ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Innovation and Practice in the Talent Training System for University Big Data Programs from the Perspective of Deep Industry–Education Integration

Rongping Wang[1] *and Cuicui Chen*[2,*]

[1] School of Computer engineering, Guangzhou Huali College, Guangzhou 511325, Guangdong, China
[2] School of Health, Guangzhou Huali Vocational College of Science and Technology, Guangzhou 511325, Guangdong, China

SUMMARY: *Proposing solutions to the lack of collaboration between coursework and workplace needs, incomplete chains of practical training, and inadequate industry-academia partnership in training big data professionals at universities, this research paper develops a talent development system on the lens of deep industry-education integration. According to job-related corpus mining, competency weighting modeling, three-way mapping of course content, project tasks, competency indicators, process data collection platform practical teaching platform, and multi-source evaluation integration approaches, a cycle of implementation pathway has been developed that encompasses the design of training objectives, organization of tasks, platform support, and evaluation of quality. A comparative validation study was carried out on a sample of two grade levels of a university program on big data. The findings suggest that the designed system is more effective than conventional training models in terms of the time of task completion, passing the module tests, error rollback rate, collaborative interactions, and overall training quality indicators. In addition, students showed greater improvements in areas like engineering implementation, job fit, data governance and collaborative communication. The research shows that this system has the potential to convert job needs of the enterprise into measurable, implementable and trackable units of instruction and increase correspondence between course delivery and actual task chains in the job environment and give a viable technical route to maximize talent development models in university big data programs.*

KEYWORDS: *Industry-education integration; Big Data major; Talent development system; Curriculum-task mapping*

1 Introduction

In the context of the expanding digital economy and information-guided decision-making, the talent development models of university big data programs are shifting towards competency-based models as opposed to knowledge-based models. Current studies mainly discuss the talent development directions analyzing job requirements and improving the curriculum organization. Investigating the disconnect between higher education training and the market needs, Han et al. have indicated that big data specialists should have cross-disciplinary comprehensive skills, such as multidimensional skills (data processing, algorithmic modeling, business understanding,

*907484996@qq.com
https://doi.org/10.65102/is2026811 Published: 30 April, 2026

etc.) and have stressed that the curriculum system should be reorganized in the competency-oriented direction [1]. Liang et al. also suggested that, when it comes to big data and artificial intelligence, the higher education must restructure the curriculum system based on interdisciplinary integration and problem-driven models and help students acquire systematic cognitive skills in complex problem situations [2]. Zhang, taking the same problem with the view of industry-education integration, searched a collaborative training model of the example of digital media major between universities and enterprises, even so his research was firmly based on the macro-level design of pathways, but he did not describe the process of competency development in detail [3]. Ni suggested a model of integrated training of 4 parts: Curriculum-Competitions-Certification-Innovation-Application, which aimed at developing the practical skills of students with the help of multi-stage cooperation; his system of evaluation focused on outcome indicators and not on the process data [4]. Ma, however, proposed deep learning and data mining techniques to create a talent development system, accented on the optimization of teaching based on algorithms, but did not critically consider the mechanism of embedding real-world enterprise tasks [5]. Cuiling et al. presented a big data talent development model, which is an industry-education integration of the Information Management and Information Systems major, and observed that enterprise involvement is a major factor in developing practical skills, although their study was not quantitatively validated [6]. Miao et al. employed the models of artificial intelligence and backpropagation neural networks to determine the effects of industry-education integration on vocational education and found out that the model could lead to improvement of the competency attainment by students, but the interpretability and consistency of the model with real teaching practices are still lacking [7]. Duan suggested a competency modelling approach of creating a pathway of talent development among composite professionals, focusing on the combination of engineering and business competencies, but a systematic mapping of the mechanisms on the curriculum and task levels has not been developed yet [8]. On the whole, although the literature has examined the problems of the development of big data talent and the integration of industries and education through different prisms, three gaps still exist: firstly, there is no sophisticated mapping mechanism between the job competency requirements and curriculum framework; secondly, there is limited data collection and behavioral analysis in the process of practical instruction, which cannot be integrated into the chain of instructional performance; and thirdly, school-enterprise collaboration is mostly

In order to solve these problems, this paper develops a talent development system of big data majors that is oriented towards thorough integration of industry-education. The first step involves using job corpus mining and competency weighting techniques to create a job competency demand model that will translate the enterprise job needs into calculable competency measures. Second, a three-way mapping system between the course material, project activities, and competence indicators is implemented, which incorporates units of course knowledge into the actual enterprise task chains, so as to have a systematic balance between teaching material and job task; and further the development of a teaching practical platform and a system of multi-source data collection is developed to continuously record and extract characteristics in the behavioral data of the students in the process of task execution; and based on The findings of the research reveal that this system is effective in improving engineering implementation abilities, job suitability, as well as collaborative expression abilities, and changing the learning process by not just training skills in isolation as also by executing whole chain of tasks. The key findings of this paper are as follows: a job-demand-driven competency modelling approach has been developed, which allows to transfer the industry-education integration process to the data-driven level; a tripartite mapping mechanism between courses and project tasks was suggested, which enhances the compatibility of teaching material with job demands; a process data collection and evaluation mechanism has been developed relying on platform logs, which makes the evaluation of training quality more precise; The above study offers a practical technical way out and programmed course of action to the big data programs of universities to attain quality talent development in multifaceted industrial settings.

2 Methods

2.1 Modeling Job Competency Requirements for Big Data Majors

Competency requirements modeling is a step-by-step process consisting of four steps: job corpus extraction—competency item consolidation—weight calculation-job cluster mapping [8]. First, the job descriptions of positions like data development engineers, data analysis engineers, and machine learning engineers, and corporate project task lists have been gathered and the size of the corpus has been kept to more than 1, 200 entries. By segmenting words, eliminating noise, combining synonyms, and standardizing terminology, a query of terms like data cleaning, feature engineering, distributed computing, model deployment, and data governance was encoded into a set of competency items in a uniform manner. [9].Subsequently, a tag matrix was established based on 8 primary competency categories and 26 secondary competency categories. A minimum term frequency threshold $f_{\min} = 12$ and a job coverage threshold $r_{\min} = 0.18$ were set to exclude sporadic skills. The standardized demand weight for the i th job category regarding the j th competency is defined as:

$$W_{ij} = \frac{F_{ij} \log\left(\frac{N}{n_j + 1}\right)}{\sum_{j=1}^{m} F_{ij} \log\left(\frac{N}{n_j + 1}\right)} \tag{1}$$

where W_{ij} is the job competency weight, F_{ij} is the frequency of the competency item in the corpus of job category i , N is the total number of jobs, n_j is the number of jobs containing competency item j , and m is the total number of competency items. This formula is used to suppress high-frequency common terms and retain discriminative skills. Further, denoting the course output competency vector as C_j , the alignment of the training program with job category i is:

$$M_i = \sum_{j=1}^{m} W_{ij} C_j \tag{2}$$

where M_i represents the job match degree, and $C_j \in [0,1]$ is the coverage coefficient of the training system for competency item j [10]. During calculation, M_i is determined separately for each job cluster based on the enterprise sample to identify course reinforcement directions and the allocation ratio of project tasks.

