Journal of Infrastructure, Policy and Development 2024, 8(12), 8751.
https://doi.org/10.24294/jipd.v8i12.8751

Article

Integrating LLMs and software-defined resources for enhanced demonstrative cloud computing education in university curricula

Wuming Pan[1,*], Ying Yang[2], Hao Yin[1]

[1] College of Computer Science, Sichuan University, Chengdu 610065, China
[2] Library, Sichuan University, Chengdu 610065, China
*** Corresponding author:** Wuming Pan, panwuming@scu.edu.cn

CITATION

Pan W, Yang Y, Yin H. (2024). Integrating LLMs and software-defined resources for enhanced demonstrative cloud computing education in university curricula. Journal of Infrastructure, Policy and Development. 8(12): 8751. https://doi.org/10.24294/jipd.v8i12.8751

ARTICLE INFO

Received: 26 August 2024
Accepted: 25 September 2024
Available online: 4 November 2024

Abstract: This paper explores the integration of Large Language Models (LLMs) and Software-Defined Resources (SDR) as innovative tools for enhancing cloud computing education in university curricula. The study emphasizes the importance of practical knowledge in cloud technologies such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), DevOps, and cloud-native environments. It introduces Lean principles to optimize the teaching framework, promoting efficiency and effectiveness in learning. By examining a comprehensive educational reform project, the research demonstrates that incorporating SDR and LLMs can significantly enhance student engagement and learning outcomes, while also providing essential hands-on skills required in today's dynamic cloud computing landscape. A key innovation of this study is the development and application of the Entropy-Based Diversity Efficiency Analysis (EDEA) framework, a novel method to measure and optimize the diversity and efficiency of educational content. The EDEA analysis yielded surprising results, showing that applying SDR (i.e., using cloud technologies) and LLMs can each improve a course's Diversity Efficiency Index (DEI) by approximately one-fifth. The integrated approach presented in this paper provides a structured tool for continuous improvement in education and demonstrates the potential for modernizing educational strategies to better align with the evolving needs of the cloud computing industry.

Keywords: LLMs; cloud computing; software-defined resources (SDR); lean principles; university curricula; entropy-based diversity efficiency analysis (EDEA)

1. Introduction

1.1. Background

Cloud computing has revolutionized the way organizations access, manage, and utilize computing resources, providing scalable, on-demand services that align with modern business needs. With enterprises increasingly shifting to cloud-based solutions, there is a growing demand for professionals with deep expertise in cloud technologies (Gartner Research, 2016). However, traditional educational models often struggle to keep up with the rapid pace of advancements in cloud computing, leading to a significant skills gap and impacting the ability of graduates to meet industry expectations.

The emergence of Large Language Models (LLMs), such as ChatGPT, presents new opportunities to enhance educational experiences. LLMs have shown significant potential in providing interactive, adaptive, and personalized learning environments, which can support students in acquiring complex technical skills more effectively (Extance, 2023). Their integration into cloud computing education offers a promising way to modernize curricula, making them more responsive to the dynamic nature of cloud technologies.

1.2. Problem statement

Despite the rapid advancements in cloud computing, educational curricula often lag behind, focusing more on theoretical knowledge than on the practical skills needed for cloud environments. This gap is exacerbated by the increasing complexity of cloud services, which now include Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), DevOps practices, and cloud-native technologies. Current educational models do not sufficiently address the need for hands-on experience with these technologies, leaving students underprepared for industry challenges.

Additionally, while LLMs have shown promise in enhancing learning, their potential remains underexplored in the context of cloud computing education. There is a pressing need to integrate LLMs into curricula to provide students with real-time feedback and personalized learning paths, thereby bridging the gap between theoretical instruction and practical application.

1.3. Research objective

This paper advocates for a transformative approach to cloud computing education by integrating Software-Defined Resources (SDR) and LLMs into university curricula. The study emphasizes the importance of embedding practical applications of IaaS, PaaS, SaaS, DevOps, and cloud-native technologies to better equip students for the evolving demands of the cloud computing industry. Furthermore, it proposes the use of LLMs as interactive learning tools to provide immediate feedback and personalized guidance, fostering deeper understanding and skill acquisition in cloud technologies (ByteDance Cloud Native, 2024; Extance, 2023).

In university education, achieving adequate diversity in a course means ensuring that: 1) the core themes of the course are properly organized; 2) the most valuable knowledge in the field is taught, while less significant content is excluded; and 3) the course is agile enough to synchronize with the evolving knowledge in the field. This approach is supported by agile course design principles that emphasize flexibility and adaptability in curriculum development (Rowan et al., 2022). Moreover, the use of agile methodologies fosters responsiveness to changing educational requirements, ensuring that students are equipped with the latest skills and knowledge (López-Alcarria et al., 2019). Lastly, incorporating innovative course design strategies like inquiry-based learning and project-based approaches ensures that the focus remains on critical knowledge areas, enhancing both engagement and relevance in the educational process (Mintz, 2021).

To achieve this, the paper introduces the innovative Entropy-Based Diversity Efficiency Analysis (EDEA) framework, which assesses and optimizes both the diversity and efficiency of educational content. While traditional methods like the Herfindahl index and Theil entropy index have been widely applied to measure diversity in higher education (Widiputera et al., 2017), the EDEA framework goes further by integrating a diversity metric with an efficiency measure. This integrated strategy bridges the gap between traditional educational practices and the rapidly evolving technological landscape, preparing students to thrive in modern cloud environments.

2. Materials and methods: Implementing software-defined resources in cloud computing education

2.1. Concept and definition

Software-Defined Resources (SDR) have two key implications in the context of cloud computing education. First, SDR refers to the abstraction of physical resources into virtual entities managed through software, enabling the dynamic allocation, configuration, and centralized management of cloud services and resources. This can be achieved through cloud platforms or code-based tools such as Azure Service Fabric (Bai, 2019) and Kubernetes (Burns et al., 2022), which allow developers to define and manage cloud resources programmatically. This aligns with the interactive and hands-on approach needed in modern cloud education, where tools like LLMs can enhance the understanding and application of these concepts.

Second, SDR serves as a guiding principle for establishing the content of cloud knowledge and skills education. Only content directly related to SDR—meaning the use of cloud technologies to define, manage, and operate resources—should be considered part of cloud computing knowledge.

The following definition summarizes the core concept of cloud computing and SDR:

Definition 1. *Cloud Computing and Software-Defined Resources (SDR). In essence, the cloud consists of resources, and cloud computing abstracts and manages those resources using software technologies. This is called Software-Defined Resources (SDR).*

Technologies or concepts that are not directly connected to SDR and can exist independently should not be taught as cloud knowledge. This distinction is crucial because the scope of cloud services has become increasingly complex, and current teaching practices often stray from this core. By clearly defining what constitutes cloud computing knowledge, we can ensure that cloud education remains focused on the essential competencies required for modern cloud environments, rather than being diluted by peripheral topics.

In the realm of cloud computing, traditional terms such as Software-Defined Infrastructure (SDI) (Kandiraju et al., 2014) and Software-Defined Data Center (SDDC) (Gartner Research, 2016) are widely recognized. While SDI and SDDC provide a solid foundation for understanding cloud infrastructure, they do not fully address the evolving needs of developers and application deployers. Recent advancements, such as machine learning-based performance prediction models for Software-Defined Networking (SDN), further demonstrate the potential of integrating ML techniques to optimize SDR environments (Jiang et al., 2024).

2.2. The lag and limitations of educational disciplinary classification

Current university software education lags behind the needs of the industry, primarily because the scope of software technologies is vast and evolves rapidly. It is difficult for educators to keep pace with these changes, as mastering the breadth and depth of modern software development is a challenge even for experts. As a result, curricula often fail to adequately prepare students for the demands of the software industry.

The traditional classifications of computer science and software engineering, as outlined in the ACM Computing Classification System (CCS) (ACM, 2012), the IEEE/ACM Joint Task Force on Computing Curricula (IEEE Computer Society and ACM, 2013), and SWEBOK (Software Engineering Body of Knowledge) (IEEE Computer Society, 2014), offer detailed standards and recommendations. However, modern software development has expanded to cover multiple programming languages, frameworks, toolchains, and emerging technologies such as cloud computing, microservices, and DevOps. Although information technology and software engineering have rapidly evolved over the past decade, SWEBOK, as a crucial software engineering knowledge classification system, has not been updated since 2014. This stagnation means that many modern practices and technologies, such as cloud-native architectures and continuous integration/continuous deployment (CI/CD), are not adequately represented.

