

Developing Big Data Analytics Course for Non-ICT Major University Students

JongBeom Lim^a, ChangHak Moon^{b,*}

^a Smart Contents Major, Division of ICT Convergence, Pyeongtaek University, 111 Yongyi-Dong, Pyeongtaek-si, Gyeonggi-do, South Korea

^b College of Japanese, Hankuk University of Foreign Studies, 107 Imun-ro, Dongdaemun-gu, Seoul, South Korea

Corresponding author: *kudosmoon@gmail.com

Abstract— In the fourth industrial revolution's education era, there is no boundary between majors or subjects, and it is common for university students to enrol in information and communication technology (ICT)-related courses as convergence education blends different disciplines. Today's job market is getting more competitive and requiring higher skills in ICT and computational thinking. Since non-ICT major students rarely have programming experiences and knowledge in regular classes, teaching a big data analytics course for non-ICT major university students is not easy. Thus, it is vital to develop a curriculum that comprises easy-to-follow and easy-to-understand modules. In this paper, we develop a big data analytics course for non-ICT major university students. The proposed big data analytics course for non-ICT major students comprises two parts: (1) basic programming skill modules with step-by-step guidelines and (2) extension to big data analytics modules with laboratory exercises, with the five principal programming modules based on the Python programming language. First, our investigation discusses the suggestions and limitations of the big data analytics course for non-ICT major university students. Then, we recommend programming languages, integrated development environments (IDEs), and useful tools that help learners perform programming exercises and milestone projects. The learning objectives and course design models are carefully selected based on Bloom's taxonomy with six thinking levels and five.

Keywords— Big data analytics; convergence education; non-major course; programming course.

Manuscript received 14 Jul. 2020; revised 18 Feb. 2021; accepted 22 Apr. 2021. Date of publication 31 Dec. 2021.
IJASEIT is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.



I. INTRODUCTION

We are living in a digitized era, where digital data are expecting to reach 44 zettabytes (1,000⁷ bytes) by 2020, and 57 percent of the world population used the Internet [1], [2]. Moreover, new technologies such as 5G and the Internet of Things (IoT) are emerging, storing, processing, and analyzing big data are essential capabilities for job applicants [3], [4].

At the same time, many universities in Korea operate convergence education courses for whole students. These courses include programming languages (such as C, Java, Python, and Scratch) and computational thinking [5], [6]. Since convergence education blends different disciplines, students majoring in humanities (as well as information and communication technology (ICT) major students) are encouraged to enroll the convergence courses [7].

However, the convergence education courses, including programming languages and computational thinking, should be different from the courses for computer science (CS) major students [8], [9]. For non-ICT major students, the

convergence education courses should be easy to follow and understandable. In this paper, we design and propose a big data analytics course for non-ICT major students to extend the convergence education courses.

The proposed big data analytics course for non-ICT major students is composed of two parts: basic programming skill modules with step-by-step guidelines and extensions to big data analytics modules with laboratory exercises. Since the expected attendances of the big data analytics course rarely have programming language knowledge and experiences, the basic programming skill modules are necessary. Furthermore, it should have extensions to the big data analytics modules with appropriate laboratory exercises, which help perform milestone projects.

The rest of this paper is organized as follows. Section 2 describes our learning objectives and related work for designing and implementing the ICT-related courses in universities. Section 3 details our proposed big data analytics course for non-ICT major university students. The findings and recommendations about programming languages, integrated development environments (IDEs), and useful

tools for the big data analytics course are summarized in Section 4. Finally, Section 5 concludes the paper.

II. MATERIAL AND METHOD

This section provides learning objectives of the proposed big data analytics course for non-ICT major students and related work that addresses the ICT-related course developments in the universities.

A. Learning Objectives

The learning objectives of the proposed big data analytics course is to implement programming language practices and foster big data analytics skills.