2.2 Design of a Talent Development Framework for Industry-Education Integration

The system architecture is presented in Figure 1 and the architecture is developed in five layers, i.e., Job Requirement Layer-Competency Mapping Layer-Curriculum and Project Layer-Platform Support Layer-Evaluation and Feedback Layer [11]. The Job Requirement Layer integrates enterprise job profiles, lists of real-world tasks, and technology stack tags; the Competency Mapping Layer generates competency target vectors based on a job weight matrix and divides them into five competency domains: data collection, governance and development, modeling and analysis, deployment and operations, and business expression;The Curriculum and Projects Layer adopts a four-tier structure comprising "Core Courses + Micro-projects + Comprehensive Training + Real-world Enterprise Problems, " with a 16-week iteration cycle, biweekly task releases. and three acceptance checkpoints per task;The platform support layer deploys code repositories, data sandboxes, containerized runtime environments, and learning behavior logging modules, with container images standardized to Python 3.11, Spark 3.5, MySQL 8.0, and Docker 24; The evaluation and feedback layer integrates instructor scores, corporate mentor scores, and platform process data [12]. The system support degree is defined as:

$$S = \sum_{k=1}^{n} \alpha_k A_k \tag{3}$$

where S represents the overall support of the training system for job competencies, A_k is the achievement coefficient for the th competency domain k , α_k is the weight of that competency domain in job requirements, and n is the number of competency domains [13]. During calculation, α_k is first derived from job cluster statistics, and A_k is then jointly determined using course coverage, project completion rates, and platform behavioral metrics. Based on these results, the allocation of course hours and the proportion of industry projects are adjusted.

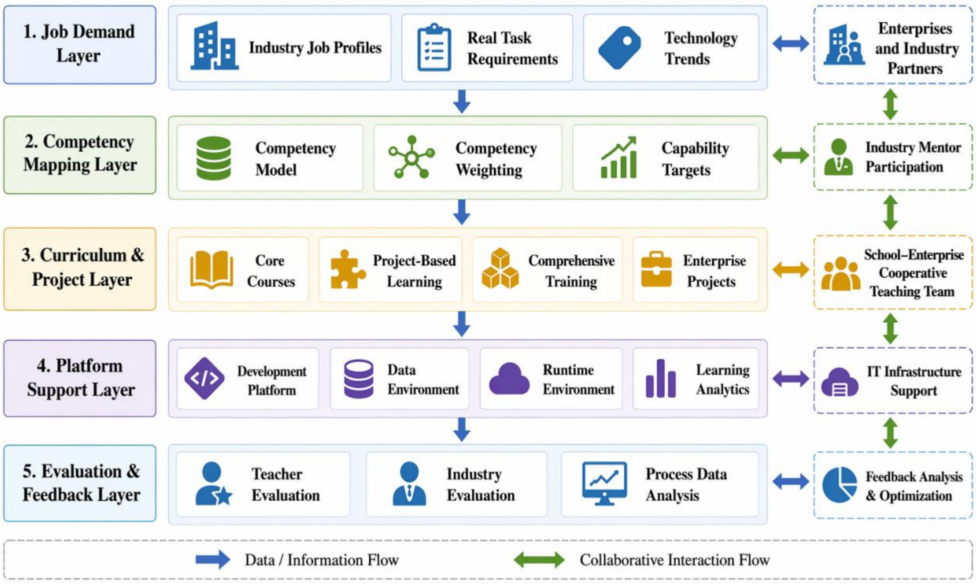

Figure 1: Overall Architecture of the Industry-Education Integration Talent Development System

2.3 Mechanism for Mapping Course Content to Project Tasks

As illustrated in Figure 2, the mapping mechanism between curriculum content, project task, and competence indicators is implemented by means of "triple coding, two-way restriction, and phase calibration".[14].First, based on job competency domains, the curriculum is broken down into knowledge units C_p . Each unit is refined to a deliverable granularity, and is annotated with four attributes: Knowledge Type, Precondition, Tool Dependency, and Actual Depth. The knowledge granularity is controlled at 2 – 4 class hours. Test units are tagged with environment tags and are uniformly bound to toolchains like Python 3.11, Pandas 2.2, Spark 3.5, MySQL 8.0, Tableau, and PowerBI.Subsequently, project tasks are extracted from real-world enterprise cases T_q and decomposed into six phases: “Data Ingestion—Cleaning and Transformation—Feature Engineering—Model Training—Deployment of Results—Visualization.” Each task is configured with input data scale, runtime environment, acceptance scripts, and output template [15].Competency metrics K_r adopt a three-tier structure: the first tier corresponds to four core competencies—data governance, analytical modeling, engineering implementation, and collaborative visualization; the second tier breaks these down into measurable items such as data quality control, API integration testing, model parameter tuning, and container deployment, while the third tier is concerned with observable variables like platform logs, code quality, test pass rate, and company mentors.There is a unique identifier for each of the test units, task nodes, and competence indicators. Once entered into a relational table, they form a three-way mapping matrix, which can be shared and used for the scheduling of courses, the work of projects, and the evaluation of the process. [16].

Mapping strength is calculated based on the extent to which a course unit supports a task node and is defined as:

$$R_{pq} = \frac{\sum_{r=1}^{m} \beta_r x_{pqr}}{\sum_{r=1}^{m} \beta_r} \tag{4}$$

where R_{pq} represents the mapping strength between course unit C_p and project task T_q ; $x_{pqr} \in \{0,1,2,3\}$ indicates the support level of the course unit for competency indicator r , with 0 denoting no support and 3 denoting strong support; β_r is the weight of competency indicator K_r ; and m is the total number of competency indicators. During application, the enterprise mentor and course director first jointly define β_r , then specify x_{pqr} based on task scripts and course experiment content. When $R_{pq} \geq 0.65$, a strong mapping relationship is established, and the unit enters the main task chain; when $0.35 \leq R_{pq} < 0.65$, it serves as a supplementary support unit;Course content below this limit will not be included in the current project configuration. In order to avoid the overlapping of courses and the discontinuity of the task, the system further verifies that each competence indicator is covered at least 3 times in a 16 week cycle, and that there is no more than 2 time interval between adjacent tasks. Based on these criteria, the system adjusts course sequencing, tiers project difficulty, and reconstructs competency attainment pathways.

