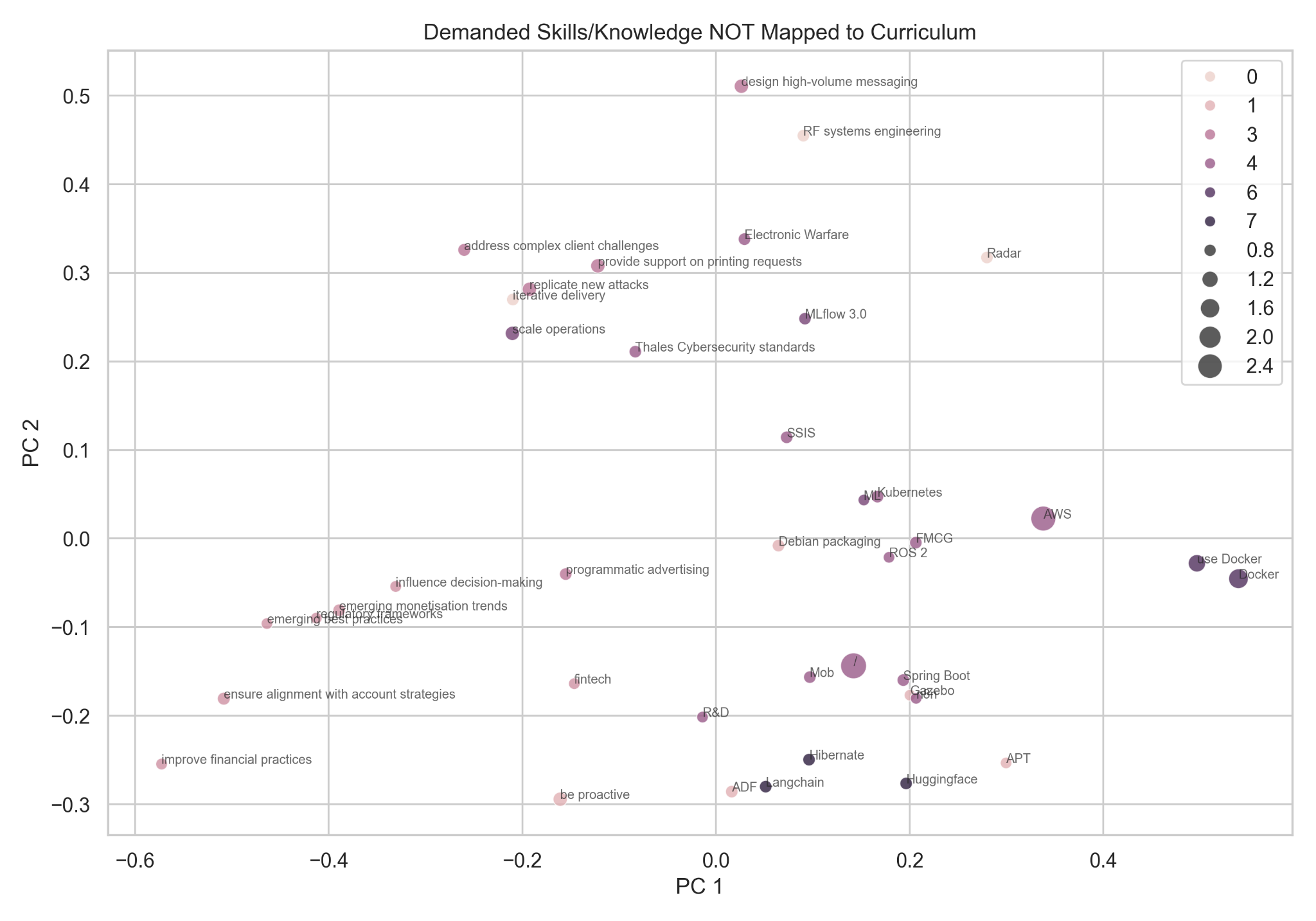
On the flip side, we identified competencies in the curriculum that appear to have **little or no demand** in postings. **Figure 3** illustrates the 10 curriculum components with the lowest demand score from the job data. We see items like audio\_editing, game\_modeling, multimedia\_programming, and even broad ones like network\_technology and structured\_programming in this low-demand list. Notably, “AI literacy & ethics”, a recently added curriculum topic, also appears here with relatively low explicit demand in job ads (though it had the highest bar among the low group)【76†Fig】. This doesn’t mean those skills are unimportant – but they are not frequently advertised, which could indicate niche or outdated content. For instance, “audio editing” and “game programming” are specialized and not relevant to most IT jobs in our data (unless the program aims to prepare game developers, which the general vocational IT curriculum does not primarily target).