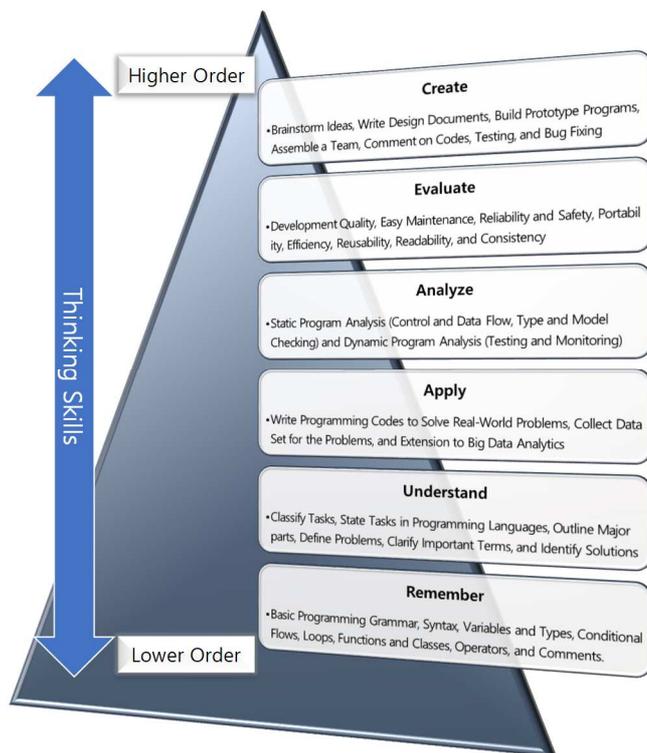


Fig. 1 The learning objectives based on bloom's taxonomy.

Fig. 1 shows the learning objectives based on the Bloom's taxonomy [10]. There are six thinking skills in the Bloom's taxonomy: remember, understand, apply, analyze, evaluate, and create.

Level 1 (remember) can be basic programming knowledges such as grammar, syntax, variables, and types, conditional flows (if, if-else, if-then-else clauses), concepts of functions and classes, operators (+, -, *, /, etc.) and comments. Level 2 is for understanding the programming language; it can be 'how to classify tasks of the problem,' 'state tasks in the programming language,' 'outline major parts of the tasks,' 'define the problem,' 'clarify important terms,' and identify the solution.'

Level 3 is to apply previously learned thinking skills to real-world problems in the programming language; it consists of 'write programming codes,' 'collect data set for the problem,' and 'extensions to big data analytics.' Based on the programming codes, it continues to level 4 to analyze the

programming codes. There are two kinds of analyses for this step: static program analysis and dynamic program analysis. The static program analysis is for control and data flows, type and model checking. The dynamic program analysis is for code module testing and monitoring.

The next thinking skill level is evaluation (level 5). It evaluates the programming codes based on development quality, easy maintenance, reliability, safety, portability, efficiency, reusability, readability, and consistency. Finally, the final level is created (level 6). It is the highest order of thinking skills and is composed of 'brainstorm ideas', 'write design documents', 'build prototype programs', 'assemble a project team', 'comment on the programming codes', 'testing', and 'bug fixing'.

Based on the proposed learning objects, we develop a big data analytics course for non-ICT major students. Because the expected attendances have no programming language knowledge and experiences, it is vital to implement basic programming skills and big data analytics simultaneously.

B. Related Work and Method

Dawson et al. [11] observed that the performance of non-CS students is worse than expected. In response, the authors developed a CS0.5 course for non-CS major students. Then, they evaluated the course in terms of total pass rates, the degree of students' satisfaction, and attitudes.

Mohamed [12] designed CS1 programming course for a mixed class (CS major students and non-CS major students). To implement the course, the author used the flipped classroom model [13] and pair-programming [14], which resulted in higher student engagement.

Ketenci et al. [15] revealed the relationship between characteristics of middle school students and their performance metrics (self-efficacy, learners' interest, computer and programming experience, and performance) in the computing course using the structural analysis based on the equation modeling.

Although several studies about CS and programming courses, designing a big data analytics course for non-ICT major students has room for development. For example, although the flipped learning or blended learning class [16] could improve the learners' satisfaction, we develop a big data analytics course by focusing on instructor-learners' interactions, which help provide motivation and inspiration to the university students.

III. RESULT AND DISCUSSION

In this section, we present our big data analytics course for non-ICT major university students. First, we introduce our course design model consisting of five procedures (glance, roadmap, design, teaching, and review). Then, we describe the details of the big data analytics course with our suggestions and recommendation.

A. Our Course Design Model

Making a good university course is not an easy task, and a well-made course can help enhance learning satisfaction while minimizing students' misleadingness by providing firm expectations. To this end, having a well-organized course design model can support building a solid university course.