Cloud computing, which plays a critical role in modern software infrastructure, further amplifies this gap. The rapid growth and complexity of cloud technologies, from infrastructure (IaaS) to platform services (PaaS), make it even harder for academic institutions to keep up. Consequently, cloud computing education has never been properly established, leaving students underprepared for the cloud-centric realities of the software industry.

2.3. Cloud computing: An evolved software engineering

Cloud Computing can be seen as an advanced version of software engineering, as it diverges significantly from traditional software engineering concepts.

- Architectural Perspective Shift: In cloud environments, issues like resilience, availability, and security, which were traditionally the focus of architecture design, are now addressed by cloud service capabilities, leading to a greater focus on microservices architecture. For example, Azure offers multiple redundancy options for storing data, which can include storing data in either three or six copies, depending on the redundancy level chosen (Koo, 2020).
- Automation of Architecture: The "microservices orchestration" in cloud computing achieves automated management of architecture, marking a significant departure from basic software engineering. These changes demonstrate that cloud computing is no longer merely a supplement to traditional soft-ware engineering but represents an evolution, bringing new challenges and opportunities. For example, ByteDance uses Kubernetes, a cloud-native container orchestration tool, to manage over 10 million Pods and more than 210,000 servers in China (ByteDance Cloud Native, 2024).
- Emphasis on DevOps Automation: Cloud computing has driven the automation of software engineering processes, particularly through DevOps practices such as deployment, scaling, and infrastructure management. Many manual operations

have been replaced by automated tools and platforms (Scholl et al., 2019). Traditional computer science and software engineering courses typically focus on foundational concepts like algorithms, data structures, and object-oriented programming, often emphasizing manual processes in software development rather than the automated practices now common in cloud-native environments. With the rise of cloud-native technologies, the focus has shifted to developing and managing applications across distributed environments, introducing new categories such as microservices architecture, continuous integration and continuous deployment (CI/CD), and Infrastructure as Code (IaC). These areas are crucial for building resilient and scalable systems but are often not deeply covered within traditional educational frameworks.

2.4. CNCF cloud native landscape: A new knowledge classification

The increasing complexity of cloud-native environments, as discussed in the previous section, has necessitated the development of new frameworks for understanding and categorizing modern software development knowledge. The CNCF Cloud Native Landscape (CNCF Cloud Native Landscape, 2024), considered a natural choice, includes hundreds of tools, providing a new classification of software development knowledge that goes beyond traditional computer science and software engineering classifications.

Table 1. Evolving knowledge framework in cloud-native technologies: Categorization and significance across key domains.

Category	Subcategories
App Definition and Development	Application Definition & Image Build, Database, Continuous Integration & Delivery, Streaming & Messaging
Orchestration & Management	Scheduling & Orchestration, Service Mesh, Remote Procedure Call, Service Proxy, API Gateway, Coordination & Service Discovery
Runtime	Cloud Native Storage, Cloud Native Network, Container Runtime
Provisioning	Security & Compliance, Automation & Configuration, Container Registry, Key Management
Observability and Analysis	Observability, Continuous Optimization, Chaos Engineering, Feature Flagging

Definition 2. *Cloud Native. Cloud-native technologies leverage cloud infrastructure, including IaaS, PaaS, and other cloud services, with node provisioning, storage, networking, and load balancing deeply integrated into microservices and pods. Built on top of these services, cloud-native approaches introduce abstractions that enable automation, scalability, and resilience, providing a modern perspective on software development and engineering. These technologies are designed to create dynamic, scalable, and distributed systems that efficiently address the complexities of modern, cloud-centric software environments, while also supporting continuous integration, continuous deployment (CI/CD), and DevOps practices.*

Table 1 outlines the evolving knowledge framework in cloud-native technologies, representing a significant departure from traditional technology frameworks. It focuses on modern cloud-native technologies that have emerged largely within the past decade. A substantial portion—likely over 70%—of the technologies and knowledge represented in this framework have developed or matured in the last 10 years.

Cloud-native technologies represent a significant shift in paradigms, rapid evolution of cloud-native tools, and a notable change in focus.

2.5. SDR: Addressing the design challenges of cloud computing courses

Cloud computing, as a critical technology in modern software development, has become an essential part of the software school curriculum. However, software schools face multiple challenges when teaching cloud computing in the current higher education environment.

- The Overlooked Perspective—Cloud as a Business: From a commercial standpoint, "cloud" is essentially the commercialization of computing technology and resources. The core of cloud services lies in the on-demand provision of computing resources, storage, networking, etc.—all of which are software-defined resources (SDR). This is fundamentally different from the traditional IT infrastructure purchase and management methods. Understanding the business model of the cloud and recognizing how it offers flexibility, scalability, and cost-effectiveness to enterprises is a core part of cloud computing education.
- Breadth and Depth of Knowledge: SDR has triggered rapid evolution in soft-ware engineering and development technologies. The expansion of software development knowledge and the diversification of technology stacks make it challenging to address software development skills education within traditional computer science knowledge and educational discipline classifications.
- Misconceptions in Course Design: There has long been confusion in course design. Distributed computing, big data, and virtualization technologies, although related to cloud services, are not unique to them. These technologies existed before the rise of cloud computing and are widely applied in non-cloud environments. These technologies themselves are not SDR, although they enable SDR. The complexity and depth of these technologies make them more suitable for specialized courses rather than being the core content of an introductory cloud computing course. Overemphasizing these non-cloud-specific technologies in a foundational cloud course may lead to two issues:
    - Lack of Depth: Due to time and course schedule constraints, students may find it challenging to deeply understand and master these complex technologies in an introductory cloud course, ultimately resulting in superficial knowledge without systematic understanding.
    - Hindering the Formation of Correct Concepts: Students may be confused by the course content and fail to accurately understand the core concepts and true value of cloud computing, mistakenly thinking that cloud computing is merely a combination of distributed computing, big data, and virtualization, thereby overlooking not only its commercial value but also its role in enabling scalability, flexibility, and innovation through services like Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).
- Balancing Course Design: How to balance sufficient theoretical background with ample practical opportunities in a foundational course is a significant challenge.

The course needs to strike a balance between breadth and depth, ensuring that students can apply this knowledge in real projects upon completing the course.

- Rapid Technological Updates: The field of cloud computing is rapidly evolving, with technology stacks and best practices potentially changing within a short period. Therefore, course content needs to be regularly updated to stay aligned with industry trends.

2.6. Teaching strategies based on SDR

To effectively address these challenges, teaching strategies based on SDR can be adopted.

- Focus on the Essence of Cloud Services: A foundational cloud computing course should primarily focus on the business model of cloud services, on-demand services, and the abstraction and management of resources, helping students understand that the essence of "cloud" is to provide flexible and efficient computing resources rather than being a collection of specific technologies.
- Distinguishing Technology from its SDR Application: Distributed computing, big data, and virtualization technologies can be briefly introduced, but the emphasis should be on how these technologies are applied within the context of Software-Defined Resources (SDR) in the cloud environment, rather than focusing on their theoretical foundations or implementation details. By discussing these technologies in relation to cloud computing applications, students can better understand how cloud services utilize them to deliver scalable, flexible, and commercially viable solutions.
- Using Real Cloud Resources: By directly operating resources on cloud platforms, students can experience high availability, auto-scaling, data redundancy, and other features that are vital components of modern cloud computing. Using real cloud resources enhances the practical aspect of learning and helps students better adapt and apply these technologies in their future careers. Therefore, hands-on experience with cloud resources is crucial for cloud computing education.
- Establishing Correct Concepts: To help students form a correct understanding of cloud computing, it is essential to base knowledge selection, hands-on practice, and case studies on the principles of Software-Defined Resources (SDR). By focusing on real-world applications that illustrate how SDR drives scalability, flexibility, and automation in cloud services, students can better grasp the commercial value of cloud computing while avoiding common misconceptions.
- Dynamic Course Content Adjustments: Dynamic adjustments to course content are a critical strategy to address rapid technological updates. By combining student feedback, course content can be regularly updated to reflect the emergence of new technologies and tools in the field of cloud computing, ensuring that students are learning practical and forward-looking knowledge.

Through these strategies, software schools can more effectively balance theory and practice when teaching cloud computing, helping students build a solid knowledge foundation and flexible application capabilities, thereby laying a strong foundation for their career development in cloud computing-related fields.

3. Integrating lean principles with SDR and LLMs in cloud computing education

Integrating SDR into curricula ensures students gain practical skills in managing cloud resources, which is crucial for modern cloud environments. LLMs can aid in this by providing immediate assistance and enhancing problem-solving capabilities. For example, tools like Microsoft Copilot in Azure (Microsoft Copilot in Azure, 2024) offer real-time guidance and support, helping students to navigate and understand complex cloud environments more effectively.