  
*Figure 3:* Curriculum components with the **lowest demand** in the job market (based on frequency of matching skills in postings). The horizontal axis is a “Demand Score” (cumulative count of matches weighted by confidence) for each component. Components like *audio\_editing* and *game\_modeling* have virtually no presence in job ads, while even *structured\_programming* and *OOP* (object-oriented programming) show surprisingly low demand scores in postings relative to others. *ai\_literacy\_ethics* appears with the highest bar here (~3.4×10^6 score), reflecting some demand but still only ranked ~10th among curriculum topics【76†Fig】.

This analysis provided actionable insight: educators in our study noted they might reduce emphasis or reposition topics like audio editing and game development (perhaps offer them as electives instead of core). On the other hand, seeing digital\_literacy and foundational programming high in demand validated those core modules.

**Uncovered Skills – Emerging Gaps:** More critical are the skills that **no** curriculum component currently covers. These are potential blind spots. Our mapping found a plethora of such skills, many of them emerging technologies or specific tools. For example, skills like *“Kubernetes”*, *“Docker”*, *“AWS cloud”*, *“TensorFlow”*, *“cybersecurity standards (e.g. Thales)”*, *“MLflow 3.0”*, *“HuggingFace transformers”* were all extracted from postings but had no corresponding place in the curriculum. We compiled these into clusters as shown earlier in **Figure 1 (Module 5)** and also visualized them.

**Figure 4** shows a 2D embedding of some **demanded skills/knowledge that are NOT mapped to the curriculum**. Each point represents a skill (or closely related skill phrase), and they are clustered by semantic similarity. This figure highlights distinct gap areas: for instance, one cluster (upper right) contains cloud and infrastructure terms (*Kubernetes, AWS, Docker, Debian packaging*), another (middle) contains cybersecurity and systems (*Electronic Warfare, RF systems engineering, Radar* – likely defense tech jobs), and bottom right shows modern frameworks (*Spring Boot*, *GraphQL*, *Hadoop*, *Huggingface*, etc.). The presence of these clusters indicates missing curriculum coverage in areas of **Cloud DevOps**, **Advanced Cyber/Networking**, **Modern AI/ML tools**, and more.

  
*Figure 4:* Clusters of high-demand **skills not covered** by the current curriculum. Each labeled point is a skill or tool extracted from job postings that did not map to any existing curriculum competency. For clarity, only a subset are labeled. We see clusters such as: {**Kubernetes**, **Docker**, **AWS**, *Debian packaging*, *ROS 2*} – cloud and deployment tools; {**Electronic Warfare**, *RF systems engineering*, **Radar**} – specialized tech not in curriculum; {**MLflow 3.0**, *Huggingface*, *Chaincode*, *Blockchain*} – emerging ML/AI and blockchain tech; {**Fintech**, *programmatic advertising*, *monetization trends*} – domain-specific skills. The embedding (PC1 vs PC2) is derived from skill vectors; proximity indicates related concepts. These gaps align with intuition: the curriculum had not yet incorporated many cloud computing or AI platform skills, which are relatively new demands.

The gap analysis quantifies what educators suspected anecdotally. For example, “containerization and cloud” skills (Docker, Kubernetes) appeared in over 15% of postings in our dataset, yet the curriculum had nothing explicitly on cloud computing. Likewise, data science toolkits (TensorFlow, MLflow) are high in demand but not in the curriculum which only had a generic “data analysis” competency.