Fig. 2 shows our course design model. The proposed course design model is composed of five procedures as below.

- *Glance*: In this procedure, we identify expected attendees. We should identify grade levels, majors, characteristics, background, academic performance, and personal attributes of students. In addition, we should reflect on the review of previous courses.
- *Roadmap*: In this procedure, we outline course topics and modules based on learning objectives. In addition, we establish the teaching approach (e.g., instructor-learner interactions, specific educational methods, flipped learning, blended learning, and e-learning).
- *Design*: In this procedure, we create weekly course plans and develop learning activities and assignments in detail. When building the course detail, we reflect learning objectives with thinking skills.
- *Teaching*: In this procedure, we teach based on the developed course materials. The important thing for this procedure is that we should collect students' feedback for every course plan to reflect them next time. At the same time, when teaching, we should adjust learning progress based on students' overall academic performance and achievement.
- *Review*: In this procedure, we keep the records of examinations and assignments to review the difficulty of assessments and testing results. Another crucial point for this procedure is perceiving students' needs and knowledge by interacting with students under the curriculum.

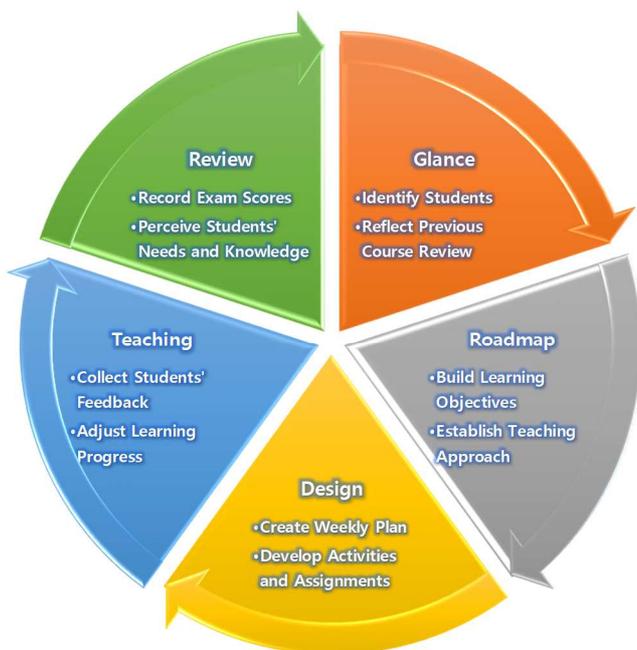


Fig. 2 The proposed course design model.

B. Details of Course Design

In this subsection, we describe the proposed big data analytics course for non-ICT major university students. Table 1 shows the outlined course modules. There are five modules of the course: (1) programming environment setup, (2)

computer programming basics, (3) functions, classes, and libraries, (4) programming for big data, and (5) mini-projects and exercises.

TABLE I
THE PROPOSED COURSE MODULES

Module	Description	Duration
#1	Programming environment setup	1 week
#2	Computer programming basics	3 weeks
#3	Functions, classes, and libraries	3 weeks
#4	Programming for big data	3 weeks
#5	Mini projects and exercises	4 weeks

The course outlines of the big data analytics for non-ICT university students are summarized in Table 2. We designed the course design for 16 weeks based on the proposed course modules described in Table 1. The first week is designed to introduce the course introduction and the programming environment setup. Since the big data analytics course uses a step-by-step approach, we encourage students to attend the first week. The programming environment setup includes installing Python and basic directions for the use of Interactive Python (IPython) and Jupyter Notebooks.

The second to fourth weeks are designed to deliver computer programming basics. This includes variables and assignment statements, arithmetic operations, input and output statements (input and print functions), and control/iterative statements (if, while, for, and Boolean operators). Because the big data analytics course is designed for non-ICT major university students, this course module should be easy-to-follow, while delivering enough background knowledge in a short period of time.

The fifth to seventh weeks are designed for the next level of programming skills in Python. In this module, we organize learning materials into three parts. That is, (1) functions (built-in functions, defining functions, and functions with parameters), (2) classes (defining user-defined classes, controlling access to attributes, and case studies), and (3) libraries (NumPy, Scipy, Matplotlib, Pandas, Sympy, Seaborn, Bokeh, Pygal and other big data-related ones).