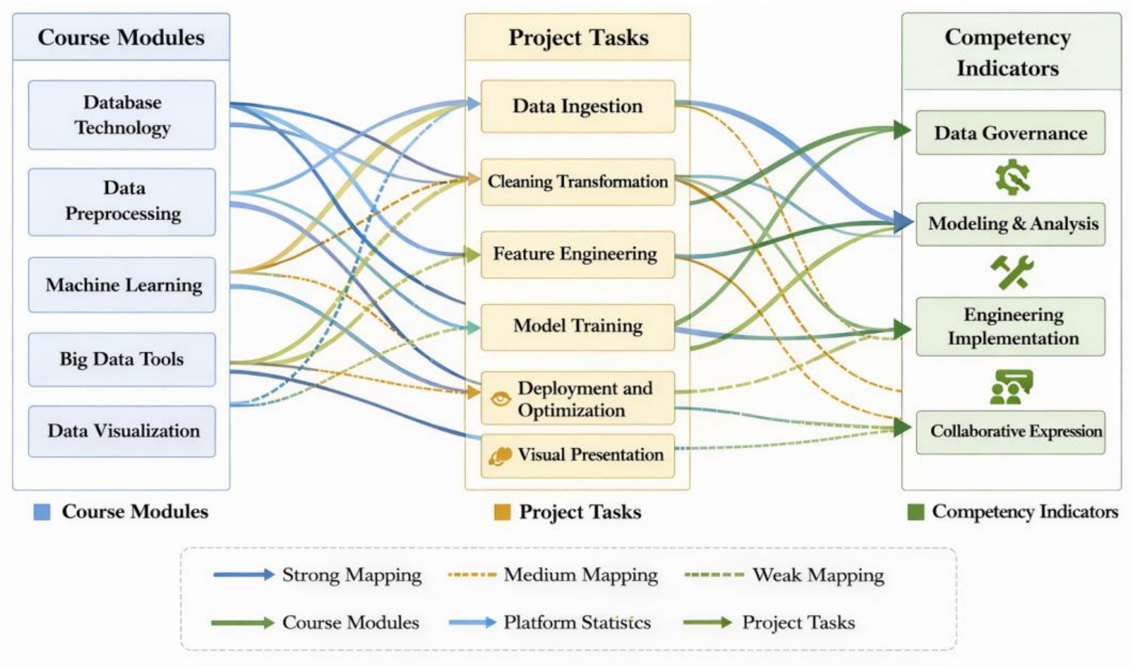

Figure 2: Triadic Mapping of Course Content—Project Tasks—Competency Indicators

2.4 Design of the Practical Teaching Platform and Data Collection Process

The pragmatic teaching platform follows a five-layer design: Task Scheduling Layer- Development and Execution Layer- Data Management Layer- Behavior Collection Layer- Analysis Service Layer. The Task Scheduling Layer will undertake the task of publishing course tasks and ensuring that tasks are completed and roles allocated. It places task orders in a bi-weekly cycle, and attaches each task to a version number of the dataset, runtime image, submission deadline and grading script. The Development Execution Layer is a combination of GitLab code repository, JupyterLab development environment, VSCodeServer, and containerized computing facilities, and a single image configuration of Python 3.11, Spark 3.5, Hadoop 3.3, MySQL 8.0, and Docker 24. By default, each instance is assigned 4 vCPUs, 16 GB of memory, and 50 GB of temporary storage; The data management layer contains the instructional sample repository, enterprise data-masked case repository, and process log repository. Sample data is stored in CSV and Parquet formats, with data-masking policies on the field, version freeze policies, task access tokens are configured;The behavior collection layer utilizes application programming interfaces (APIs) to gather data on login, code commits, execution counts, types of errors, test pass rates, task latency, and collaboration records. The sampling rate is established to 5 seconds; once the events are written to the message queue, the uniform routing of the events to the analysis service layer is done; the analysis service layer cleanses the logs, aggregates the features, detects anomalies, and calculates process metrics, which are then presented as role-based views to instructors, corporate mentors, and students [17].

The sequence of data collection is as follows: event triggering-log standardization-feature extraction-metric calculation-result write-back. The platform assigns each kind of interaction to a unique event identifier, storing the user ID, task ID, and timing of the interaction with the resource, execution results, and so on in a structured format; it uses rule-based deduplication of consecutive submissions, invalid retries, and abnormal interruptions, The student's s process engagement index during the task cycle is defined as:

$$P_s = \lambda_1 \frac{n_s}{N} + \lambda_2 \frac{t_s}{T} + \lambda_3 \frac{c_s}{C} + \lambda_4 \frac{u_s}{U} \tag{5}$$

where P_s is the process engagement index, n_s is the number of valid task completions, t_s is the duration of valid online learning, c_s is the number of code or experiment submissions, and u_s is the number of unit tests passed; N , T , C , and U are the standardized benchmarks for the corresponding metrics; λ_1 , λ_2 , λ_3 , λ_4 are weighting coefficients, and $\sum \lambda_i = 1$ .In the course of the application, the original values of each metric are calculated according to the course week, then interval-normalized, and then combined with task quality metrics to be written back into the student competency profile, which is used to drive task recommendations, milestone alerts, and trigger instructor interventions [18].

2.5 Construction of the Training Quality Evaluation Indicator System

The evaluation indicator system based on 4 steps is designed: "indicator stratification—data binding—weight calculation—result validation. First, a three-level evaluation system with primary dimensions, secondary indicators, and observable variables is built based on the job competency model, the three-way mapping relationship, and the platform log structure. The broad dimensions are established with six categories, namely: knowledge mastery, data governance, analytical modeling, engineering implementation, collaborative expression, and job fit;Second-level measures are further subdivided into 18 specific items, such as database design, data cleaning accuracy, feature engineering quality, model tuning capability, API integration success rate, container deployment completion rate, documentation standardization, and enterprise task response timeliness; the variables of observation are directly The platform summarizes 16-week rolling data, and a 7-day data window. The missing indicator rate threshold is set to 0.1; anything above it will cause data supplementation or removal[19]. To eliminate dimensionality differences, the original indicators

$$Z_{sr} = \frac{x_{sr} - x_r^{\min}}{x_r^{\max} - x_r^{\min} + \varepsilon} \tag{6}$$

where Z_{sr} is the standardized score, $x_r^{\min}$ and $x_r^{\max}$ are the minimum and maximum values of the r th metric, respectively, and ε is a smoothing term to prevent the denominator from becoming zero, set to 10^{-6}. After standardization, the comprehensive training quality index is calculated using a coupled approach of position weight and process weight, defined as:

$$E_s = \sum_{r=1}^{m} \omega_r Z_{sr} \tag{7}$$

where E_s is the comprehensive training quality index for student s , ω_r is the weight of the r th indicator, satisfying $\sum_{r=1}^{m} \omega_r = 1$, and m is the total number of indicators [20]. When calculating weights, initial weights are first generated from the job competency requirements matrix, then adjusted based on the consistency coefficient of enterprise mentor evaluations and the behavioral contribution of the platform; when the engineering implementation capability requirements for a certain job cluster are elevated, the corresponding

ω_r is adjusted upward accordingly. Table 1 presents the specific structure and data sources of the training quality evaluation indicator system.