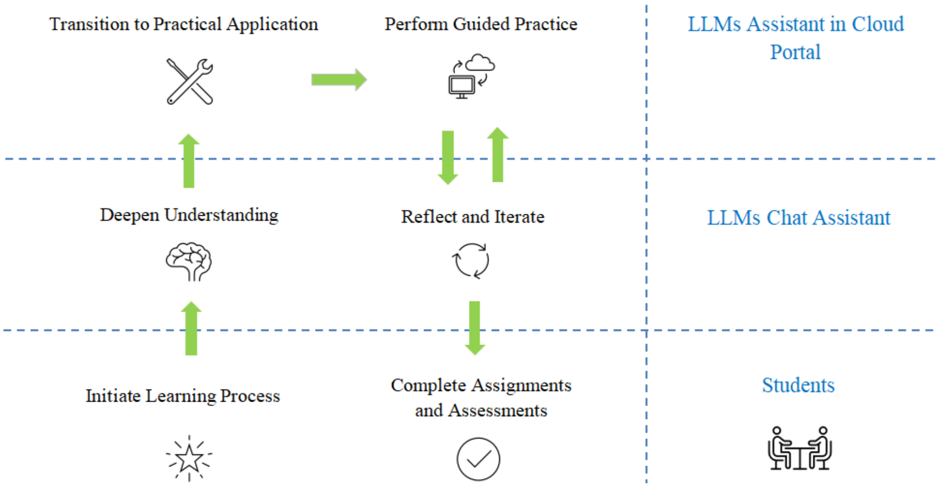

Figure 1. Workflow for Integrating GPT-4o (LLMs Chat Assistant) and Azure Copilot (LLMs Assistant in Cloud Portal) in Cloud-Native Learning.

Figure 1 illustrates the integrated workflow for utilizing GPT-4o (LLMs Chat Assistant) and Azure Copilot (LLMs Assistant in Cloud Portal) in a cloud-native learning environment. This diagram depicts the step-by-step process where students begin by initiating the learning process, deepening their understanding of cloud-native concepts through interactions with GPT-4o, and then transitioning to practical applications facilitated by Azure Copilot. The workflow emphasizes the iterative nature of learning, where theoretical insights gained through GPT-4o are applied in real-world.

As cloud computing continues to evolve, tools like Microsoft Copilot in Azure are expected to become increasingly prevalent across all major cloud platforms and cloud-native tools. These integrated tools will provide real-time intelligent assistance, helping users manage and utilize cloud resources more efficiently. As this trend progresses, each major cloud service provider is likely to introduce its own intelligent assistant, further enhancing the user experience and driving innovation in cloud computing education.

The concept of Lean principles, as detailed by Scholl et al. (2019), focuses on eliminating waste and optimizing processes to enhance efficiency. Originally rooted in manufacturing, particularly the Toyota Production System, these principles have profound implications in the context of modern cloud computing and software development. This can be particularly seen in the integration of Software-Defined Resources (SDR) and Large Language Models (LLMs) in cloud computing education.

Incorporating SDR and LLMs into cloud computing education is a natural extension of Lean principles. Both SDR and LLMs enhance efficiency, reduce waste, and optimize processes—key tenets of Lean methodology. In the context of cloud computing courses, these technologies ensure that resources are used effectively, and students receive high-quality, streamlined education.

LLMs streamline the learning process by providing real-time support and personalized feedback, ensuring that students can learn more effectively and efficiently.

- Immediate Support: LLMs can provide instant answers to student queries, reducing downtime and ensuring the learning process remains smooth and uninterrupted. This immediate support helps maintain a continuous and efficient learning environment, which is particularly beneficial given the numerous foundational concepts involved in cloud technology, allowing students to complete the learning content within the given timeframe.
- Personalized Learning Paths: By tailoring responses to each student's needs, LLMs ensure that every student receives the specific guidance they require, thereby enhancing learning efficiency. In advanced courses like cloud computing and application development, students often have diverse interests and technical backgrounds, making a personalized learning path highly valuable.
- Automated Assessment: LLMs can assist in grading and providing feedback on assignments, accelerating the assessment process and allowing instructors to focus on more complex teaching tasks. With advanced models like OpenAI's o1 (OpenAI, 2024), which inherently utilize techniques such as Chain of Thought (CoT) reasoning (Wei et al., 2022), LLMs can significantly automate the assessment process in cloud technology learning, reducing the reliance on manual prompt engineering.
- Resource Utilization: LLMs can curate and summarize vast amounts of educational material and cloud technology knowledge, making it easier for students to grasp complex concepts without being overwhelmed by information overload. Additionally, this prevents students from wasting time on software installation and VM configuration.

By integrating SDR and LLMs into cloud computing education, educators can apply Lean principles in a modern technological context, fostering an environment of continuous improvement and efficiency. This approach not only enhances the educational experience but also prepares students to effectively manage and utilize cloud technologies in their future careers.

4. Practical application and curriculum integration

Having explored the integration of Lean principles with Software-Defined Resources (SDR) and Large Language Models (LLMs) in cloud computing education, the next step is to demonstrate how these theoretical frameworks are practically applied within a cloud computing curriculum. It is through hands-on experience, real-world applications, and access to essential cloud resources that students can solidify their understanding of these concepts. In this section, we outline the cloud resources utilized in the course and highlight key practice topics that ensure students gain the necessary skills to thrive in modern cloud environments.

Table 2. Cloud resources usage in cloud computing course.

Cloud Resources	Unit	Total Amount
Cloud VM Servers (on Tencent Cloud)	2 CPU cores, 4GB of memory, and 4Mbps of bandwidth.	Approximately 40,000 h
Kubernetes Cluster (on Aliyun Cloud)	2 nodes	3 months, deployed over 270 applications
Azure Container Registry	Container images	Approximately 300 images
Azure Cognitive AI	Hours * Endpoints	4 weeks * 2 endpoints
Azure OpenAI	Hours	3 weeks
Azure Functions	Hours	4 weeks
Azure Data Lake	Hours	3 months

4.1. Cloud resources usage table

Our cloud computing course typically accommodates around 60–80 students per semester. The course is designed to provide hands-on experience with a range of cloud services and technologies, requiring substantial cloud resources to support the various practical exercises and projects. **Table 2** summarizes the cloud resources utilized throughout a semester-long cloud computing course, detailing the specific cloud services used, their configurations, and the total amount of each resource consumed. These resources include virtual machines (VMs), Kubernetes clusters, container registries, cognitive AI services, OpenAI services, Azure Functions, and data lake storage, all of which are integral to providing students with a comprehensive learning experience in cloud computing.

4.2. Course practice topics

To ensure students gain hands-on experience with cloud computing technologies, the following practice topics are integrated into the course:

a) Creating cloud servers
b) Constructing and managing containers
c) Using Azure App Service
d) Using Azure Container Registry
e) Programming with Azure Data Lake objects and tables
f) Deploying Pods to Kubernetes Cluster and accessing via NodePort
g) Utilizing Azure DevOps for CI/CD pipelines
h) Large-scale deployment of applications and services on Kubernetes Cluster
i) SaaS AI for object detection
j) SaaS AI for ChatGPT conversations

4.3. Impact on other courses

Cloud platforms and services not only influence cloud computing courses but also impact other subjects. For example, in the course "Algebraic Code and AI Frameworks," which provides an overview of all mainstream programming languages, cloud technology is leveraged to provide practical tools to students. This course uses VMs and containers to facilitate hands-on practice as outlined in **Table 3**.

Table 3. Practical teaching of 20 popular programming languages in a single course.

Course Component	Technology Used	Details
Swift, Haskell, Rust	VM	Hands-on REPL programming
C#, F#, Python	NET Interactive Notebook (NET Interactive, 2024)	Container-based learning
Scala, Kotlin, et al.	Zeppelin (Apache Zeppelin, 2024)	Container-based learning

5. Development and definition of entropy-based diversity efficiency analysis (EDEA)

By integrating cloud technologies and hands-on practice in the curriculum, students gain valuable real-world skills. However, beyond the technical aspects, it's equally important to ensure that the educational content itself is diverse and efficiently aligned with students' evolving needs in modern cloud computing environments. This leads to the need for a systematic evaluation of how well the curriculum balances diversity and effectiveness. In the following section, we introduce the Entropy-Based Diversity Efficiency Analysis (EDEA), a framework developed to assess and optimize the distribution and efficiency of educational content, informed by the datasets derived from cloud computing course implementations.