After the mid-term examination, we teach big data-related frameworks and skills for three weeks (the ninth to eleventh weeks). The big data-related topics for the course are NoSQL [17], [18], NewSQL [19], [20], Apache Hadoop MapReduce [21], [22], Apache Spark [23], [24], and Apache Storm [25], [26].

Then, the students are asked to undertake programming projects for big data analytics based on the learning materials of the course (the twelfth to fifteenth weeks). The topics related to the programming projects are natural language processing (preferred in Japanese since students are majoring in Japanese), data mining Twitter, IBM Watson [27], [28] APIs, and machine learning.

In addition to the natural language processing project, the students are encouraged to use Japanese data sets. Finally, the final examination is scheduled for the sixteenth week. Note that in the examinations (mid-term and final), the students must use computers and program the code, then upload the written source code to the online learning management

system for grading. In other words, we conduct practical examinations rather than pencil-and-paper tests.

TABLE II
THE PROPOSED COURSE OUTLINES

Week	Contents	Note
1	Programming environment setup - Installing Python, using Interactive Python (IPython) and Jupyter Notebooks	Incl. course introduction
2	Computer programming basics I - Variables and assignment statements, and arithmetic operations	Use IDE, IPython, and Jupyter
3	Computer programming basics II - Input and output statements (input and print functions)	"
4	Computer programming basics III - Control statements (if, while, for, and Boolean operators)	"
5	Functions - Built-in functions, defining functions, and functions with parameters	"
6	Classes - Defining user-defined classes, controlling access to attributes, and case studies	"
7	Libraries - NumPy, Matplotlib, Pandas, Scipy, and Sympy	"
8	Mid-term examination	Practical exam.
9	Programming for big data I - NoSQL and NewSQL	
10	Programming for big data II - Apache Hadoop MapReduce	
11	Programming for big data III - Apache Spark	
12	Mini Project I - Natural language processing	Allow late turn in
13	Mini Project II - Data mining Twitter	"
14	Mini Project III - IBM Watson APIs	"
15	Mini Project IV - Machine learning	Not allow late turn in
16	Final examination	Practical exam.

Table 3 shows the details of laboratory exercises and projects corresponding to the course modules described in Table 2. Although the content of the first project is guided in the first week, we have the students perform the exercise at their homes for homework and exercises.

The details of four mini-projects are as follows: (1) TextBlob, tokenizing text, parts-of-speech tagging, and word frequencies for the natural language processing project; (2) Twitter APIs, Tweepy, searching tweets, and tweet sentiment analysis for the data mining Twitter project; (3) Watson SDK, Watson cloud services, and language translator for the IBM Watson APIs project; and (4) Scikit-Learn, k-nearest neighbors, regression, and clustering for machine learning and artificial intelligence projects.

TABLE III
DETAILS OF LABORATORY EXERCISES AND PROJECTS

Project	Description	Duration
#1 (first week)	Programming environment setup - Installing Python, using IPython and Jupyter Notebooks	1 week
#2 (twelfth week)	Natural language processing - TextBlob, tokenizing text, parts-of-speech tagging, and word frequencies	1-4 weeks
#3 (thirteenth week)	Data mining Twitter - Twitter APIs, Tweepy, searching tweets, and tweet sentiment analysis	1-3 weeks
#4 (fourteenth week)	IBM Watson APIs - Watson SDK, Watson cloud services, and language translator	1-2 weeks
#5 (fifteenth week)	Machine learning - Scikit-Learn, k-nearest neighbors, regression, and clustering	1 week

C. Discussion

In this subsection, we discuss the proposed big data analytics course for non-ICT university students. We use the Python programming language for the course since there is no obstacle for everyone who has no previous programming experience and knowledge.