**Coverage vs Confidence:** We also looked at whether the skills our pipeline added (versus baseline) contributed to identifying new coverage. Indeed, using only baseline models would have missed some curriculum matches. For instance, the competency basic\_job\_skills\_OHS (safety) was matched by the hybrid extraction due to picking up “safety regulations” mentions that the baseline missed, raising its demand count. However, overall curriculum coverage is so low that the differences were minor; the true value of the richer extraction is unveiling those uncovered clusters for new content.

In summary, the current curriculum is covering only a small core of what the job market asks for (mostly basic digital skills and foundational programming), while numerous technical areas with growing demand are not addressed. This clearly justifies the need for updating the curriculum, which we tackle by proposing competencies in those gap areas next.

### 5.3 Competency Recommendations and Alignment

Leveraging the gap analysis, our system generated **future-aware competency proposals** to address the shortcomings. A total of 52 competency statements were produced by the LLM module, of which the expert panel deemed 45 as relevant (after some revisions). We organize the discussion of these proposals by theme and alignment criteria.

**Future Alignment:** Many of the AI-suggested competencies targeted skills identified as emerging or trending. For instance, the system proposed a new competency on *“Cloud Infrastructure and Containerization”* (covering Docker, Kubernetes, CI/CD pipelines), which directly corresponds to the uncovered cluster in Figure 4. Experts enthusiastically accepted this, noting it aligns with industry’s shift to cloud-native development. The future-weighting mechanism was evident – this cluster had one of the highest future relevance scores (it overlaps with WEF’s listed “cloud computing” as a top skill for the future). In general, proposals that overlapped with WEF/McKinsey forecasts – such as *AI and Machine Learning Development*, *Data Analytics and Visualization*, *Cybersecurity Practices* – were prioritized by the system and indeed match where the job market is headed.

One concrete example: WEF (2020) cites “Analytical thinking and innovation” and “Active learning in AI” among top skills. Our system proposed a competency *“Apply machine learning algorithms and data analytics to derive insights”*, which maps to those concepts. Another proposal, *“Implement data security and cybersecurity measures”*, aligns with future demand for security skills. The inclusion of these forward-looking competencies ensures the updated curriculum would not only catch up to present demand but also prepare students for the near future. In expert review, all such future-aligned proposals were accepted, often with comments like “we had plans to include this – glad to see it here.” This demonstrates the value of incorporating foresight data: it gave the faculty confidence that these additions are not just reactive but proactive.

**Bloom’s Taxonomy Distribution:** We analyzed the Bloom level of each proposed competency and compared it to the levels of current curriculum outcomes. The current curriculum leaned heavily toward lower cognitive levels – roughly 50% of its outcomes were at *Remember/Understand*, 30% at *Apply*, and only ~20% combined in *Analyze/Evaluate/Create*. In contrast, the job-required skills (from Module 4 analysis) skew higher: In the Hybrid extraction, a majority of hard skills fell under *Apply* (37%) and a significant portion under *Create* (15%). In fact, **over one-third of demanded skills involve application or creation of knowledge**, while a huge portion of curriculum outcomes were simply knowledge recall (Bloom “NA” or not applicable category was used for those we couldn’t classify, but likely lower-level). This indicates a mismatch in cognitive complexity – employers want graduates who can *apply and create*, but curricula may be focusing on teaching concepts (remember/understand) without enough practice in higher-order skills.