Table 1: Training Quality Evaluation Indicator System

Primary Dimensions	Second-Level Indicators	Observed Variables	Data Source	Sampling Period
Mastery of Knowledge	Theoretical Knowledge Attainment	Chapter Quiz Scores, Final Exam Scores	Course System	Week/Semester
	Toolchain Proficiency	Python, Spark, SQL Lab Completion Rate	Lab Platform	Week
Data Governance	Data Cleaning Accuracy	Missing Value Handling Accuracy, Outlier Detection Rate	Task Scripts + Test Set	Task-level
	Data Modeling Compliance	Table Structure Design Score, Field Naming Compliance Rate	Project documentation + database	Task Level
Analytical Modeling	Feature Engineering Capabilities	Number of Valid Features, Reasonableness of Feature Selection	Code Repository + Review Records	Task-level
	Model Training Quality	Validation set accuracy, recall, and F1score	Training logs	Task-level
	Hyperparameter Tuning Capability	Number of parameter searches, performance improvement	Training Log	Task-level
Engineering Implementation	Code Conformity	Static analysis pass rate, code duplication rate	Code Repository Scan	Week
	Interface integration success rate	Number of successful API calls, error rollback rate	Platform logs	Task-level
	Container Deployment Success Rate	Image build success rate, service startup success rate	Runtime logs	Task Level
	Task latency control	Time from task submission to acceptance	Task Scheduling System	Task-level
Collaboration Representation	Collaboration Contribution	Commit share, pull request participation	Code Repository + Collaboration Log	Week
	Technical Documentation Quality	Requirement Specifications, Lab Reports, Deployment Documentation Scores	Instructor ratings	Phase
	Visual Presentation Skills	Compliance with charting standards, completeness of result interpretation	Project Defense Record	Stage
Job Fit	Completion of corporate tasks	Corporate Case Delivery Score	Corporate Mentor Score	Stage
	Response Time	Time from Task Acceptance to First Valid Submission	Platform Logs	Task Level
	Overall Matching Score	Job Cluster Matching Index	Evaluation model calculation	Stage
	Stability Metrics	Continuous Task Completion Rate, Abnormal Interruption Rate	Platform Logs	Week

3 Results and Discussion

3.1 Implementation Plan and Sample Design of the Training System

The training system was implemented among 2022 and 2023 undergraduate students majoring in Big Data at a certain university. The implementation period spanned two consecutive semesters, totaling 32 weeks. The sample was divided into an experimental group and a control group. The experimental group adopted the training system described earlier—"job competency modeling—course task mapping—platform process data collection—multi-source evaluation integration"—while the control group continued with the original course organization and centralized training model.To control for confounding factors, both groups were kept consistent in terms of faculty composition, core course credits, class hours, laboratory conditions, and assessment cycles. Differences existed only in project organization methods, the depth of corporate involvement, the granularity of platform data collection, and the evaluation mechanisms.The experimental group was assigned 12 real-world enterprise task chains covering four primary task domains: data ingestion, cleaning and transformation, modeling and analysis, and deployment and presentation; the control group used traditional course experiments and end-of-term comprehensive assignments. During sample screening, individuals who switched majors midway, had prolonged absences, or had a process data missing rate exceeding 10% were excluded. The final analysis sample comprised 118 participants, including 59 from the experimental group and 59 from the control group.Baseline consistency tests were conducted across six variables: entrance exam scores, programming fundamentals, database course grades, mathematical foundations, pre-test ability indices, and tool usage experience. Continuous variables were analyzed using independent samples t-tests, while categorical variables were analyzed using chi-square tests, with a significance level set at 0.05. The results showed no statistically significant differences between the two groups on the primary baseline variables, indicating comparability.Table 2 presents the basic demographic information of the sample and the results of the baseline homogeneity test.

Table 2: Basic Sample Information and Baseline Consistency Test Results

Indicator	Experimental Group (n=59)	Control Group (n=59)	Statistic	P-value
Number of male students/person	34	33	0.034	0.854
Number of female students/person	25	26	—	—
Average age/years	20.4 ± 0.7	20.3 ± 0.6	0.812	0.419
Admission Composite Score/points	79.8 ± 6.4	80.1 ± 6.1	-0.261	0.795
Grade in Introduction to Programming / Points	76.5 ± 7.3	77.1 ± 7.0	-0.455	0.65
Database Fundamentals Score/Points	74.9 ± 8.1	75.6 ± 7.7	-0.481	0.631
Calculus Score/Points	78.7 ± 6.8	79.0 ± 6.5	-0.247	0.805
Pre-test Ability Index	0.482 ± 0.071	0.489 ± 0.068	-0.54	0.59
Python experience ⩾ 1 semester per person	21	23	0.142	0.706
Team project experience: ⩾1 project per person	18	17	0.037	0.848

3.2 Analysis of Course Implementation Process Data

The information on the course implementation process was gathered constantly through the practical teaching platform. Statistical aspects were task completion time, valid submissions, unit test pass rate, error rollback rate, collaboration participation, and enterprise task integration rate. In order to note the gradual changes in the development of competency, two semesters were categorized into an initial phase (Weeks 1-5), an intermediate phase (Weeks 6-11) and a comprehensive phase (Weeks 12-16) which corresponded to the three periods of operation of familiarization of basic tool, parallel project task and final comprehensive delivery respectively. The data on the processes were summed up weekly and the average of the processes were averaged by phase and standard deviation obtained. The findings indicate that the experimental group had a more consistent task execution pattern in all the three phases of the experiment. Interestingly, the rate of change in time spent in completing tasks was higher than that of the control group, but the unit test pass rate and collaboration participation were still on the increase after the intermediate phase. This implies that there was a definite inhibiting force on the learning path due to the mechanism mapping course content to project tasks. Table 3 summarizes the key process indicators for different phases.

Table 3: Statistical Summary of Key Process Indicators by Stage

Phase	Group	Task Completion Time/min	Valid Submissions/ Times	Unit Test Pass Rate / %	Error Rollback Rate / %	Collaboration Engagement	Enterprise Task Adoption Rate / %
Initial Phase	Experimental group	186.4±31.2	8.7±2.1	71.8±8.4	18.5 ± 4.2	0.46 ± 0.09	28.4
	Control group	193.1 ± 34.5	8.1±2.4	70.2 ± 8.9	19.7 ± 4.6	0.44 ± 0.08	0
Intermediate stage	Experimental group	148.7 ± 26.8	10.9 ± 2.7	81.3 ± 7.1	12.6 ± 3.5	0.61±0.10	46.2
	Control group	171.5 ± 29.9	9.0 ± 2.5	76.1 ± 7.8	16.4 ± 4.1	0.51±0.09	0
Comprehensive phase	Experimental group	126.9 ± 22.4	12.4 ± 2.9	88.6 ± 6.5	8.9 ± 2.7	0.73 ± 0.11	63.7
	Control group	159.8 ± 27.6	9.8 ± 2.6	83.2 ± 6.9	14.1 ± 3.8	0.58 ± 0.10	0

Table 3 indicates that the mean time in the first stage of the experiment, the experimental group took 186.4 minutes to complete the task and this was not significantly different to the control group which took 193.1 min in the same stage showing that both groups were in the platform adaptation and task cognition stage at the start of the experiment. When the intermediate stage was reached, the time of the experimental group dropped to 148.7 min, whereas the control group did not change and continued to be 171.5 min, and this is when the gap has started to widen; In the integration phase the time of the experimental group dropped to 126.9 min, whereas the control group did not change and remained 159.8 min, and this is where the Unit test pass rate rose by 13.0 percentage points more than in the control group as it stood at 71.8 in the first phase then 88.6 in the integration phase. The error rollback rate went down to 8.9% (compared to 18.5%), and the trend is evidently downward, which is in agreement with the acceptance scripts, task chain decomposition, and process early warning mechanisms that the platform offers. The experimental group showed an improvement of 0.46 to 0.73 in the collaboration engagement, which was higher than that of the control group, which was 0.58 at the end, showing that the involvement of dual-mentor guidance and the task-based collaborative mechanism stabilized the interactions within the team.