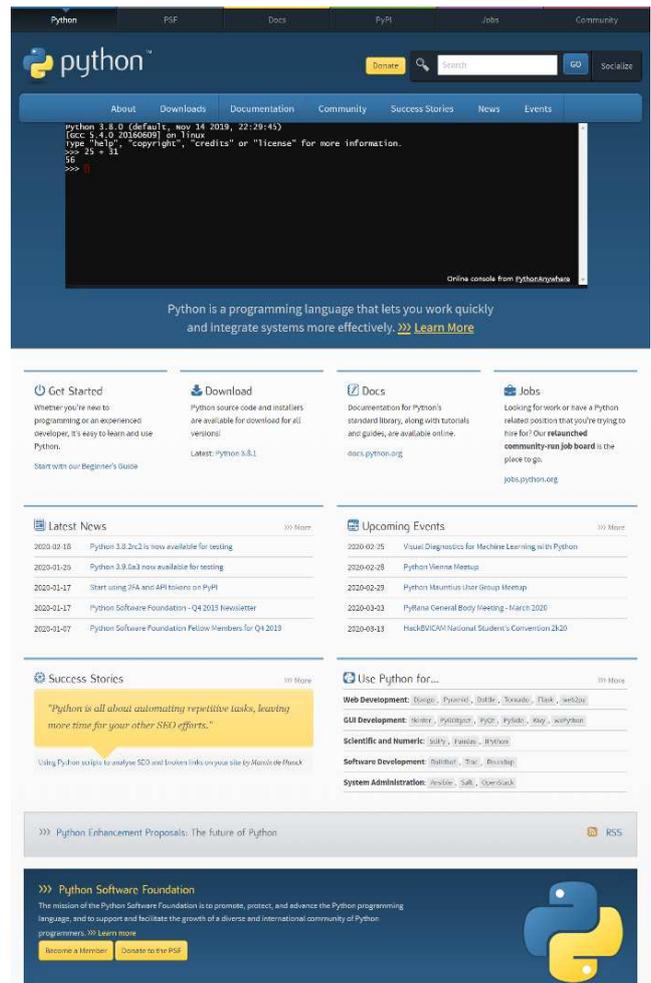


Fig. 3 An example of a Python shell in a web browser (<https://www.python.org/shell/>).

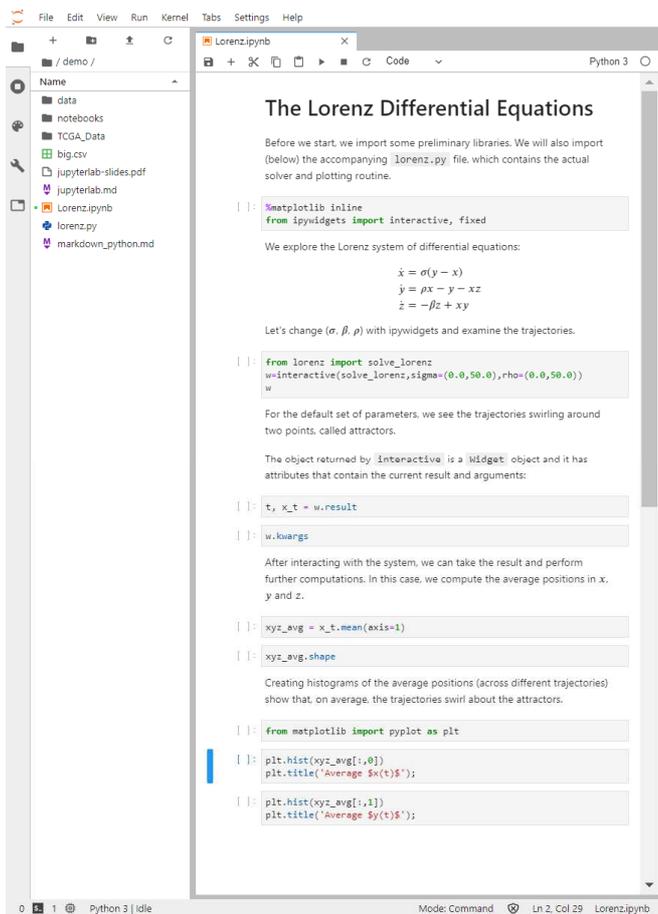


Fig. 4 An example of a Jupyter Notebook in a web browser (<https://jupyter.org/try>).

Furthermore, the Python-related tools that help study programming exercises are easy-to-use with no program installations when needed; the only required one is a web browser. Fig. 3 shows the Python shell in a web browser. With the Python shell in a web browser, anyone who wants to learn Python programming can access the programming environment. However, we teach the big data analytics course in a computer-based lecture room with Python and related tools installed for the learning effectiveness.