In educational research, several methods have been established to measure diversity, particularly in the context of curriculum design and institutional policies. Commonly used metrics such as the Herfindahl index, Shannon entropy, and the Theil entropy index have provided valuable insights into the distribution and concentration of diverse elements within an educational setting (Widiputera et al., 2017). These methods primarily focus on measuring diversity by examining the spread or concentration of different categories, such as courses, disciplines, or student demographics, within a given framework. While these indices offer quantitative assessments of diversity, they often overlook the efficiency or relevance of the distributed content, which is crucial for dynamic learning environments that must adapt to rapidly evolving fields like cloud computing.

Recognizing this gap, the Entropy-Based Diversity Efficiency Analysis (EDEA) framework presented in this paper introduces a novel approach by integrating traditional diversity measurement with efficiency metrics. Unlike conventional methods that solely focus on the extent of diversity, the EDEA framework enables the comparison of the diversity efficiency across different courses, allowing educators to reference these metrics when assessing the effectiveness of Software-Defined Resources (SDR) and Large Language Models (LLMs). Additionally, the EDEA framework can be extended to analyze other educational factors, providing a flexible tool for evaluating a wide range of educational interventions. This dual focus ensures that, while the curriculum is diverse, it also aligns with the practical and evolving needs of students, particularly in technical fields where both depth and breadth are essential. This approach is in line with recent studies that emphasize the need for more comprehensive diversity measures that consider statistical efficiency and robustness (Ghosh and Basu, 2023).

In the context of rapid advancements in cloud computing and Large Language Models (LLMs), educators have unprecedented opportunities to expand the breadth and depth of their teaching content. This expansion is not merely about increasing the quantity of content, but more importantly, it enables the inclusion of material that would otherwise be constrained by traditional teaching conditions. By leveraging cloud technology and LLMs, we have been able to extend our teaching content to its fullest potential, allowing students to engage with more real-world applications and technical practices.

To systematically analyze and quantify the effects of this expanded teaching content, we propose the Entropy-Based Diversity Efficiency Analysis framework (EDEA framework). The EDEA framework evaluates and optimizes the diversity and efficiency of course content through three key steps, ensuring that students are exposed to a broad and balanced knowledge system throughout their learning process, while also allowing comparisons of efficiency across different learning environments and courses.

Definition 3. *Entropy-Based Diversity Efficiency Analysis (EDEA). The Entropy-Based Diversity Efficiency Analysis (EDEA) is a quantitative framework designed to assess and optimize the diversity and efficiency of educational content using entropy metrics. It evaluates how well-balanced and efficiently distributed key knowledge components are within a course to ensure an optimal learning experience. EDEA can also incorporate external influencing factors into the time-series data of the Diversity Efficiency Index (DEI). By applying both linear and nonlinear regression analyses, it determines the extent and effectiveness of these factors in improving DEI.*

Having introduced the EDEA framework and its purpose in optimizing the diversity and efficiency of educational content, we now delve into the step-by-step process for applying EDEA. The following sections detail the methodology, beginning with the calculation of entropy for teaching content.

5.1. Step 1: Calculating the entropy of teaching content relative to a given range

Information Entropy is a fundamental concept in information theory used to measure uncertainty or diversity within a system. In the analysis of educational content, entropy can be used to evaluate the distribution uniformity of various knowledge modules within a course. Specifically, the entropy calculation formula is as follows:

$$P(x_i) = \frac{\text{Value of } (x_i)}{\text{Sum of all values}} \tag{1}$$

Here, $P(x_i)$ represents the proportion of the *i*-th knowledge module in the total teaching content of a course. Based on this, the actual entropy of the course can be calculated as:

$$\text{Entropy} = -\sum P(x_i) \log_2 P(x_i) \tag{2}$$

A higher entropy value indicates a more uniform distribution of course content and a broader coverage of knowledge.

5.2. Step 2: Calculating the diversity efficiency index (DEI)

The Diversity Efficiency Index (DEI) is used to measure the diversity and distribution efficiency of teaching content. The calculation of DEI involves using the Weighted Max Entropy, which considers the relative importance of each knowledge module in the course content. The formula for calculating Weighted Max Entropy is:

$$\text{Weighted Max Entropy} = -\sum_{i=1}^{n} \frac{w_i}{\sum w_i} \log_2 \left(\frac{w_i}{\sum w_i}\right) \tag{3}$$

where w_i represents the weight of the i-th knowledge module. When all weights w_i are equal, the formula for Weighted Max Entropy simplifies to:

$$\text{Max Entropy} = \log_2 n \tag{4}$$

which represents the maximum possible entropy for uniformly distributed knowledge modules. The DEI is then defined as the ratio of actual entropy to weighted max entropy:

$$\text{DEI} = \frac{\text{Actual Entropy}}{\text{Weighted Max Entropy}} \tag{5}$$

The DEI value ranges between 0 and 1. A value closer to 1 indicates that the distribution of course content is near the ideal state, balancing the breadth of coverage with the appropriate allocation of teaching time and focus for each module. In the calculation of weighted maximum entropy, different categories are assigned different weights according to their importance. Therefore, the DEI may exceed 1, indicating that the diversity of the actual data distribution surpasses the expected diversity of the weighted distribution.

5.3. Step 3: Analyzing the impact of different factors on DEI using regression and nonlinear regression

To further understand and optimize the design of teaching content, we use linear regression and nonlinear regression analysis to measure the quantitative impact of different factors on DEI and assess the significance of these factors on DEI values. In this study, we mainly consider the following factors:

- Course Year (Year): Evolution of course setup and content over time.
- Course (Series): Differences among various course series.
- Use of Cloud Technology (X1): Whether cloud technology and Software-Defined Resources (SDR) were used.
- Use of LLMs (X2): Whether Large Language Models (LLMs) were introduced in teaching.

Using Linear Regression Analysis, we construct the following regression model to evaluate the impact of these factors:

$$\text{DEI} = \beta_0 + \beta_1 \times \text{year} + \beta_2 \times \text{series} + \beta_3 \times x1 + \beta_4 \times x2 \tag{6}$$

where β_0 is the intercept and β_1, β_2, β_3, β_4 are the coefficients estimated by the regression model. Notably, the variable series is a categorical variable that has been transformed using one-hot encoding. As a result, β_2 represents a vector of coefficients corresponding to the different categories within series. In this case, the regression equation expands to include multiple terms for each one-hot encoded category of series. For example, if series includes two categories, the model becomes:

$$\text{DEI} = \beta_0 + \beta_1 \times \text{year} + \beta_{2,1} \times \text{series}_1 + \beta_{2,2} \times \text{series}_2 + \beta_3 \times x1 + \beta_4 \times x2 \quad (7)$$

Here, $\beta_{2,1}$ and $\beta_{2,2}$ represent the specific effects of the individual categories of *series*. This allows the model to account for the distinct influence of each category on the dependent variable DEI.

Building upon this, we further employ Random Forest Regression to capture the nonlinear relationships between these factors and DEI, especially when the interactions among different factors have complex effects on the diversity efficiency of teaching content. Through Random Forest analysis, we can derive the importance ranking of each factor on DEI, providing data support for educators to optimize course design.

5.4. The value of the EDEA framework

The EDEA framework not only equips educators with a systematic method to assess and compare the diversity and efficiency of teaching content across different courses, but also provides valuable insights into how Software-Defined Resources (SDR) and Large Language Models (LLMs) enhance educational efficiency. By integrating entropy calculations, DEI metrics, and regression analysis, educators can pinpoint specific course modules for optimization and adjust teaching strategies accordingly. This ensures that courses remain aligned with the evolving technical landscape, addressing students' practical needs in rapidly changing environments such as cloud computing and AI-driven technologies. The flexibility of the EDEA framework also allows for its extension to analyze other educational factors, providing a comprehensive tool for continuous improvement in curriculum design.

6. Data analysis

6.1. Dataset description

This study utilizes two datasets, SDR-Dataset and AI-Dataset, spanning six years from 2019 to 2024. Both datasets contain the same amount of data each year, allowing for consistent year-over-year analysis.

- SDR-Dataset focuses on the "Cloud Computing and Cloud Services" course, detailing the allocation of total course hours across six categories: IaaS, PaaS, SaaS, DevOps, Cloud-Native, and SDR. The weight of each category in the SDR-Dataset is equal to 1, reflecting a balanced curriculum approach.
- AI-Dataset centers on the "Algebraic Code and AI Framework" course, showing the proportional distribution of course hours across 20 programming languages. Each language is rated on a 5-point scale (from 1 to 5), indicating its importance in the course content, with higher numbers signifying greater emphasis. The total proportion of all languages each year sums to 1.

These datasets provide a basis for calculating entropy and Diversity Efficiency Index (DEI) to analyze diversity and effectiveness in educational content across different years and categories.

To maintain conciseness and clarity, the raw data tables for the SDR-Dataset and AI-Dataset are not included in the main body of this paper. Instead, they are provided as supplementary material, allowing interested readers to reference and verify the details as needed.