As shown in Figure 3, the analysis of coupling relationships among key process indicators across the three stages reveals a synergistic trend: shortened task completion times correlate positively with increased unit test pass rates, enhanced collaboration participation, and higher enterprise task adoption rates, while the error rollback rate continues to decline. This indicates that the course-task mapping mechanism, combined with platform process interventions, collectively drives simultaneous improvements in execution efficiency, task quality, and collaboration levels.

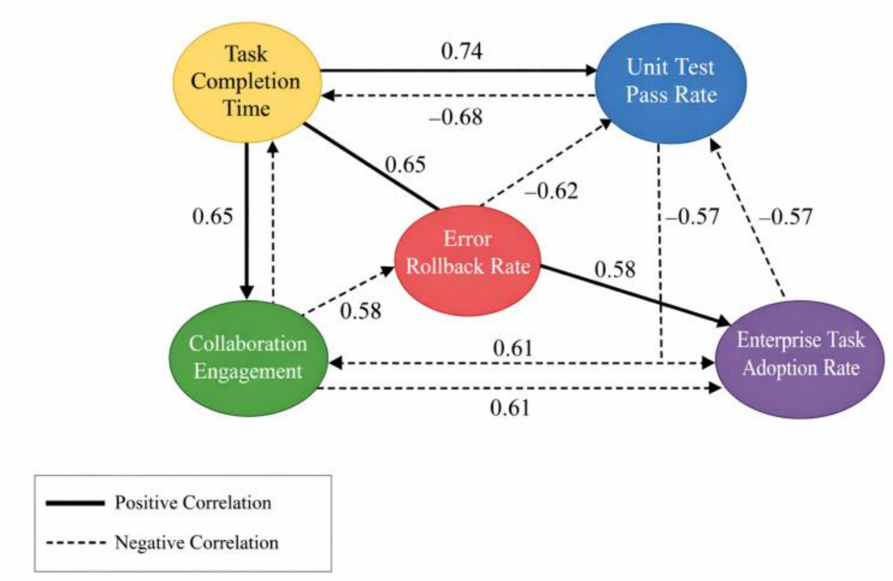

Figure 3 Multi-index Correlation Coupling Analysis Diagram

The dual-plot graph (shown in Figure 4) presents the evolution of the time to complete the tasks by phase and the alterations in the unit test pass rates. In the left figure, the curve of the experimental group has a more negative slope after Week 6, which means that the efficiency of the execution of the students has increased faster when they learned data cleaning, feature engineering, and model training activities. Pass rate in the experimental group kept on increasing in the middle and late stages, whereas in the control group, it stabilized after Week 10, indicating that the compatibility between course material and project activities facilitated the effectiveness of knowledge transfer.

The triptych as illustrated in Figure 4 shows the error rollback rate distribution, the box plot comparison of collaborative engagement and the task type coverage structure: the error rollback rate distribution demonstrates that the long tail of the experimental group became much narrower during the comprehensive phase, the box plot comparison of collaborative engagement demonstrates that the median of the experimental group shifted to the higher level in general and the rate of dispersion decreased, and the task type coverage Together, these alterations prove that the data collection process on the platform is not only capturing the results but also visualizing the process of transferring students to isolated skills training to the execution of entire task chains.

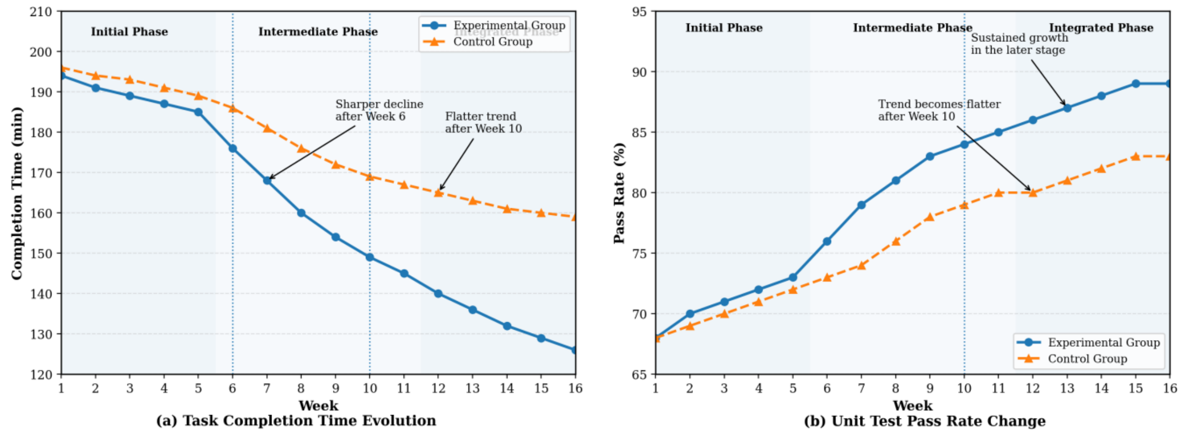

Figure 4: Comparison of Process Performance Across Different Teaching Stages

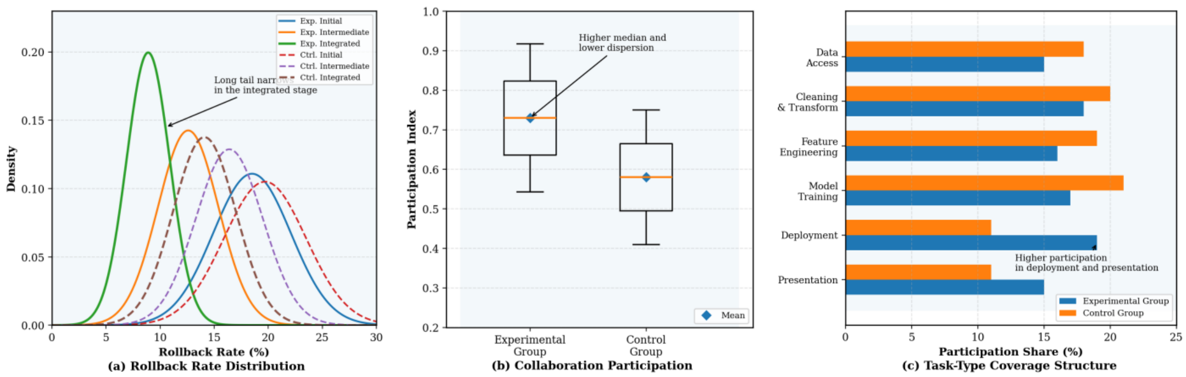

Figure 5: Multidimensional Analysis of Process Behavior and Task Coverage

3.3 Validation of Student Competency Improvement Results

The results of competency improvement in students were computed using the previously described evaluation metric system, and six assessment dimensions were used that consisted of six categories, namely: knowledge mastery, data governance, analytical modeling, engineering implementation, collaborative expression, and job suitability. Pre-test data were based on baseline competency tests at the start of semester and current course data, whereas post-test data were determined through the combination of final comprehensive project grades, platform process metrics, corporate mentor reviews, and instructor reviews. To ensure that individual course grades do not conceal the difference in competencies, the dimension was made standardized and subsequently weighted and then aggregated. The findings indicate that the experimental group recorded increases in all six dimensions and most significant increases were recorded in engineering implementation, job suitability, and collaborative expression. On the contrary, the gains of the control group were mostly focused on the aspects of knowledge mastery and analytical modeling. This means that conventional training routes are better suited to the improvement of theoretical learning outcomes but have a comparatively lesser effect on the development of cross-task execution, collaborative development, and adaptability of real-world projects. The statistical results of competency improvement among students are provided in Table 4.