In addition to the big data analytics course lecture slides, we provide Jupyter Notebooks to the students, which can be easily accessed with a web browser as well. Fig. 4 shows an example of a Jupyter Notebook in a web browser. Like the Python shell in Fig. 3, Jupyter Notebooks can be used with and without program installations.

The above-mentioned Python-related tools are helpful since the students can utilize the tools with smart devices (e.g., tablets, smartphones, and laptops) and PC-based environments. Furthermore, with an Internet connection of the smart devices, the students can learn about the big data analytics course from anywhere.

We use a flexible turn in policy for milestone projects (the natural language processing, data mining twitter, and IBM Watson APIs projects). In other words, for the three milestones mentioned above projects, the students are allowed late turn in before the final examination.

We use the flexible turn-in policy to encourage the students to practice as much as possible without restrictions of firm deadlines. However, the period for the last milestone project

(the machine learning project) is one week. In other words, no late turn-in is allowed for the machine learning project since the next week is the final examination.

If we extend the deadline of the milestone projects to the final week (final examination), the students may suffer from examination nerves for those who have not finished the milestone projects. Therefore, we do not deduct points for the late turn-in to remedy examination nerves and stress about the milestone projects.

Our course design of the milestone projects is an individual, not team-based, approach to avoid the free-rider problem. The additional advantages of an individual approach are as follows: (1) the students can build responsibility and ownership for the milestone projects; (2) no free-rider problem occurs; and (3) the students will get credits so far forth as they go.

The disadvantage of an individual approach is that the students could not have cooperation experiences of a team-based approach. However, the advantages outweigh the disadvantage of the big data analytics course since many of the students do not have programming experiences and knowledge before the semester.

As far as learning materials are concerned, we did not specify a textbook for the big data analytics course for two reasons. One is that we cannot find a suitable textbook for the big data analytics course for non-ICT major university students. The other one is that we do not limit the students' potential learning experiences. In other words, if we specify a textbook for the big data analytics course, the students' learning area may be limited by the textbook.

Instead, we provide various learning materials to the students, such as lecture slides, Jupyter Notebooks, free online course URLs for Python and big data analytics, and source repositories. We conjecture that this approach helps extend the students' learning domains.

After the final examination and its grading, we provide the solutions to the milestone projects for self-grading and review. We observed that students are suffering from no solution policy for laboratory exercises and projects. Specifically, some students want to review and check their solutions and perform troubleshooting.

The potential issue for providing solutions to laboratory exercises and projects is that students may share the solutions with juniors in the next year's class. Our resolution is to modify the project problems every semester by reflecting on recent technology trends and developments.

IV. CONCLUSION

In this paper, we developed a big data analytics course for non-ICT major university students. The learning objectives and course design models are carefully selected based on Bloom's taxonomy with six thinking levels and five procedures. The proposed course is composed of five principal modules. Since the proposed big data analytics course is designed for non-ICT major university students who do not have previous programming experiences and knowledge, the curriculum is developed considering conciseness and efficiency for lectures. To this end, we used useful tools that help learners to perform programming exercises and milestone projects effectively.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2018R1D1A1B07045838) and by Hankuk University of Foreign Studies Research Fund of 2021. The author to whom correspondence should be addressed is ChangHak Moon.