6.2. Entropy calculation and visualization

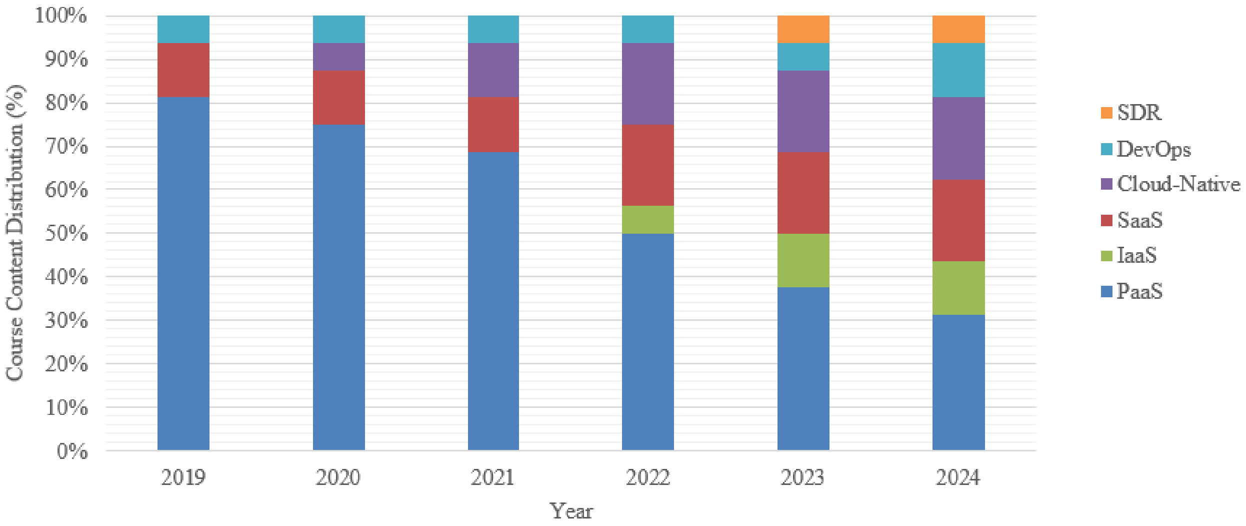

Figure 2. Evolution of course content distribution in cloud computing education (2019–2024).

Figure 2 illustrates the evolution of course content distribution in cloud computing education from 2019 to 2024, using a 100% stacked bar chart. This chart shows the allocation of course hours across six categories: PaaS, IaaS, SaaS, Cloud-Native, DevOps, and SDR. The vertical axis represents the percentage of total course hours dedicated to each category, highlighting changes in course emphasis over time. The data presented in this figure will be used to calculate entropy, measuring the diversity of course content distribution across the years.

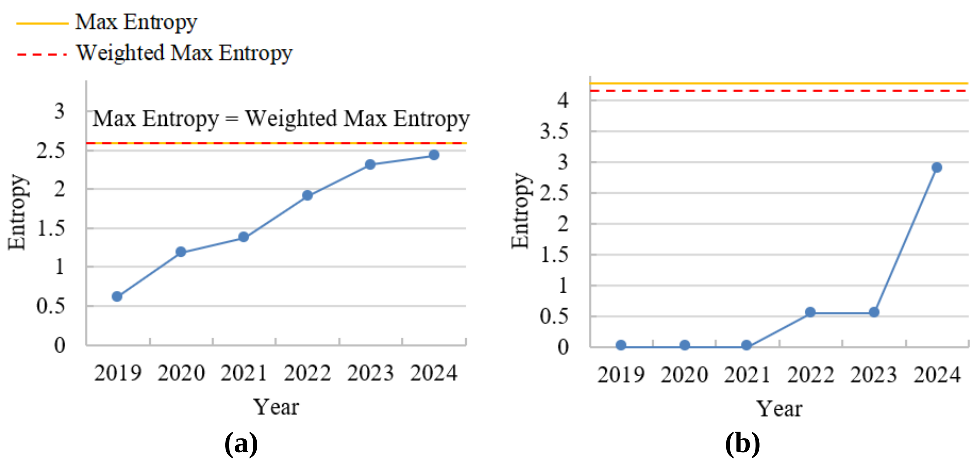

Figure 3. Entropy Changes Over Time for. **(a)** SDR-Dataset; **(b)** AI-Dataset.

Figure 3 displays the entropy values calculated from two datasets across the years 2019 to 2024. **Figure 3a** shows the entropy trend for the SDR-Dataset, indicating a gradual increase from 2019 to 2024, suggesting a steady diversification in the course content distribution over the years. The Weighted Max Entropy for the SDR-Dataset is marked as 2.585, providing a reference for the maximum potential entropy based on the weighted distribution of content modules.

Figure 3b illustrates the entropy trend for the AI-Dataset, where entropy remains low until 2023, followed by a sharp rise in 2024. The Weighted Max Entropy for the AI-Dataset is marked as 4.177, and the Max Entropy is 4.322, reflecting the diversity of programming languages taught in that year. The sudden rise indicates a significant expansion in the diversity of the course content.

6.3. Diversity efficiency index (DEI) calculation

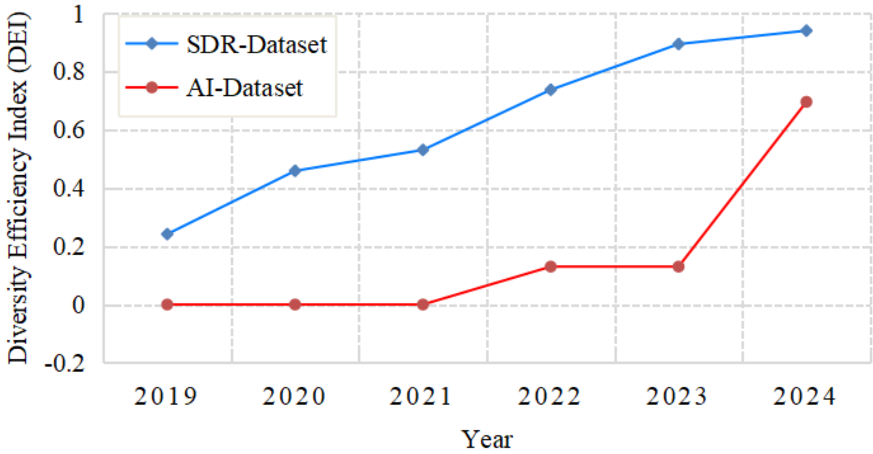

Figure 4. Trends of diversity efficiency index (DEI) for SDR-Dataset and AI-Dataset from 2019 to 2024.

In this section, we summarize the DEI calculations for the SDR-Dataset and AI-Dataset to evaluate the diversity and balance of course content over time. DEI is computed as the ratio of observed entropy to the maximum possible entropy, providing a normalized measure of content distribution efficiency.

- SDR-Dataset DEI: The DEI for the SDR-Dataset shows a steady increase from 2019 to 2024, indicating a more balanced and efficient distribution of content across various categories (PaaS, IaaS, SaaS, Cloud-Native, DevOps, SDR). This trend reflects the course content's diversification and even distribution among these topics.
- AI-Dataset DEI: The DEI for the AI-Dataset remains low until 2022, indicating less efficient distribution. However, a sharp rise in DEI is observed in 2023 and 2024, highlighting a significant diversification and integration of multiple programming languages into the course content. This increase is influenced by the weighted calculation of maximum entropy.

Figure 4 displays the DEI trends for both datasets over six years. The two series in the plot compare the evolution of diversity efficiency in the "Cloud Computing and Cloud Services" course (SDR-Dataset) and the "Algebraic Code and AI Framework" course (AI-Dataset). The SDR-Dataset shows a steady, gradual increase in diversity efficiency, reflecting continuous diversification and balanced content distribution. In contrast, the AI-Dataset exhibits a sharp rise in DEI after 2023, largely driven by the weighted calculation of maximum entropy, which accounts for the varying importance of different programming languages. These DEI trends offer valuable insights into the content evolution and instructional effectiveness of these courses, helping educators optimize curriculum design for more balanced and comprehensive learning outcomes.

6.4. Regression analysis and random forest analysis

6.4.1. Introduction of key variables and their impact

This subsection introduces the timing of cloud technology (X1) and LLMs (X2) usage and examines how these variables, along with year and series, affect the Diversity Efficiency Index (DEI). We will analyze how the variables—year, series, use of cloud technology (X1), and use of LLMs (X2)—influence the Diversity Efficiency Index (DEI) using regression and random forest models. The dataset applied for both analyses is detailed in **Table 4**, which includes the yearly DEI values and the associated indicators for cloud and LLMs usage across two courses from 2019 to 2024. In this table, the Series column is represented using one-hot encoding, where (1, 0) denotes the "Cloud Computing and Cloud Services" course (Series 1) and (0, 1) denotes the "Algebraic Code and AI Framework" course (Series 2). This encoding ensures that the courses are treated as categorical variables without implying any ordinal relationship between them.