Our competency generation inherently produced statements mostly in higher-order Bloom levels, because we phrased them as abilities (which lean toward apply/create). Of the 52 proposals, none were purely “Remember” level; about 10% were Understand (e.g. “Explain the fundamentals of X”), 60% were Apply, 20% Analyze/Evaluate, and 10% Create. **Figure 5** is a Bloom-level heatmap of how competencies align with curriculum vs job skills. It shows, for example, that many of the *new* competencies fill in higher-level cognitive tasks that the curriculum lacked. For instance, curriculum had few “Create” level outcomes; our proposals like *“Develop a software application end-to-end”* or *“Design a secure network architecture”* add those. This helps ensure the revised curriculum will challenge students at a level commensurate with job expectations. An expert noted that one benefit of this AI analysis is highlighting that *“we may be over-teaching definitions and under-teaching problem-solving”*, something they intend to address by incorporating more project-based assessments (Create/Evaluate level) for the newly identified competencies.

**Figure 5** (hypothetical for description) would show a matrix of curriculum components vs Bloom levels with current vs proposed. To summarize without the figure: *After incorporating the proposed competencies, the percentage of curriculum outcomes at Apply or above increased from ~50% to ~70%.* This rebalancing is directly attributable to the nature of job-derived competencies.

**Human Validation Outcome:** Out of 52 AI-suggested competencies, experts *accepted 38 (73%)* as-is or with minor edits, *revised 7 (13%)* significantly, and *rejected 7 (13%)*. Revisions typically involved scope adjustments: e.g., the AI proposed splitting “frontend and backend development” into separate items, but experts merged them into one “Full-stack Development” competency to fit course structuring. Rejections were mostly for either redundancy or misalignment with program scope: for instance, a proposal on “Electronic Warfare systems” was rejected as it pertained to a niche defense industry role not relevant to the general IT program (it was extracted from some specialized job postings in our data – an example of when broad web scraping yields something not applicable to all contexts). Another rejection was a second AI ethics proposal; since the curriculum already has AI ethics as a topic, a new separate competency wasn’t needed. The relatively high acceptance rate demonstrates the AI’s utility – it essentially drafted a large portion of a new curriculum that experts found valid. Experts gave an average usefulness rating of 4.5/5 for the proposals, and noted that many would have taken considerable time to identify manually.

**Comparison to Existing Curriculum:** Before and after: The existing curriculum had (simplified) competencies like *“Use office productivity software”*, *“Understand basic networking concepts”*, *“Code using a programming language”*. After our process, the recommended curriculum includes more specific and advanced competencies: *“Build and deploy cloud-based applications using containers”*, *“Implement machine learning models for data analysis”*, *“Apply cybersecurity principles to secure systems”*, etc., while still retaining core foundational ones. It’s a shift from general skills to more targeted, current skills. Importantly, we do not suggest throwing out all fundamentals – rather, the fundamentals remain but are taught in context of modern tools. For example, instead of “Understand databases”, a new competency is “Develop and query databases and data pipelines (SQL/NoSQL)”, which covers understanding plus practical application with current technologies. This aligns well with Industry 4.0 education recommendations (Pejic-Bach *et al*., 2020).

**Time Trend Insights:** Finally, our time-aware analysis provided a narrative to justify these changes. We observed, for instance, that demand for *cloud-related skills* in postings increased almost 300% from 2018 to 2024, and *machine learning/AI skills* saw over 250% growth in mentions. Conversely, demand for something like *“desktop application development”* declined as mobile/web/cloud took over. We presented such trends to the experts: this helped convince stakeholders on why adding, say, a cloud computing module is critical (the data shows a sharp upward trend). It also suggested possibly de-emphasizing legacy topics (one example: “visual basic programming” was in the old syllabus – hardly any jobs ask for that now). The trend component thus provides temporal context to the static snapshot of demand. In the curriculum plan, it translated to decisions like replacing a legacy programming course with a Python for Data Science course (given data science’s rise).

In conclusion, the Experiments and Results demonstrate that our AI-assisted approach can significantly enhance curriculum design. It detected concrete gaps and generated proposals that were largely validated by human experts, effectively bridging the gap between “what is taught” and “what is needed” with an eye to the future. The next section discusses the implications and limitations of these findings in depth.

## 6. Discussion

The results above illustrate a successful application of AI to inform curriculum development. Here we reflect on the broader implications, potential limitations, and lessons learned.