Table 4: Statistical Summary of Student Competency Improvements

Dimension	Experimental Group Pre-test	Experimental Group Post-Test	Improvement Rate (%)	Control Group Pre-test	Post-test for the control group	Percentage Increase
Knowledge Mastery	0.531 ± 0.072	0.712 ± 0.069	34.1	0.536 ± 0.070	0.681 ± 0.071	27.1
Data Governance	0.472 ± 0.066	0.736 ± 0.064	55.9	0.478 ± 0.068	0.651 ± 0.067	36.2
Analysis and Modeling	0.498 ± 0.074	0.758 ± 0.061	52.2	0.503 ± 0.071	0.689 ± 0.065	37
Engineering Implementation	0.438 ± 0.069	0.759 ± 0.058	73.3	0.445 ± 0.066	0.613 ± 0.062	37.8
Co-expression	0.507 ± 0.081	0.727 ± 0.063	43.4	0.511 ± 0.077	0.628 ± 0.069	22.9
Job Fit	0.469 ± 0.073	0.754 ± 0.060	60.8	0.472 ± 0.071	0.631 ± 0.064	33.7
Comprehensive Training Quality Index	0.486 ± 0.058	0.741 ± 0.049	52.5	0.491 ± 0.055	0.632 ± 0.053	28.7

The comprehensive training quality index of the experimental group (as indicated by Table 4) proceeded to rise by 52.5% (0.486 to 0.741); and the index of the control group (0.491 to 0.632), increased by 28.7%. The post-test score of the experimental group is 0.736, and the control group is 0.651 in data governance dimension, indicating that the data governance sub-tasks like data cleansing, structure design of the tables, and field governance are more of a strong training feedback loop in the enterprise case studies. Both groups had an improvement in the analysis and modeling dimension, and the advantage of the experimental group was largely in the stability of parameter tuning and the quality of feature engineering. The highest difference was found in the engineering implementation dimension, where the experimental group had a 0.438 to 0.759 improvement, and the control group only achieved 0.613. This has direct links to the code repository, containerized runtime environment and automated acceptance testing processes as part of the platform support layer as discussed above. The experimental group had a high score in the job fit dimension, where they scored higher in both enterprise task completion and response timeliness, which implies that being involved in real-world task chains allowed students to develop more concrete operational capacity in terms of job rhythms, delivery standards, and collaboration interfaces.

Figure 6 shows that in the dual-plot diagram, the slope of the change in the six-dimensional competency attainment of the experimental group is steeper, and the slope of the line in the engineering implementation and job fit is particularly steep. The overall distribution of the post-test scores of the experimental group has moved to the right in the comprehensive index distribution chart and has a lower dispersion, indicating that most students made consistent improvement and not a few high-achieving students who cause the mean to be skewed in the upward direction.

As indicated in Figure 7, in the tri-plot, engineering implementation capabilities in the tri-plot comparisons, the box-and-whisker comparison, the median and upper quartile of the experimental group moved all at the same time. The kernel density plot shows a larger percentage of samples in the large score range of the experimental group, whereas the contribution decomposition plot shows that the main sources of the overall index improvement are engineering implementation, job fit, and data governance. This implies that in the synergistic interaction of the course-task-competency mapping and platform-based process interventions, competency gains are not distributed throughout but manifest mainly in those dimensions that are nearest to the enterprise task execution chain.

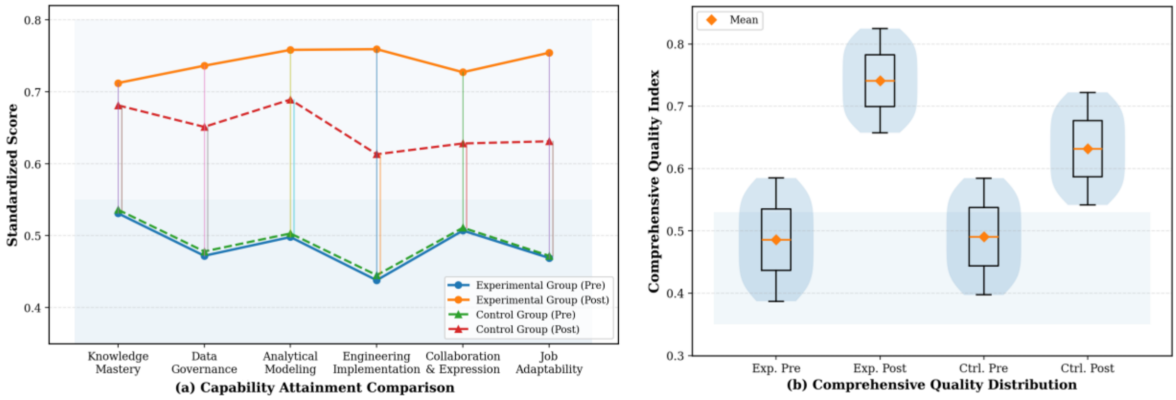

Figure 6: Comparison of Student Competency Achievement and Comprehensive Quality Distribution

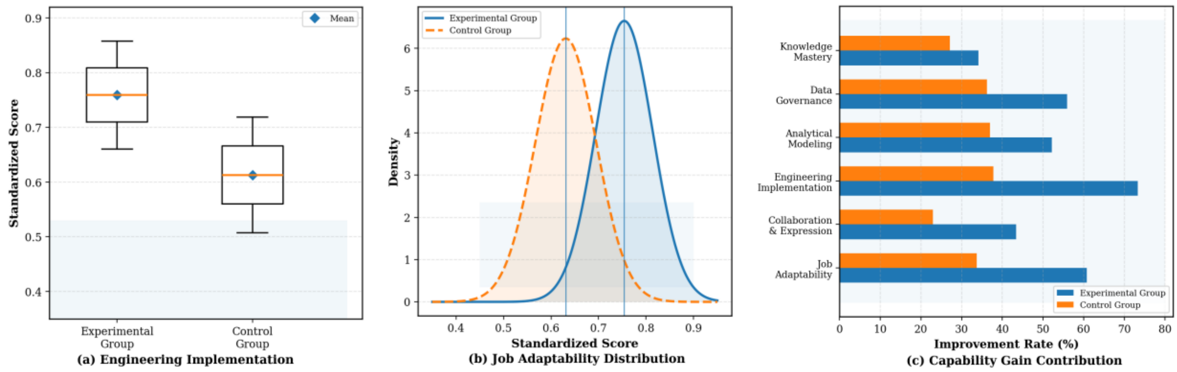

Figure 7: Multidimensional Validation of Student Competency Gains

3.4 Discussion on the Outcomes of University-Enterprise Collaborative Education

The impact of industry-university joint education is largely examined in three dimensions, the degree to which the involvement of the enterprise is introduced throughout the process, the level of such involvement, and the results of the changes in the student competencies. Enterprise involvement during implementation is not a matter of end-of-term defenses or short term lectures, but a multistage process, such as modeling job requirements, developing task scripts, project acceptance, process feedback and outcome evaluation. Industry mentors were involved in 6 rounds of requirement interviews, 12 times reviewed and edited task scripts, 3 interim reviews and scored comprehensive projects in two dimensions. The industry input is not a single external proposal as before but directly incorporated into the design of course activities.Table 5 presents the correspondence between industry participation stages and corresponding educational outcomes.

