Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025 ASEE
Paper ID #45388

Data Science in Environmental Engineering Curriculum

Prof. Ashraf Badir, Florida Gulf Coast University

Dr. Badir is a Professor in the Bioengineering, Civil Engineering, and Environmental Engineering Department at the U.A. Whitaker College of Engineering in Florida Gulf Coast University. He earned his B.Sc. (1982) in Civil Engineering and M.Sc. (1985) in Structural Engineering. He also holds a M.Sc. (1989) and a Ph.D. (1992) in Aerospace Engineering from Georgia Institute of Technology. Dr. Badir is a licensed Professional Engineer in Florida, and a civil engineering program evaluator for ABET.

Ahmed S. Elshall, Florida Gulf Coast University

Ahmed S. Elshall (https://orcid.org/0000-0001-8200-5064) is an Assistant Professor in the Department of Bioengineering, Civil Engineering, and Environmental Engineering at Florida Gulf Coast University, with a joint appointment at The Water School. His research focuses on sustainable groundwater management under uncertainty. He teaches courses in groundwater hydrology and environmental data science.

©American Society for Engineering Education, 2025

Data Science in Environmental Engineering Curriculum

Abstract

Data science is increasingly integral to various STEM domains, offering promising career opportunities across diverse engineering applications. Several studies have emphasized that the use of data science in the present engineering undergraduate curricula is mostly restricted to simple introductory subjects, usually with the use of Microsoft Excel. This restricts students' exposure to advanced data science techniques and big data applications. Additionally, there is a lack of clear guidance on how to strengthen the synergy between common water resources and environmental engineering courses and data science.

In this paper, the authors report on a recently introduced course entitled “Environmental Data Science,” at Florida Gulf Coast University (FGCU), incorporating the integration of Python programming for the analysis and visualization of environmental and water resources data. It is project-based, with self-directed learning opportunities. In this course, students learn how to gather and analyze data as part of the engineering design process, apply systems thinking to an engineering or societal phenomenon, collaborate with peers to find solutions, and effectively present solutions to an audience. Moreover, students are exposed to the introduction of the application of machine learning techniques to environmental datasets and Google Earth engine for remote sensing datasets.

This work will aim at reporting four main issues, namely (1) the unique components of the current integrated data science course, (2) an account of selected environmental engineering projects using Python, (3) a survey result collecting data on students’ perception about the new course, and (4) strategy to enhance synergy between data science and other engineering courses within the curriculum. It is anticipated that a thorough examination of the course's features, students' perceptions, and course synergy-enhancing factors would help to develop a guideline for data science curriculum development, implementation, and evaluation in environmental engineering.

1. Introduction

Data science has emerged as a transformative discipline, leveraging statistical, computational, and machine learning techniques to derive insights from complex datasets. In water resources and environmental engineering, where challenges often involve vast, multi-dimensional data, data science plays a pivotal role in addressing issues such as climate change, water resource management, pollution control, and sustainable development. Machine learning models and statistical techniques have been extensively employed to predict climate patterns, assess climate risks, and model atmospheric phenomena. Studies often focus on using neural networks to improve the accuracy of global climate models [1], employing unsupervised learning for anomaly detection in temperature and precipitation datasets [2], and integrating remote sensing data with machine learning for high-resolution climate simulations [3]. Data science aids in optimizing water distribution, forecasting water quality, and managing water scarcity. Key advancements include predictive analytics for hydrological modeling using time series analysis and deep learning [4], real-time monitoring systems employing Internet of Things (IoT) sensors

combined with big data analytics [5], and optimization algorithms for efficient groundwater and irrigation management [6]. Air, water, and soil pollution monitoring have seen significant improvements through data-driven approaches. Examples include Geographic Information Systems (GIS) integrated with machine learning to map pollution sources [7], linear regression models for estimating air quality using Python [8], and a green IoT based efficient system that detects and monitors the outdoor air pollution level [9]. Data science contributes to sustainable waste management by optimizing collection and transportation routes using clustering and network analysis [10], predicting waste generation trends through statistical modeling [11], and designing recycling processes based on material flow analysis supported by machine learning [12].