6.4.2. Linear regression analysis

Linear regression was conducted to analyze the impact of the variables Year, Series, X1 (indicating the use of cloud technology, SDR), And X2 (indicating the use of LLMs) on the DEI.

Table 4. Summary of Yearly Diversity Efficiency Index (DEI) Values for Two Courses with Cloud and LLMs Usage (2019–2024).

Year	Series	X1(Use Cloud)	X2(Use LLMs)	DEI
2019	1, 0	1	0	0.241783005
2020	1, 0	1	0	0.458915023
2021	1, 0	1	0	0.530623079
2022	1, 0	1	0	0.737201821
2023	1, 0	1	1	0.894124431
2024	1, 0	1	1	0.940066454
2019	0, 1	0	0	0
2020	0, 1	0	0	0
2021	0, 1	0	0	0
2022	0, 1	0	0	0.13014622
2023	0, 1	0	0	0.13014622
2024	0, 1	1	1	0.69570029

Table 5 summarizes the linear regression results, providing insights into the significance and strength of these variables' relationships with DEI. The analysis highlights which factors most influence the efficiency of content diversity in

educational settings, serving as a basis for understanding their impact.

Table 5. Summary of linear regression results for analyzing the impact of year, series, use of cloud technology, and use of LLMs on the diversity efficiency index (DEI).

Variable	Coefficient	Standard Error	*t*-Statistic	*p*-Value	95% Confidence Interval (Lower)	95% Confidence Interval (Upper)
Intercept	−155.1606	45.4791	−3.4117	0.0113	−262.7015	−47.6197
Year	0.0769	0.0225	3.4177	0.0112	0.0237	0.1302
Series1	0	0	65535	N/A	0	0
Series2	−0.2599	0.1091	−2.3815	N/A	−0.5179	−0.0018
X1 (Use Cloud)	0.2187	0.1237	1.7673	0.1205	−0.0739	0.5113
X2 (Use LLMs)	0.1942	0.1022	1.8996	0.0993	−0.0475	0.4359

Figure 5 illustrates the coefficient impact of cloud technology (X1), Large Language Models (LLMs, X2), year, and course series on DEI growth. As shown, both cloud technology (X1) and LLMs (X2) contribute positively to the DEI, with cloud technology increasing DEI by approximately 0.2187 and LLMs by 0.1942. The year variable also has a positive but smaller contribution of 0.0769, while Series 1 has no observable effect (coefficient = 0). However, the negative coefficient for Series 2 (−0.2599) suggests a less favorable impact on DEI growth. The red dashed line represents the optimal DEI growth, indicating the maximum potential. This figure visually demonstrates the influence of these variables on content diversity efficiency, complementing the regression analysis results from **Table 5**.

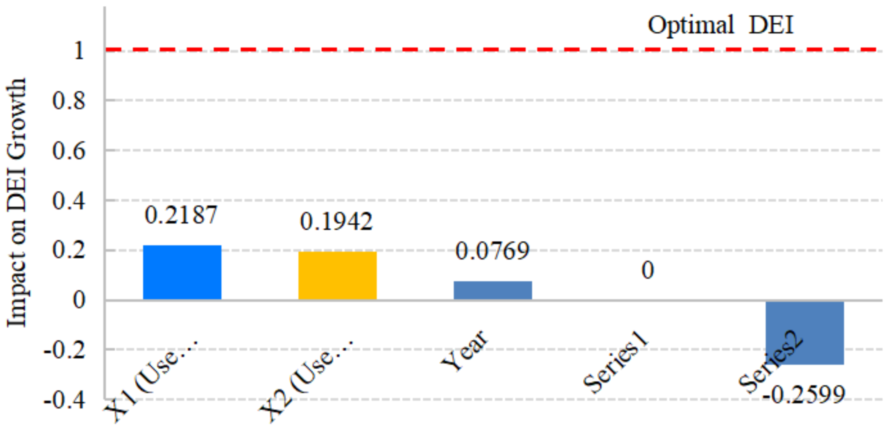

Figure 5. Coefficient impact on DEI Growth for use of cloud (X1), LLMs (X2), and other factors.

Figure 6 illustrates the Diversity Efficiency Index (DEI) trends for both the SDR-Dataset and AI-Dataset from 2019 to 2024, alongside their respective regression fitting curves. The solid lines represent the actual DEI values observed over time, while the dashed lines show the regression model's predictions based on the data trends. The regression fitting provides insight into the projected DEI growth, indicating how well the diversity and efficiency of course content is expected to evolve. For the SDR-Dataset, the regression curve closely follows the observed values, suggesting a steady and reliable improvement in course diversity. In contrast, the AI-Dataset shows a significant increase in DEI starting in 2023, and the regression fitting aligns with this sharp rise, reflecting the substantial integration of new content in recent years. This figure highlights the potential for future improvements in both datasets, guided by the regression analysis.

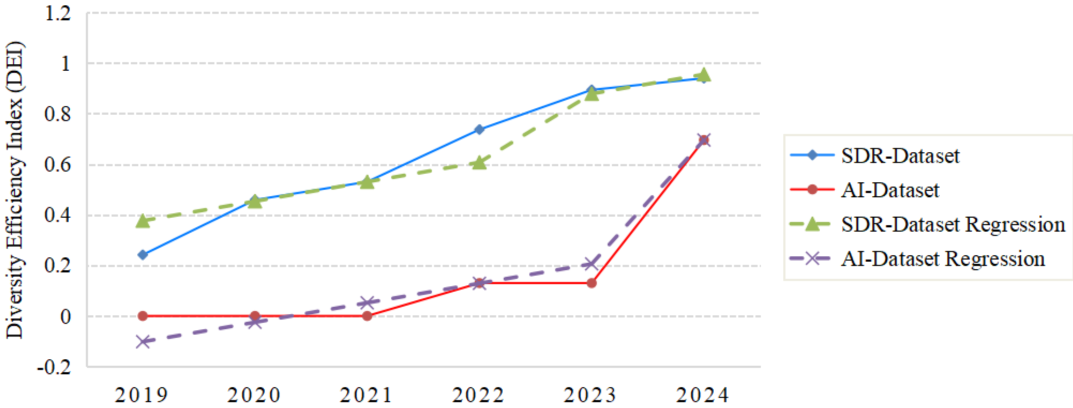

Figure 6. Trends of diversity efficiency index (DEI) and regression fitting for SDR-Dataset and AI-Dataset from 2019 to 2024.

6.4.3. Model evaluation and statistical significance

The statistical data provided by the regression analysis offers valuable insights into the overall fit and robustness of the model, as shown in **Tables 6** and **7**. The model's performance is supported by both the regression statistics and ANOVA analysis.

Table 6. Summary of regression statistics for the diversity efficiency index (DEI) model.

Statistic	**Value**
Multiple *R*	0.9803
R Square	0.9609
Adjusted *R* Square	0.7958
Standard Error	0.0886
Observations	12

Table 7. Summary of ANOVA for the diversity efficiency index (DEI) model.

Source	**df**	**SS**	**MS**	***F***	**Significance *F***
Regression	5	1.35306966	0.27061393	43.0750753	0.000126696
Residual	7	0.05497081	0.00785297		
Total	12	1.40804047			

As seen in **Table 6**, the Multiple *R* value of 0.9803 and *R* Square of 0.9609 indicate a strong correlation between the independent variables and the dependent variable, Diversity Efficiency Index (DEI). Specifically, the model explains 96.09% of the variance in DEI, demonstrating a highly accurate fit to the data.

Additionally, the Adjusted *R* Square of 0.7958 accounts for the number of predictors in the model. While this value is slightly lower than the unadjusted R

Square, it remains robust, reflecting the model's ability to generalize well despite increased complexity.

The model's Standard Error is 0.0886, suggesting a relatively low level of prediction error. This value indicates that the predicted DEI values are close to the actual observed values, further confirming the model's accuracy.

The ANOVA analysis shown in **Table 7** reinforces the statistical significance of the model. With an *F*-statistic of 43.08 and a Significance *F* (*p*-value) of 0.0001267, the model as a whole is statistically significant. The low *p*-value suggests that the independent variables included in the regression model have a meaningful impact on the DEI.