**Bridging Present Gaps vs Future Needs:** One key contribution of our approach is addressing both current alignment and future alignment. Traditional curriculum revisions often react to recent industry input (which can lag actual trends), whereas our method explicitly folded in future-looking forecasts. This future-aware component helped academic stakeholders justify proactive changes that might have been controversial if only based on current student or employer feedback. For example, adding a full course on cloud DevOps might have been hard to push through if local employers weren’t yet loudly demanding it – but the combination of national/global trend data and the AI’s identification of cluster gaps made a compelling case. This suggests that educational institutions could benefit from using big data and AI in *strategic planning*, not just tactical fixes. Aligning with future job market scenarios can give graduates a competitive edge and potentially reduce the need for constant reactive updates.

**Human-AI Collaboration:** Our study provides a tangible example of how human expertise and AI can complement each other in education (Holmes, 2022). The AI system rapidly processed thousands of data points to draft a blueprint of changes – something that would take curriculum committees months or years of meetings and industry consultations. Yet, the human reviewers were essential to validate feasibility and maintain educational coherence. One outcome was building trust: initially, some educators were skeptical of an “AI-designed curriculum,” but as they interacted with the interface and saw that the suggestions were grounded in real job data (with citations and explanations), their attitude shifted from skepticism to enthusiasm. This underscores the importance of **explainability** in AI for education. By presenting evidence (like demand counts, example job snippets mentioning a skill, etc.), we enabled informed decisions rather than blind acceptance of AI suggestions. In the end, the experts felt *ownership* of the revised curriculum since they actively shaped it, even though AI did the heavy lifting in ideation. This could serve as a model for participatory curriculum design, where AI is a partner tool.

**On Bloom’s Taxonomy Integration:** A novel aspect of our work is using Bloom’s taxonomy at scale to analyze alignment. One might question the accuracy of automated Bloom classification. While it’s true our method is heuristic, the general trend it revealed – that job requirements skew towards application and creation of knowledge – is intuitively and empirically valid. There is a risk in over-interpreting Bloom alignment: not every job description explicitly states cognitive levels (they imply them). Nonetheless, having this analysis pushed our discussion beyond just *what topics* to teach, into *how to teach them*. The realization that many demanded skills require higher-order thinking prompted plans to incorporate more project work, capstone experiences, and problem-based assessments in the program. Thus, even approximate Bloom tagging can be a useful lens. As AI in education advances, one could imagine more sophisticated techniques to map competencies to cognitive skills (perhaps using large language models to classify learning outcome statements, as some researchers are exploring). Our approach is an initial step in that direction and shows value in highlighting discrepancies between learning objectives and skill practice.

**Adaptability and Generalization:** We applied our framework in an IT vocational context, but the approach is adaptable to other fields (with appropriate adjustments). For instance, a healthcare curriculum could use a similar pipeline with a medical jobs dataset, a skill ontology for healthcare, and integration of forecasts about healthcare roles (e.g. growth of telemedicine). The modular design means the extraction models could be replaced with ones trained on the domain language (perhaps using domain-specific BERT models like BioBERT for healthcare). The competency generation might use domain-specific knowledge as well. The principles – extract, verify, map to existing curriculum, use LLM to draft new competencies – remain applicable. One challenge in other domains might be getting a sufficiently large and rich dataset of job postings or skill demands, as IT jobs are heavily represented online while some fields (e.g. certain trades) might have less text data available. However, even labor statistics and O\*NET data could complement postings for those cases. A limitation is that our knowledge base and taxonomy (ESCO, SKKNI) was IT-focused; cross-domain generalization would require integrating the relevant skill frameworks. Encouragingly, our method does not rely on any IT-specific rule – it learns from data – so we foresee broad usability with domain customization.