Python has become a cornerstone in data science applications for water resources and environmental engineering due to its extensive libraries, flexibility, ease of use, and multiple domain specific libraries that address water resources and environmental issues. First, key Python applications include climate analysis. Libraries like NumPy, SciPy, Pandas and Xarray are utilized for analysis of climate data, while Matplotlib, Seaborn and CartoPy enable effective visualization [13]. Packages like Xarray and CartoPy are particularly important for the analysis and vitalization labeled, multidimensional spatiotemporal data in NetCDF format. Specilied analysis tools are used in visualization, data manipulation, evaluation and diagnostics include xCDAT (Xarray Climate Data Analysis Tools) specialized analysis of model output data, and E3SM-Unified including the analysis tools of E3SM Earth system model. Second, for machine learning applications, packages such as TensorFlow, PyTorch, and Scikit-learn facilitate the development of predictive models for climate risk assessment and pollution monitoring [14]. Third, for geospatial analysis tools like GeoPandas and rasterio allow the integration and analysis of geospatial datasets, critical for land use and hydrological studies [15]. For environmental data science tools such as Earth Engine combines a multi-petabyte catalog of satellite imagery and geospatial datasets with Python API. Additionally, Geemap package integrates Earth Engine with Jupyter and Collab notebooks to automate workflow across python and Earh engine and provide advanced GIS and visualization features. Similarly, Python in ArcGIS Pro automate workflows in ArcGIS Pro with python. These tools and packages are widely water resources and environmental management. Fourth, in hydrological modeling tools like PySWMM and HydroPython support simulation and optimization of water distribution networks, stormwater management systems [16]. Other commonly used packages include FloPy that supports MODFLOW 6, MODPATH, MT3D-USGS, and SEAWAT for the modeling surface water-groundwater flow, particle tracking, reactive contaminant transport, and reactive transport modeling, respectively. Fight, these applications can leverage big data processing tools including Dask and PySpark to handle large-scale environmental datasets efficiently and enable real-time analytics [17]. Finally, advanced Artificial Intelligence (AI) algorithms are increasingly applied to solve non-linear and complex environmental problems. Transfer learning allows leveraging knowledge from related domains to enhance model performance [18]. The growing availability of environmental data, from satellites, sensors, and citizen science initiatives, necessitates scalable tools such as Hadoop and Spark for efficient processing and analysis [19].

The integration of data science into water resources and environmental engineering is essential for addressing the growing complexity of modern environmental challenges. First, incorporating data science into the environmental engineering curriculum is vital as industries are increasingly adopting analytical approaches. This raises demand for graduates proficient in tools such as Python, R, and machine learning frameworks to analyze large and complex environmental datasets. Embedding data science education into engineering programs ensures students are prepared to meet workforce demands. Second, beyond enhancing employment opportunities, data science education fosters interdisciplinary collaboration and promotes innovation across environmental engineering and related fields. By integrating concepts from computer science, mathematics, and environmental studies, students gain a holistic approach to solving environmental problems. Additionally, training in data visualization and decision-support systems enables students to communicate insights effectively, facilitating better stakeholder engagement and decision-making. Thus, as technology and environmental issues continue to evolve, providing students with a strong foundation in data science ensures they remain adaptable and relevant throughout their careers.

In the next sections, a detailed description of the newly introduced EDS course at FGCU is presented as a tool for other programs who are considering adding data science in their environmental engineering curriculum. The introduction of EDS early on in undergraduate curriculum is crucial in attracting future environmental engineers to the field of data science with its massive environmental applications.

2. Data Science Course in an Undergraduate Environmental Engineering Curriculum

The Environmental Data Science (EDS) course offered at FGCU introduces data science skills targeting applications in environmental engineering. The students participating in this course are juniors in environmental engineering majors. The course detailed herein was first introduced in Spring 2024 on campus face-to-face, with an enrollment of 12 students. The course meets twice a week, for a total of two-and-a-half contact hours, with 3 credit hours. We developed a Jupyter book for this course, which is freely available online (https://aselshall.github.io/edsbook). The is required as generic textbooks on Data Science with Python such as “Python for Data Analysis, O’Reilly, 3rd Edition” [20], which freely available online https://wesmckinney.com/book, do not cover specific tools and core package required for environmental data science such as Xarray, CartoPy, Earth engine, and Geemap.

This EDS course serves as an introduction to water and environmental data analysis using Python, a dynamic programming language with rich libraries for data science and scientific computing. These libraries include Pandas for spreadsheet analysis, Matplotlib for plotting, Plotly for interactive graphs, NumPy for scientific computing, Xarray for n-dimensional geospatial data analysis, and Cartopy for n-dimensional geospatial data visualization. Throughout this course, students not only learn how to use these tools, but also how to leverage Python for the analysis and visualization of water and environmental data. Data is explored from various sources such as NOAA, NASA, Copernicus, USGS, and Data.Gov. Data is handled with formats such as csv, shapefile, and NetCDF. Specialized resources tailored to students’ interests are utilized, such as Geemap Python-API of Google Earth Engine (multi-petabyte catalog of satellite imagery and geospatial datasets), CMIP6 datasets for climate projections, and FloPy

Python-API of MODFLOW for groundwater modeling. Additionally, students are introduced to the application of machine learning techniques to water and environmental datasets, employing machine learning libraries such as Scikit-learn and TensorFlow. More details about the course curriculum are provided in the syllabus: https://aselshall.github.io/eds

This elective course is designed for engineering and science students who have a genuine interest or practical necessity to delve into water and environmental data analysis using Python. The course is project-based and with self-directed learning opportunities. Accordingly, hands-on learning and practical applications are key criteria for assessing and evaluating students' progress in water and environmental data analysis using Python.