These statistical measures serve as important supplements to the regression results presented earlier, highlighting the model's goodness-of-fit, the relevance of the variables, and the overall significance of the model. The combination of high R Square, low standard error, and significant ANOVA results underlines the reliability of the findings and supports the robustness of the regression model used to analyze the DEI.

6.4.4. Random forest regression analysis

This subsection presents the results of the random forest regression analysis, which was used to capture non-linear relationships between the variables—year, series, use of cloud technology (X1), and use of LLMs (X2)—and the Diversity Efficiency Index (DEI). Random forest regression is particularly useful in understanding variable importance and their impact on DEI.

Table 8. Feature importance summary from random forest regression analysis for predicting diversity efficiency index (DEI).

Feature	**Importance**
Year	26.40%
Series 1	14.48%
Series 2	13.60%
X1 (Use Cloud)	21.01%
X2 (Use LLMs)	24.51%

To achieve optimal model performance, hyperparameters were tuned using random search, resulting in the following best parameters:

- The number of trees in the forest: 50
- Minimum number of samples required to split an internal node: 2
- Minimum number of samples required to be at a leaf node: 1
- Number of features to consider when looking for the best split: 1
- Maximum depth of the tree: 4

The best score obtained for the model was a negative mean squared error (MSE) of - 0.0326, indicating strong predictive performance for the dataset.

These feature importances are summarized in **Table 8**. The random forest regression analysis confirms that the year is the most impactful factor in determining DEI, followed by the use of cloud technology (X1) and the use of LLMs (X2). The findings suggest that technological integration in education significantly enhances content diversity and efficiency, providing valuable insights for curriculum development.

6.5. Results discussion

The regression and random forest analyses provide valuable insights into the factors that influence the Diversity Efficiency Index (DEI) in cloud computing and AI education. The linear regression analysis reveals that both cloud technology (X1) and LLMs (X2) have a positive effect on DEI, with cloud technology showing a stronger impact. Although LLMs also contribute to DEI growth, their effect is less significant in the regression model, reflecting that cloud technology plays a more central role in diversifying and balancing course content.

Despite the relatively small dataset used in this study, the results from both the regression and random forest analyses are acceptable. A key advantage of combining these two analyses is that while multiple linear regression effectively isolates the impact of each individual factor, the random forest analysis provides a more reliable assessment of the overall importance of each factor in a non-linear fashion. Additionally, the non-linear nature of random forest analysis requires significantly more computation time compared to linear regression, indicating that a much broader solution space was explored. This makes the combined approach especially valuable in drawing insights from smaller datasets.

The random forest regression analysis further confirms the dominant influence of year, which accounted for 26.40% of the total feature importance, followed by the use of LLMs at 24.51%, and cloud technology at 21.01%. These findings indicate that technological integration, particularly the use of cloud-based resources, is crucial for optimizing course content diversity and efficiency. The increasing DEI for the SDR-Dataset over the years suggests a steady improvement in balancing different content categories, reflecting an effective cloud education strategy. In contrast, the sharp rise in DEI for the AI-Dataset in 2023 and 2024 highlights the rapid diversification of programming languages, aligning with the evolving curriculum needs to accommodate a broader set of technical skills.

These results imply that while both cloud technology and LLMs contribute to enhancing educational diversity, cloud technology plays the more dominant role. The findings also demonstrate how targeted technological integration, particularly of cloud infrastructure and LLMs, can optimize curriculum development to better balance traditional and modern teaching methods, resulting in more comprehensive learning outcomes.

7. Benefits and evidence from educational reform project

7.1. Improved student outcomes

Data from our educational reform project shows that focusing on SDR significantly enhances students' ability to manage cloud resources, resulting in better application performance and resource utilization. Additionally, the use of LLMs like ChatGPT has shown potential in improving student engagement and creativity by providing real-time feedback and personalized learning experiences.

By applying SDR concepts and Lean principles, teaching efficiency has significantly improved, as evidenced by:

a) Broader Content in Cloud Computing Courses: The scope of technology taught has expanded from initially focusing solely on PaaS to now encompassing a comprehensive knowledge base that includes IaaS, PaaS, SaaS, DevOps, and cloud-native technologies.
b) Expansion in Programming Language Courses: There has been a dramatic shift from teaching just one programming language to now covering 20 languages simultaneously, enhancing both the breadth and depth of the curriculum.
c) Optimized Use of Teaching Time: Students no longer spend valuable class time configuring environments and resources. Instead, they focus on acquiring the most valuable knowledge, skills, and insights. For example, prior to SDR implementation, each student needed to configure their computers 5-6 times for setting up environments; now, this step is no longer required.

7.2. Establishing cloud computing course content with SDR concepts

The knowledge surrounding cloud platforms and cloud-native technologies has become extensive. Topics such as distributed applications, big data, and virtualization technologies are no longer appropriate for inclusion in cloud computing courses. These subjects are better suited for dedicated courses offered by computer science or software engineering departments. By applying SDR concepts, we can clearly delineate the knowledge specific to cloud platforms and services from those technologies that do not exclusively belong to cloud computing. This distinction allows for better curriculum planning and helps shape a clear and distinct understanding of cloud technology.

After several reforms and teaching practices, the fundamental course on cloud computing should include the following content:

a) Cloud Computing: This refers to the technology of software-defined resources.
b) IaaS (Infrastructure as a Service): This includes services related to computing, storage, networking, security, management, and Infrastructure as Code (IaC). This level should also cover hybrid cloud, multi-cloud, and edge technologies.
c) PaaS (Platform as a Service): Cloud services built and managed on top of IaaS using tools such as Azure Service Fabric and Kubernetes. Typical examples include App Services, serverless technology, as well as cloud-based middleware and databases. Once a product is cloud-based, its availability is generally described by a Service Level Agreement (SLA).
d) SaaS (Software as a Service): This includes various software functions accessed via API, with AI functionalities being a typical service. SaaS enterprise applications existed even before cloud platforms.
e) DevOps: This involves software engineering automation services enabled by the cloud.
f) Cloud-Native: This refers to large-scale application builders on the cloud, with Kubernetes and related technologies being typical examples.

These content areas appear to be relatively stable and suitable for teaching over the coming years.

8. Discussion

This study highlights the transformative potential of combining Software-Defined Resources (SDR) and Large Language Models (LLMs) in enhancing both the diversity and efficiency of educational content. By leveraging these technologies, educators can create more streamlined and effective teaching practices, which are not only applicable to cloud-based courses but also extend to courses outside the cloud domain. Below are several key points of discussion:

a) Lean Optimization in Teaching Processes with SDR and LLMs. The integration of SDR and LLMs leads to a more streamlined and lean teaching process, impacting not only courses that utilize cloud technologies but also those that do not. The ability to dynamically generate task-specific software tools through LLMs, combined with transparent cloud deployment, enables students to access these tools easily. This transparency significantly enhances teaching efficiency, as students no longer have to configure complex environments individually. Moreover, the use of these technologies is expected to transform not only educational processes but also a wide range of administrative and configuration, and knowledge and skill consumption tasks, making them more agile and efficient.

b) Significance of the Data Used in This Study. Although the dataset used in this study is not large, it spans six consecutive years of teaching practices, representing a longitudinal perspective on educational changes. The variations in the data over time are driven by significant, real-world factors, and the patterns observed are not random. These clearly defined trends make the analysis results highly valuable for understanding the impact of technological interventions in education. The distinct shifts in the Diversity Efficiency Index (DEI) provide meaningful insights into how SDR and LLMs influence educational outcomes. The analysis conducted in this study demonstrates that the integration of cloud technologies and Large Language Models (LLMs) in educational courses significantly enhances Diversity Efficiency Index (DEI) by approximately 0.2187 and 0.1942, respectively. These results provide a valuable benchmark for educators and course designers aiming to optimize teaching practices.

c) EDEA Framework as a New Approach. The Entropy-Based Diversity Efficiency Analysis (EDEA) framework offers a novel method for evaluating educational diversity efficiency. Unlike existing diversity analysis methods based solely on entropy, such as those proposed by Widiputera et al. (2017) and Ghosh and Basu (2023), the EDEA framework integrates both diversity and efficiency into its assessment. This makes it particularly relevant for analyzing the impact of SDR and LLMs, as these technologies enable dynamic and efficient content delivery. The EDEA framework can become a critical tool for assessing the efficacy of combining cloud technologies with LLMs in optimizing educational outcomes.

d) Generative AI Supported Agile Paradigm for Course Creation. The integration of cloud technologies (SDR) and LLMs will facilitate the development of numerous new courses across various fields. This is a shift from traditional workflows, as it introduces innovations in both teaching techniques and course design, paving the way for an agile teaching paradigm supported by generative AI. Unlike the agile methodologies discussed by López-Alcarria et al. (2019), which focus on flexibility in educational content design, the combination of LLMs and SDR will enable real-time dynamic process support. LLMs bring extensive knowledge of new techniques and information, while SDR provides an agile, scalable infrastructure that enables the transparent and dynamic deployment of process tools for any potential tasks or workflows across diverse domains, not limited to cloud computing or software development. Educators, equipped with their expertise in teaching, can leverage these technologies to continuously update and create new courses, ensuring university education stays aligned with industry needs. The lag between academia and the demands of the job market, which has long been an issue in fields like software development and engineering, will be significantly reduced as the integration of SDR and LLMs allows for constant innovation in teaching content. This combination promises to eliminate the educational gap, ensuring students receive up-to-date, relevant knowledge and skills across all domains.