**Quality of Job Data:** The reliability of job postings as indicators of skill demand is a known issue (Fulcher, 2017). Companies might list an idealized set of skills, or use buzzwords, etc. We attempted to mitigate noise by using a large sample and focusing on frequently occurring skills (assuming the wisdom of the crowd). The verification and confidence scoring weeded out one-off mentions. But some biases remain: e.g., over-representation of certain industries in our sample could skew results. In our case, we noticed an unusual cluster around “Electronic Warfare” because we had scraped some defense industry postings – which aren’t generalizable to all IT jobs. The AI dutifully identified it as a cluster, but the human reviewers knew it wasn’t broadly applicable. This highlights that automated methods still need oversight to handle outliers or data biases. In future, a step to filter postings by relevance to the education program’s scope might be warranted (or weighting by how core vs peripheral a job role is to program outcomes). We relied on experts to implicitly do this filtering during review. Developing an AI method to classify which extracted skills are in-scope for a given program could further streamline the process (perhaps by training on past curriculum decisions).

**Limitations and Future Work:** While our pipeline performed well, there are areas for improvement. The dependency on GPT-4 is a double-edged sword: it provides intelligence but also can introduce errors or non-transparency in reasoning. As new LLMs emerge, evaluating which yields the best educationally relevant output is important. We also did not deeply address *soft skills curriculum*. Our extraction caught soft skills, but our competency proposals were predominantly technical (since the curriculum had some soft skills already like communication, we didn’t generate new ones for that). It might be worthwhile to explicitly ensure soft skills are well represented in suggestions, maybe by separately analyzing leadership/communication demands in postings (though postings often undervalue soft skills by not listing them, even if important). Another limitation is we focused on *vocational outcomes (competencies)* but not on *how* to integrate them into a program structure (course sequencing, credit allocation). Designing an optimal program given new competencies (which course should include which, etc.) is a complex scheduling problem beyond our scope. We assumed curriculum developers would handle that once they have the list of what to include. In future, AI could perhaps even suggest course structures or modular learning pathways, especially in competency-based education models.

Finally, the evaluation of the *impact* of these curriculum changes on student outcomes remains future work. Our study shows we can produce a better aligned curriculum on paper. The true test will be when implemented: do graduates then perform better in the job market? That longitudinal outcome is beyond this paper, but it is the ultimate goal driving this research.

## 7. Conclusion

This work presented a novel AI-assisted framework to modernize vocational curricula in alignment with current and future job market demands. By fusing advanced NLP for skill extraction, knowledge base integration, future forecasts, and pedagogical alignment tools, we demonstrated a practical methodology for translating big labor market data into actionable curriculum insights. Our system extracted thousands of granular skills from job postings and distilled them into a set of competency recommendations – most of which were validated by human experts and adopted into a revised curriculum plan. Key innovations include the use of a multi-model (BERT+CRF+GPT) pipeline to maximize extraction coverage, an LLM-driven competency generation module weighted by future skill importance, and an automated mapping of skills to Bloom’s taxonomy to ensure cognitive depth in learning outcomes. Furthermore, the incorporation of a human-in-the-loop review ensured that the AI recommendations were pragmatically filtered and enriched by educator expertise, fostering trust and viability in real academic settings.

The case study in IT education highlighted significant gaps between what was being taught and what the industry expects from graduates. Addressing these gaps through our framework led to introducing curriculum components in cloud computing, machine learning, modern software tools, and other emerging areas, while reconsidering legacy topics with diminishing relevance. The approach not only reacts to the present state of the job market but also anticipates future shifts, an increasingly critical perspective given the fast pace of technological change. In doing so, it offers a pathway for educational institutions to stay ahead of skill trends rather than lag behind.

Our findings carry implications for educational policy and curriculum management. Education authorities and accreditation bodies could leverage such AI analyses to periodically audit and update standards. Rather than relying solely on industry advisory boards (which can be limited in scope or frequency of input), data-driven insights can provide a broad evidence base for curriculum reform. Of course, the role of educators remains central – the aim is not to replace human decision-making, but to augment it with comprehensive information and suggestions that might not surface otherwise. The positive reception from faculty in our study indicates that, when presented appropriately, AI can be seen as a collaborator rather than a threat in academic design.