REFERENCES

- [1] S. ElMassah and M. Mohieldin, "Digital transformation and localizing the Sustainable Development Goals (SDGs)," *Ecological Economics*, vol. 169, p. 106490, 2020, doi: <https://doi.org/10.1016/j.ecolecon.2019.106490>.
- [2] J. Kaur, R. Shedge, and B. Joshi, "Survey of Big Data Warehousing Techniques," *Inventive Communication and Computational Technologies*. Springer Singapore, Singapore, pp. 471–481, 2020.
- [3] H. B. Santoso and P. O. H. Putra, "Bridging the Gap between IT Graduate Profiles and Job Requirements: A Work in Progress," 2017 7th World Engineering Education Forum (WEEF). pp. 145–148, 2017, doi: [10.1109/WEEF.2017.8467146](https://doi.org/10.1109/WEEF.2017.8467146).
- [4] M. H. ur Rehman, I. Yaqoob, K. Salah, M. Imran, P. P. Jayaraman, and C. Perera, "The role of big data analytics in industrial Internet of Things," *Future Generation Computer Systems*, vol. 99, pp. 247–259, 2019, doi: <https://doi.org/10.1016/j.future.2019.04.020>.
- [5] S. Kim and H. Y. Kim, "A Computational Thinking Curriculum and Teacher Professional Development in South Korea," in *Computational Thinking in the STEM Disciplines: Foundations and Research Highlights*, M. S. Khine, Ed. Cham: Springer International Publishing, 2018, pp. 165–178.
- [6] A. Pürbudak and E. Usta, "Collaborative group activities in the context of learning styles on web 2.0 environments: An experimental study," *Participatory Educational Research*, vol. 8, no. 2, pp. 407–420, 2021, doi: [10.17275/per.21.46.8.2](https://doi.org/10.17275/per.21.46.8.2).
- [7] S. Siva, T. Im, T. McKlin, J. Freeman, and B. Magerko, "Using Music to Engage Students in an Introductory Undergraduate Programming Course for Non-Majors," *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. ACM, Baltimore, Maryland, USA, pp. 975–980, 2018, doi: [10.1145/3159450.3159468](https://doi.org/10.1145/3159450.3159468).
- [8] P. Dempster, D. Onah, and L. Blair, "Increasing academic diversity and inter-disciplinarity of Computer Science in Higher Education," *Proceedings of the 4th Conference on Computing Education Practice 2020*. Association for Computing Machinery, Durham, United Kingdom, p. Article 10, 2020, doi: [10.1145/3372356.3372366](https://doi.org/10.1145/3372356.3372366).
- [9] A. v Arzhanovskaya, E. A. Eltanskaya, and L. M. Generalova, "Convergence of Technologies in Education: New Determinant of the Society Development," *Lecture Notes in Networks and Systems*, vol. 155. Springer Science and Business Media Deutschland GmbH, Volgograd State University, Volgograd, Russian Federation, pp. 619–624, 2021, doi: [10.1007/978-3-030-59126-7_69](https://doi.org/10.1007/978-3-030-59126-7_69).
- [10] R. F. DeMara, T. Tian, and W. Howard, "Engineering assessment strata: A layered approach to evaluation spanning Bloom's taxonomy of learning," *Education and Information Technologies*, vol. 24, no. 2, pp. 1147–1171, 2019, doi: [10.1007/s10639-018-9812-5](https://doi.org/10.1007/s10639-018-9812-5).
- [11] J. Q. Dawson, M. Allen, A. Campbell, and A. Valair, "Designing an Introductory Programming Course to Improve Non-Majors' Experiences," *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. Association for Computing Machinery, Baltimore, Maryland, USA, pp. 26–31, 2018, doi: [10.1145/3159450.3159548](https://doi.org/10.1145/3159450.3159548).
- [12] A. Mohamed, "Designing a CS1 Programming Course for a Mixed-Ability Class," *Proceedings of the Western Canadian Conference on Computing Education*. Association for Computing Machinery, Calgary, AB, Canada, p. Article 8, 2019, doi: [10.1145/3314994.3325084](https://doi.org/10.1145/3314994.3325084).
- [13] H.-C. Hung, I.-F. Liu, C.-T. Liang, and Y.-S. Su, "Applying Educational Data Mining to Explore Students' Learning Patterns in the Flipped Learning Approach for Coding Education," *Symmetry*, vol. 12, no. 2, p. 213, 2020, [Online]. Available: <https://www.mdpi.com/2073-8994/12/2/213>.
- [14] J. S. Saltz and I. Shamshurin, "Exploring pair programming beyond computer science: a case study in its use in data science/data engineering," *International Journal of Higher Education and Sustainability*, vol. 2, no. 4, pp. 265–278, 2019, doi: [10.1504/IJHES.2019.103360](https://doi.org/10.1504/IJHES.2019.103360).