Table 5: Corporate Participation Stages and Corresponding Educational Outcomes

Corporate Participation Stages	Specific Methods	Corresponding Educational Outcomes	Key Observation Indicators
Job Requirement Modeling	Job Interviews, Technology Stack Alignment	Competency Objectives Better Aligned with Job Cluster Requirements	Job Match Index, Consistency of Competency Weights
Project Task Design	Provide real-world case studies and review task scripts	Task chain structure more closely aligned with real-world business processes	Task coverage and task completion time
Step-by-Step Guidance	Dual-mentor feedback and issue-tracking system	Decrease in error rollback rate during intermediate stages	Rollback rate, test pass rate
Project Acceptance	Corporate mentors participate in grading	Improved standardization of deliverables	Project score, document quality
Results evaluation	Job Fit and Responsiveness Evaluation	More significant improvement in job fit	Company Rating, Response Timeliness
Feedback Optimization	Jointly Revising the Case Library and Evaluation Criteria	More Timely Updates to Course Assignments	Task update cycles and frequency of metric adjustments

The participation in the stage review and final acceptance phases also has a more profound effect on the three dimensions of data governance, engineering implementation, and job fit, as indicated in Table 5; however, the participation in capability modeling and task design stages has a more direct influence on the collaborative expression and delivery standardization. It is worth mentioning that the influence of various phases of participation is not identical; the closer the participation is to the task definition and acceptance criteria, the more the restrictions on the capability development pathway.

As Figure 8 displays, in the dual-diagram, the structure of enterprise participation by stage, and the contribution of gains in capability corresponding to various stages of participation are shown, where the left diagram reveals that the major input in the collaborative work is conducted by participating in task design, stage guidance, and project acceptance. As shown in the right diagram, task design plays a significant role in engineering implementation and data governance, and project acceptance plays an even more prominent role in job suitability and collaborative expression. This difference shows that industry-university collaboration is not always more effective with increased involvement; the success of the collaboration is related to the presence of corporate input at crucial stages of the task chain and the presence of evaluation criteria in the closed-loop of course execution.

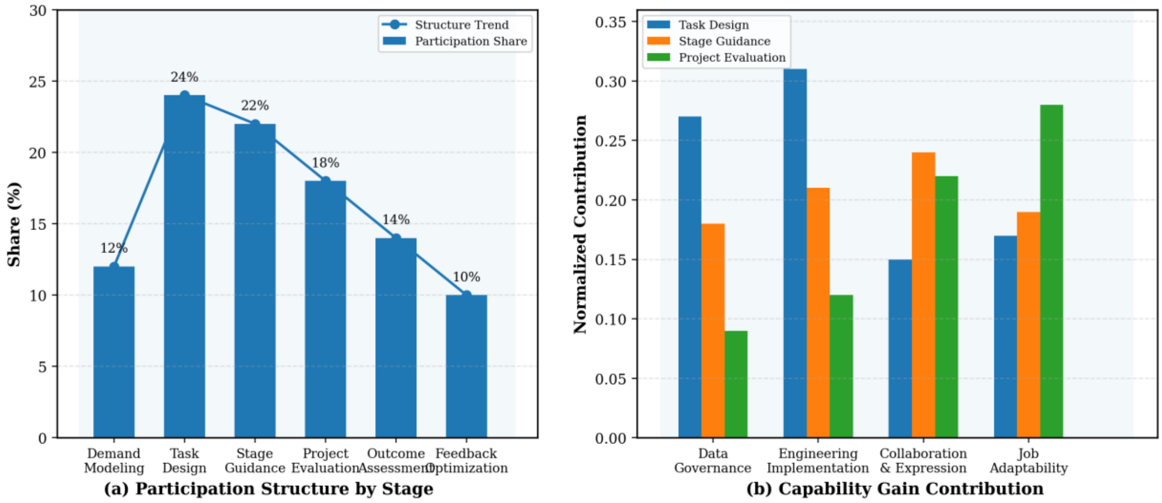

Figure 8: Correlation Analysis of University-Enterprise Participation Structure and Educational Outcomes

3.5 Recommendations for Optimizing the Talent Development System

Judging by the above implementation outcomes, it is possible to optimize the talent development system in a hierarchic way at structural, platform and evaluation layers.

On the structural level, a dynamic update mechanism of job competency models must be created, which includes the incorporation of alterations in enterprise job descriptions, technology stack, as well as project case iterations into semester-level updates. The existing experimental findings suggest that the enhancement of engineering implementation and job fit are relatively fast, whereas the expansion in the analysis and modeling dimension decelerates during the middle and late stages. This implies that the number of tasks connected to model interpretation, deployment monitoring, and cross-module debugging in the course workflow can be further increased. In later versions, full-scale projects might be broken down into a two-level system of chains of basic tasks and chains of enhanced tasks with extra nodes like feature engineering, model compression, and interface encapsulation added to enable students of higher ability.

On the platform level, the granularity of the data collection and diagnostic possibilities of the process data should be improved even more. Although the current platform has the capacity to capture submissions, tests, time-span, and collaborative behavior, it does not have enough features to describe the code quality evolution, exception tracing trajectories, and the cost of switching tasks. Developments in the future might involve code analysis at rest, resource profiling at runtime, automatic grouping of error types, and early detection of potentially dangerous learning nodes. Also, task-level logs will be connected in real-time with competency profiles to create more specific intervention cues. In the case of instructors, it should be not about showing results, but about pointing out the so-called chains of anomalies, in other words, the direct identification of the bottlenecks in the students when it comes to certain task nodes, tool stacks, or testing stages.

At the level of evaluation we need to improve adaptive weighting mechanism to consider the differences between job clusters. The existing overall assessment already encompasses the course grades, platform logs, and the corporate mentors scores, but the relevance of each of the competency dimensions to various job roles is different. It is possible to create evaluation templates specifically to job clusters (data development, data analysis, and machine learning applications) in the future and dynamically adjust the weight settings to engineering implementation, data governance, analytical modeling, and collaborative communication.