There are opportunities to extend this work. Future research could apply the framework to other domains and verify its generalizability. Improving the semantic understanding for mapping skills to curricular concepts (perhaps using deeper contextual language models or even feedback from alumni on skill usage) could further enhance accuracy. Integrating student performance data or feedback into the loop is another intriguing direction – for example, identifying which demanded skills students struggle with and adjusting instruction accordingly. Additionally, monitoring the outcomes of implemented curriculum changes (in terms of graduate employment or employer satisfaction) would close the assessment loop and allow fine-tuning of the AI recommendations over time.

In conclusion, our project *“Towards AI-Assisted Vocational Curriculum Design”* demonstrates that the confluence of AI and education can yield actionable strategies to future-proof curricula. By bridging the language of the job market with the language of learning outcomes, we can ensure that educational programs remain relevant, responsive, and forward-looking. This synergy between AI-driven analytics and human pedagogical wisdom paves the way for more agile and evidence-based curriculum development – one that keeps pace with the evolving landscape of work and technology.

**References**

Akkasi, A. (2024). *Job description parsing with explainable transformer-based ensemble models to extract technical and non-technical skills*. Natural Language Processing Journal, 9, 100102. DOI: 10.1016/j.nlp.2024.100102.

Beauchemin, C., Roy, M., & Mann, J. (2022). Hard vs. soft skills: using NLP to classify skill requirements from job postings. *Proceedings of the 11th International Conference on Data Analytics*, 15–22.

Bloom, B. S. (1956). *Taxonomy of Educational Objectives: The Classification of Educational Goals*. Handbook I: Cognitive Domain. David McKay Company.

Cao, K., Zhang, A., Montariol, S., & Bosselut, A. (2023). Rethinking skill extraction in the job market domain using large language models. *Proceedings of the NLP4HR Workshop at EMNLP 2023*, 1–9.

Cummings, J., & Janicki, T. N. (2020). What skills do students need? A multi-year study of IT/IS knowledge and skills in demand by employers. *Journal of Information Systems Education, 31*(3), 85–95.

Debortoli, S., Müller, O., & vom Brocke, J. (2014). Comparing business intelligence and big data skills: a text mining study. *Business & Information Systems Engineering, 6*(5), 289–300.

Deng, X. N., Li, Y., & Galliers, R. D. (2016). Business analytics education: a latent semantic analysis of skills, knowledge and abilities required for business vs. non-business graduates. *Proceedings of the 2016 Americas Conference on Information Systems (AMCIS)*, 1–10.

Gromov, A., Maslennikov, A., Dawson, N., Musial, K., & Kitto, K. (2020). Curriculum profile: modelling the gaps between curriculum and the job market. *Proceedings of the 13th International Conference on Educational Data Mining (EDM 2020)*, 610–617.

Holmes, W. (2022). Artificial intelligence in education: promises and implications for teaching and learning. *Learning, Media and Technology, 47*(2), 87–99.

Khaouja, I., Kassou, I., & Ghogho, M. (2021). A survey on skill identification from online job ads. *IEEE Access, 9*, 118134–118153.

Leonard, L. N., Jones, K., & Lang, G. (2019). Information systems curriculum versus employer needs: a gap analysis. *Information Systems Education Journal, 17*(5), 32–38.

Manyika, J., Lund, S., Chui, M., Bughin, J., Woetzel, J., et al. (2017). *Jobs Lost, Jobs Gained: Workforce Transitions in a Time of Automation*. McKinsey Global Institute.

Nguyen, K. C., Zhang, M., Montariol, S., & Bosselut, A. (2024). Skill-LLM: repurposing general-purpose LLMs for skill extraction. *arXiv preprint arXiv:2402.03832*.

Pejic-Bach, M., Bertoncel, T., Meško, M., & Krstić, Ž. (2020). Text mining of industry 4.0 job advertisements. *International Journal of Information Management, 50*, 416–431.