- [15] T. Ketenci, B. Calandra, L. Margulieux, and J. Cohen, "The Relationship Between Learner Characteristics and Student Outcomes in a Middle School Computing Course: An Exploratory Analysis Using Structural Equation Modeling," *Journal of Research on Technology in Education*, vol. 51, no. 1, pp. 63–76, 2019, doi: [10.1080/15391523.2018.1553024](https://doi.org/10.1080/15391523.2018.1553024).
- [16] A. Alammery, "Blended learning models for introductory programming courses: A systematic review," *PloS one*, vol. 14, no. 9, pp. 1–26, 2019, doi: [10.1371/journal.pone.0221765](https://doi.org/10.1371/journal.pone.0221765).
- [17] P. Atzeni, F. Bugiotti, L. Cabibbo, and R. Torlone, "Data modeling in the NoSQL world," *Computer Standards & Interfaces*, vol. 67, p. Article 103149, 2020, doi: <https://doi.org/10.1016/j.csi.2016.10.003>.
- [18] A. I. Sanka, M. H. Chowdhury, and R. C. C. Cheung, "Efficient High-Performance FPGA-Redis Hybrid NoSQL Caching System for Blockchain Scalability," *Computer Communications*, vol. 169, pp. 81–91, 2021, doi: [10.1016/j.comcom.2021.01.017](https://doi.org/10.1016/j.comcom.2021.01.017).
- [19] M. T. Özsü and P. Valduriez, "NoSQL, NewSQL, and Polystores," in *Principles of Distributed Database Systems*, M. T. Özsü and P. Valduriez, Eds. Cham: Springer International Publishing, 2020, pp. 519–557.
- [20] I. Astrova, A. Koschel, N. Wellermann, and P. Klostermeyer, "Performance Benchmarking of NewSQL Databases with Yahoo Cloud Serving Benchmark," vol. 1289. Springer Science and Business Media Deutschland GmbH, Department of Software Science, School of IT, Tallinn University of Technology, Akadeemia Tee 21, Tallinn, 12618, Estonia, pp. 271–281, 2021, doi: [10.1007/978-3-030-63089-8_17](https://doi.org/10.1007/978-3-030-63089-8_17).
- [21] R. R. Asaad, H. B. Ahmad, and R. I. Ali, "A Review: Big Data Technologies with Hadoop Distributed Filesystem and Implementing M/R," *Academic Journal of Nawroz University*, vol. 9, no. 1, 2020, [Online]. Available: <http://journals.nawroz.edu.krd/index.php/ajnu/article/view/530>.
- [22] B. Elghadyry, F. Ouardi, and S. Verel, "Composition of weighted finite transducers in MapReduce," *Journal of Big Data*, vol. 8, no. 1, 2021, doi: [10.1186/s40537-020-00397-4](https://doi.org/10.1186/s40537-020-00397-4).
- [23] G. Cheng, S. Ying, B. Wang, and Y. Li, "Efficient Performance Prediction for Apache Spark," *Journal of Parallel and Distributed Computing*, vol. 149, pp. 40–51, 2021, doi: [10.1016/j.jpdc.2020.10.010](https://doi.org/10.1016/j.jpdc.2020.10.010).
- [24] D. Lunga, J. Gerrand, L. Yang, C. Layton, and R. Stewart, "Apache Spark Accelerated Deep Learning Inference for Large Scale Satellite Image Analytics," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 271–283, 2020, doi: [10.1109/JSTARS.2019.2959707](https://doi.org/10.1109/JSTARS.2019.2959707).
- [25] J. I. Requeno, J. Merseguer, S. Bernardi, D. Perez-Palacin, G. Giotis, and V. Papanikolaou, "Quantitative Analysis of Apache Storm Applications: The NewsAsset Case Study," *Information Systems Frontiers*, vol. 21, no. 1, pp. 67–85, 2019, doi: [10.1007/s10796-018-9851-x](https://doi.org/10.1007/s10796-018-9851-x).
- [26] A. Muhammad and M. Aleem, "A3-Storm: topology-, traffic-, and resource-aware storm scheduler for heterogeneous clusters," *Journal of Supercomputing*, vol. 77, no. 2, pp. 1059–1093, 2021, doi: [10.1007/s11227-020-03289-9](https://doi.org/10.1007/s11227-020-03289-9).
- [27] A. Katiyar et al., "Timestamp Anomaly Detection Using IBM Watson IoT Platform," 2020, pp. 771–782.
- [28] K. Khalil et al., "Cognitive Computing for Human-Machine Interaction: An IBM Watson Implementation," vol. 1201 AISC. Springer, School of Mechanical and Manufacturing Engineering (SMME), National University of Sciences and Technology (NUST), H-12, Islamabad, 44000, Pakistan, pp. 400–406, 2021, doi: [10.1007/978-3-030-51041-1_53](https://doi.org/10.1007/978-3-030-51041-1_53).