9. Conclusion

9.1. Summary

This paper underscores the transformative impact of integrating Software-Defined Resources (SDR) and Large Language Models (LLMs) into university cloud computing curricula. These technologies contribute to the creation of dynamic, engaging, and efficient learning environments, aligning educational content with the evolving demands of the cloud computing industry. By incorporating real-world applications and hands-on experience with cloud tools, students are better equipped with the practical skills and knowledge required to navigate modern cloud environments. LLMs enhance the learning process by providing interactive and adaptive feedback, fostering a more personalized and efficient educational experience.

The introduction of the Entropy-Based Diversity Efficiency Analysis (EDEA) framework further strengthens this approach by offering educators a systematic tool for assessing and optimizing the diversity and efficiency of course content. Notably, the EDEA analysis revealed that the application of SDR (i.e., cloud technologies) and LLMs can each improve a course's Diversity Efficiency Index (DEI) by approximately one-fifth. This demonstrates the significant impact of integrating these technologies in enhancing both content diversity and efficiency.

While the influence of SDR and LLMs on course design and teaching is well-suited to be measured through diversity efficiency metrics, other benefits—such as the enhancement of students' confidence—are also evident, though more challenging to quantify. The EDEA framework empowers educators to ensure that their courses maintain a balanced and relevant knowledge base, adapting to rapid advancements in cloud computing and AI. This combined approach ensures that students engage with critical topics in cloud computing while benefiting from the efficiencies introduced by SDR and LLM technologies.

9.2. Future directions

As cloud computing continues to evolve, emerging technologies such as edge computing, serverless architectures, and the integration of artificial intelligence and machine learning (AI/ML) will shape the future landscape of cloud services. Educational curricula must remain dynamic and continuously updated to reflect these advancements. Collaboration with industry partners will be key to maintaining the relevance and applicability of curricula to real-world needs.

The ongoing integration of LLMs, such as Microsoft's Copilot in Azure, presents new opportunities for enhancing student engagement and deepening understanding of SDR and cloud technologies. Future research could extend the application of the EDEA framework to other fields, such as business management or healthcare, where diversity and efficiency play critical roles. Additionally, incorporating advanced machine learning models like deep learning could help capture more complex, non-linear relationships among variables, leading to more precise evaluations of diversity and efficiency in educational content.

By embracing these technological trends and integrating innovative tools, educators can ensure that students not only develop proficiency in current cloud technologies but also cultivate the adaptability needed to thrive in an ever-changing technological landscape.

Author contributions: Conceptualization, WP and YY; methodology, WP; validation, WP, HY; formal analysis, WP; investigation, YY; resources, WP; data curation, YY; writing—original draft preparation, WP; writing—review and editing, WP and YY; visualization, YY; supervision, WP; project administration, HY; funding acquisition, WP. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by: (a) Research on Lean Integration of Cloud and Local Environments Using Language Containers for Software Engineering Education. This project is part of the 2023 Higher Education Teaching Reform Project (10th Edition) of Sichuan University, Project Number SCU10337. (b) Research on Instant Distribution Service Containers for Teaching Cloud Servers. This project is part of the 2021 Annual Educational Teaching Reform Project of the College of Software Engineering, Sichuan University, Serial Number 18.

Acknowledgments: We would like to express our gratitude to Sichuan University for their support and funding, which have made this research possible. Additionally, we extend our appreciation to the faculty and students at the College of Software Engineering for their contributions and participation in these projects. Their insights and feedback have been invaluable in advancing our understanding and application of cloud computing in educational settings.

Conflict of interest: The authors declare no conflict of interest.

References

ACM. (2012). ACM Computing Classification System (CCS). Available online: https://www.acm.org/publications/class-2012 (accessed on 29 July 2024).

Apache Zeppelin. (2024). Available online: https://zeppelin.apache.org/ (accessed on 29 July 2024).

Bai, H. (2019). Programming Microsoft Azure Service Fabric (2nd ed.). PHI Learning.

Burns, B., Beda, J., Hightower, K., et al. (2022). Kubernetes: Up and running: Dive into the future of infrastructure (3rd ed.). O'Reilly Media.

ByteDance Cloud Native. (2024, August 2). KubeAdmiral v1.0.0 released! Available online: https://www.infoq.cn/article/LKV6kOvE9wSdjdSZGTlT (accessed on 20 August 2024).

CNCF Cloud Native Landscape. (2024). Available online: https://landscape.cncf.io (accessed on 20 August 2024).

Extance A. (2023). ChatGPT has entered the classroom: how LLMs could transform education. Nature, 623: 474–477. https://doi.org/10.1038/d41586-023-03507-3.

Gartner Research. (2016). SDx: Build and Market Software-Defined Data Center Offerings Primer for 2016. Available online: https://www.gartner.com/en/documents/3193219 (accessed on 29 July 2024).

Ghosh A., Basu A. (2023). On Entropy Based Diversity Measures: Statistical Efficiency and Robustness Considerations. N. Balakrishnan et al. (eds.) Trends in Mathematical, Information and Data Sciences, Studies in Systems, Decision and Control, 445: 267–283. https://doi.org/10.1007/978-3-031-04137-2_18.

IEEE Computer Society, ACM. (2013). Computer Science Curricula 2013: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. Available online: https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf (accessed on 29 July 2024).

IEEE Computer Society. (2014). Software Engineering Body of Knowledge (SWEBOK), Version 3.0. Available online: https://www.computer.org/education/bodies-of-knowledge/software-engineering/v3 (accessed on 29 July 2024).

Jiang W., Han H., He M., et al. (2024). ML-based pre-deployment SDN performance prediction with neural network boosting regression. Expert Systems with Applications, 241: 122774. https://doi.org/10.1016/j.eswa.2023.122774.

Kandiraju G., Franke H., Williams M. D., et al. (2014). Software defined infrastructures. IBM Journal of Research and Development, 58(2/3): 2:1–2:13. https://doi.org/10.1147/JRD.2014.2298133.

Koo S. (2020, June 10). Azure Files enhances data protection capabilities. Available online: https://azure.microsoft.com/en-us/blog/azure-files-enhances-data-protection-capabilities (accessed on 29 July 2024).

López-Alcarria, A., Olivares-Vicente, A., Poza-Vilches, F. (2019). A systematic review of the use of agile methodologies in education to foster sustainability competencies. Sustainability, 11(10), 2915. https://doi.org/10.3390/su11102915.

Microsoft Copilot in Azure. (2024). Available online: https://learn.microsoft.com/en-us/azure/copilot/overview%20?wt.mc_id=copilot_1b_webpage_gdc (accessed on 29 July 2024).

Mintz, S. (2021, February 7). 7 innovative approaches to course design. Inside Higher Ed. Available online: https://www.insidehighered.com/blogs/higher-ed-gamma/7-innovative-approaches-course-design (accessed on 28 September 2024).

NET Interactive. (2024). Available online: https://github.com/dotnet/interactive (accessed on 29 July 2024).

OpenAI. (2024). Introducing OpenAI o1. https://openai.com/o1 (accessed on 28 September 2024).

Rowan, J. A., Stanislav, T. A., Bernknopf, A. C., et al. (2022). Agile course design: Modeling flexibility, empowering engagement, and prioritizing community. Pedagogicon Conference Proceedings, 1. https://encompass.eku.edu/pedagogicon/2021/newdesigns/1 (accessed on 28 September 2024).

Scholl B., Swanson T., Jausovec P. (2019). Cloud native: using containers, functions, and data to build next-generation applications. O'Reilly Media, Inc.

Wei, J., Wang, X., Schuurmans, D., et al. (2022). Chain of thought prompting elicits reasoning in large language models. arXiv:2201.11903v6. https://doi.org/10.48550/arXiv.2201.11903.

Widiputera, F., De Witte, K., Groot, W., et al. (2017). Measuring diversity in higher education institutions: A review of literature and empirical approaches. IAFOR Journal of Education, 5(1), 47-63. https://doi.org/10.22492/ije.5.1.03.