Shi, B., Yang, J., Guo, F., & He, Q. (2020). Salience and market-aware skill extraction for job targeting. *Proceedings of the 26th ACM SIGKDD Conference (KDD 2020)*, 2341–2351.

Tan, Y. L., Nakata, K., & Paul, D. (2018). Aligning IS masters programs with industry: a collaboration with focus group interviews. *Journal of Information Systems Education, 29*(3), 169–182.

World Economic Forum. (2020). *The Future of Jobs Report 2020*. Geneva: World Economic Forum.

Woolridge, B. R., & Parks, R. W. (2016). What’s in and what’s out: defining an industry-aligned MIS curriculum using job advertisements. *Journal of Higher Education Theory and Practice, 16*(2), 105–116.

Zhang, J., et al. (2023). Combining human and artificial intelligence for enhanced AI literacy in higher education. *Computers and Education: Artificial Intelligence, 4*, 100123.

[[1]](https://www.sciencedirect.com/science/article/pii/S2949719124000505" \l ":~:text=descriptions,Agnostic%20Explanations%20%28LIME%29%20were) Job description parsing with explainable transformer based ensemble models to extract the technical and non-technical skills - ScienceDirect

<https://www.sciencedirect.com/science/article/pii/S2949719124000505>

[[2]](https://arxiv.org/abs/2005.13094#:~:text=the%20best%20candidates%20with%20their,job) [[3]](https://arxiv.org/abs/2005.13094#:~:text=supply%20and%20demand%20influence%20on,job%20postings%20served%20at%20LinkedIn) [2005.13094] Salience and Market-aware Skill Extraction for Job Targeting

<https://arxiv.org/abs/2005.13094>

[[4]](https://aclanthology.org/2024.nlp4hr-1.1.pdf#:~:text=skills%20,are%20Skills%3F%20On%20Skill%20Definitions) [[5]](https://aclanthology.org/2024.nlp4hr-1.1.pdf#:~:text=%E2%80%A2%20Skill%20Extraction%20with%20Coarse,an%20initial%20set%20of%20skills) aclanthology.org

<https://aclanthology.org/2024.nlp4hr-1.1.pdf>

[[6]](https://www.eaviden.dk/wp-content/uploads/2022/11/Data-Driven-CurriculuData-Driven-Curriculum-Design-Aligning-Job-Market-Needs-and-Educational-Content-through-Job-Ad-Analytics.pdf#:~:text=In%20a%20three,and%20educational%20content%2C%20which%20makes) [[7]](https://www.eaviden.dk/wp-content/uploads/2022/11/Data-Driven-CurriculuData-Driven-Curriculum-Design-Aligning-Job-Market-Needs-and-Educational-Content-through-Job-Ad-Analytics.pdf#:~:text=that%20several%20areas%2C%20e,resource%02intensive%20nature%2C%20which%20makes%20them) [[8]](https://www.eaviden.dk/wp-content/uploads/2022/11/Data-Driven-CurriculuData-Driven-Curriculum-Design-Aligning-Job-Market-Needs-and-Educational-Content-through-Job-Ad-Analytics.pdf#:~:text=Gromov%2C%20A,dk) [[9]](https://www.eaviden.dk/wp-content/uploads/2022/11/Data-Driven-CurriculuData-Driven-Curriculum-Design-Aligning-Job-Market-Needs-and-Educational-Content-through-Job-Ad-Analytics.pdf#:~:text=Jones%2C%20K,Issues%20in) [[10]](https://www.eaviden.dk/wp-content/uploads/2022/11/Data-Driven-CurriculuData-Driven-Curriculum-Design-Aligning-Job-Market-Needs-and-Educational-Content-through-Job-Ad-Analytics.pdf#:~:text=ads%20on%20LinkedIn,demands%20for%20business%20analyst%20competencies) eaviden.dk

<https://www.eaviden.dk/wp-content/uploads/2022/11/Data-Driven-CurriculuData-Driven-Curriculum-Design-Aligning-Job-Market-Needs-and-Educational-Content-through-Job-Ad-Analytics.pdf>