Simultaneously, the corporate scoring criteria must be elaborated into measurable items like the completeness of delivery, interface standardization, responsiveness to issues, and business interpretation accuracy. This will minimize subjective scoring inconsistency and make sure that the results of evaluations are consistent with the job competency model.

4 Conclusion

With the construction of a talent cultivation system in the hands of university big data majors in terms of deep industry-education coordination, the overall technical process has been built: the modeling of job competency requirements - mapping of course content to project tasks - practical teaching platforms and process data collection - multi-source quality assessment and result validation. The findings show that the training system can successfully translate enterprise job requirements into calculable, executable and traceable instructional units, Consistency of improvement trajectories is more pronounced in such dimensions like engineering implementation, job fit, data governance, and collaborative expression, whereas university-industry collaboration takes the form of end-stage involvement rather than upfront, embedded interaction.

(1) In terms of systems construction, job corpus modeling and the ternary mapping mechanism have led to a greater fit between course objectives, project tasks as well as competency indicators, lessening the gap between course content and job requirements. The process data collection mechanism based on platforms has enhanced observability of teaching practices, allowing the task time, test passing rates, teamwork, and quality of delivery to be incorporated into a single evaluation system.

(2) In terms of implementation results, the fact that the increase in process measures is accompanied by an increase in competencies means that this training channel not only streamlines the learning process but also enhances the skills of the students to perform chains of tasks in real life. The more the enterprise is involved closer to the task definition, process guidance, and project acceptance, the stronger the competency gains are. This implies that the real trick to industry-academia collaboration is not the frequency of the participation, but the extent of integration between the points of participation and the instructional structure.

(3) As to its weaknesses and future directions, the present scope of validation is mainly focused on one academic field, and a small sample. The variations in institutional resources, occupation type and long-term employment feedback are yet to be introduced into a single analysis. Future research might broaden the sample to grade levels, schools, and clusters of jobs, provide more fine-grained learning behaviour analysis, dynamic weight change, and generative teaching support systems to further optimize the talent development loop of big data professionals.

About the Author

Rongping Wang was born in Fuzhou, Jiangxi, P. R. China, in 1985, He is a lecturer in Guangzhou Huali College, He received the bachelor's degree from East China University of Technology, his master's degree from Guangdong University of Technology. His research interests include big data analysis and mining, machine learning.

Cuicui Chen was born in Guilin, Guangxi, P.R. China, in 1986. She obtained a bachelor's degree from Jinan University in China. Now, she is working at the Health College of Guangzhou Huali Technology Vocational College. Her research interests include big data and accounting.

References

[1] Han, F., Ren, J. (2024). Analyzing big data professionals: Cultivating holistic skills through university education and market demands. IEEE Access, 12, 23568-23577.

[2] Liang, Y., Wang, J., Wang, J. (2025). Reconstructing higher education in the big data and AI era: Interdisciplinary integration and problem-driven talent cultivation. In 2025 International Conference on Distance Education and Learning (ICDEL) (pp. 47-51). IEEE.

[3] Zhang, Z. (2021). Research on talent cultivation for digital media art majors based on the “integration of production and education” in the context of big data. In 2021 International Symposium on Artificial Intelligence and Its Application on Media (ISAIAM) (pp. 86-90). IEEE.

[4] Ni, M. (2025). Developing an integrated talent cultivation model of course-competition-certificate-innovation-application in the context of new quality productive forces: A case study of the business data analysis and application program in higher vocational colleges. Journal of Computing and Electronic Information Management, 18(2), 1-5.

[5] Ma, H., Ding, A. (2022). Construction and implementation of a college talent cultivation system based on deep learning and data mining algorithms. The Journal of Supercomputing, 78(4), 5681-5696.

[6] Cuiling, Z., Chengwei, Y. (2021). Exploring the model of big data talent training based on industry-education integration for IMIS. In 2021 11th International Conference on Information Technology in Medicine and Education (ITME) (pp. 586-590). IEEE.

[7] Miao, Y., Xiao, Z., Zhang, Y. (2025). Impact of a talent cultivation model for industry-education integration in vocational education using artificial intelligence and BPNN. Scientific Reports, 15(1), 38019.

[8] Duan, X. (2025). Driving mechanisms and competency modeling for cultivating business-engineering integrated talent in inland open-economy hubs. iEducation, 1(3), 30-45.

[9] Saleema, J. S., Iyer, L. S., Sankaran, L. (2025). Framework for bridging the skill gap between higher education institutions and industry: A machine learning approach. In Cloud Computing for Smart Education and Collaborative Learning (pp. 222-244). Chapman and Hall/CRC.

[10] Wu, X. (2022). Construction and practice of multiple mixed teaching modes based on big data analysis: A case study of the “International Trade” course. Discrete Dynamics in Nature and Society, 2022(1), 7369920.

[11] Huang, K. (2025). Integrating industry certification into STEM-oriented vocational undergraduate education: Taking the major of financial technology application as an example. Eurasia Journal of Mathematics, Science and Technology Education, 21(9), em2692.

[12] Zou, L., Meilin, F. (2026). A study on the reform of an integrated talent cultivation model

combining medicine, education, healthcare, and tourism under the background of the silver economy. Frontiers in Interdisciplinary Educational Methodology, 3(1), 1-7.

[13] Wei, X. (2021). A classification method of tourism English talents based on feature mining and information fusion technology. Mobile Information Systems, 2021(1), 5520079.

[14] Gao, Y., Chao, Y., Li, D., et al. (2025). Research on the construction of a cloud-edge collaborative system for intelligent processing of outsourced data in industry-education integration. In Proceedings of the 2025 2nd International Conference on Cloud Computing and Big Data (pp. 117-133).

[15] Zhou, Y., Xu, X., Peng, S. (2026). Research on the construction of a knowledge map technology framework for core curricula in the “AI+” context. Frontiers in Economics and Management, 7(1), 130-137.

[16] Wang, C., Xie, J. (2021). Research on the application adaptability of big data and IoT in the discipline construction of mechanical engineering specialties in universities. Security and Communication Networks, 2021(1), 3130841.

[17] Cai, Y. (2025). Pathways to enhancing employment quality for vocational college graduates in construction under the background of intelligent construction: The “four integrations, three supports, and two micro-initiatives” model—A longitudinal tracking study across 15 institutions and industry big data analysis. In Proceedings of the 2025 International Conference on Digital Society and Intelligent Computing (pp. 297-305).

[18] Chilba, S., Kabiraj, S. (2026). Advancing green talent: A Finnish blueprint for sustainable upskilling and university-industry integration. In University-Industry Partnerships for ESG Talent Development (pp. 57-82). IGI Global Scientific Publishing.

[19] Wang, K., Gao, T., Liang, F., et al. (2025). Research on AI-driven teaching reform of microelectronics science and engineering in applied undergraduate institutions from the perspective of industry-education integration: A case study on the curriculum construction of design and implementation of artificial intelligence systems. In Proceedings of the 2nd International Conference on Intelligent Education and Computer Technology (pp. 891-900).

[20] Bhutoria, A. (2022). Personalized education and artificial intelligence in the United States, China, and India: A systematic review using a human-in-the-loop model. Computers and Education: Artificial Intelligence, 3, 100068.