Figure 1 shows the EDS course within the undergraduate Environmental Engineering curriculum (spring of Junior year). The direct prerequisites for the course are Statistical Methods, and Computational Tools for Engineers (Excel and MATLAB), both with a minimum grade of C. Statistical Methods and Computational Tools for Engineers provide students with essential knowledge and skills to succeed in the EDS class. In Computational Tools for Engineers, students learn Excel spreadsheets and basic programming skills with MATLAB. As "distant" prerequisite, Introduction to the Engineering Profession discusses and fosters students research in topics related to the Grand Challenges in Engineering identified by the National Academy of Engineering (NAE). This is beneficial for students to find relevant project topics for the EDS course.

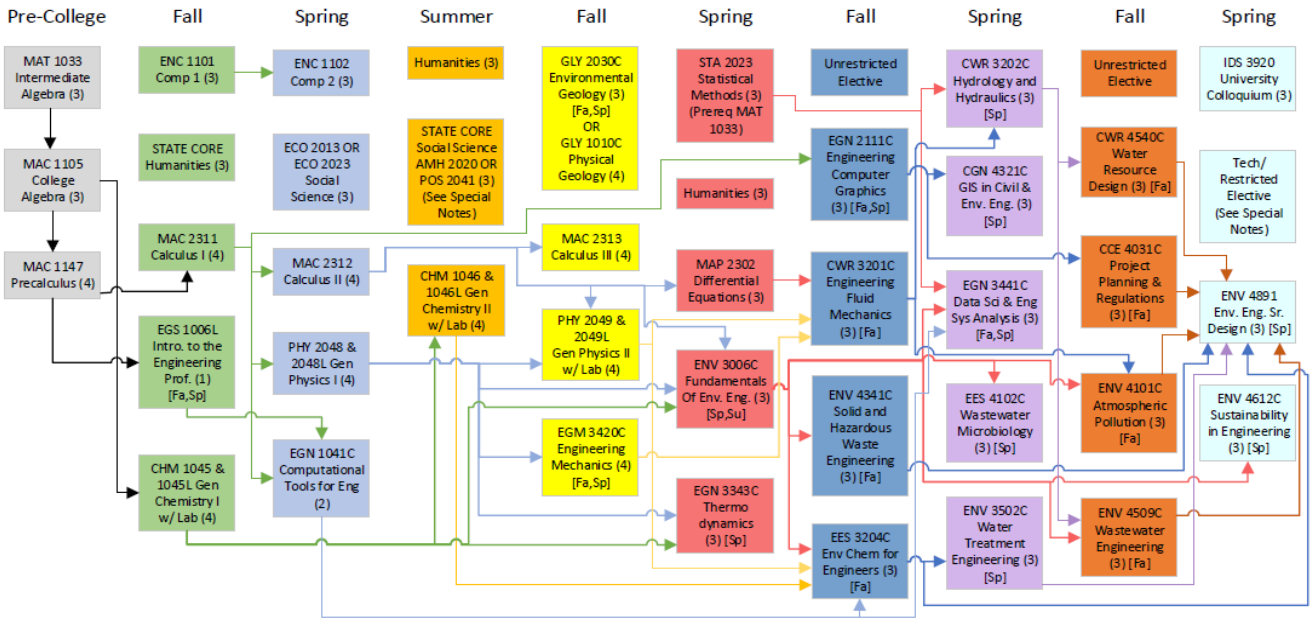

Figure 1. EDS course within Undergraduate Environmental Engineering Curriculum

3. Data Science Topics Covered, Delivery and Assessment

The newly introduced course aims for the students, after completing the course, to be able to (1) apply basic Python programming skills and libraries related to scientific computing such as NumPy, SciPy, Pandas, Matplotlib, Xarray and Catropy, (2) write Python codes to access,

manage, analyze, visualize, and share environmental data from various sources such as files, databases, dashboards, and web scraping, (3) use Python to analyze climate change data using datasets such as Coupled Model Intercomparison Project Phase 6 (CMIP6) data, (4) perform spatial analysis of environmental and water resources data using platforms and libraries such as Google Earth Engine, USGS ScienceBasePy, (5) execute projects using Python, encompassing data analysis and visualization to address environmental and water resources, and (6) effectively present solutions to peers and instructors. Appendix A shows the topics covered in the course. A JupyterBook was developed, which serves as a textbook for the course and includes all the lessons (https://aselshall.github.io/edsbook). In this course Python is used with Jupyterlab, which is an open-source interactive development environment that facilitates data science in Python through a web-based interface. Jupyterlab notebooks include code and markdown cells to allow rich metadata annotation. Other tools developed for this course include JupyterHub to allow student collaboration on the server.

The major assessments were homework, final exam, and a term project. Students learn a programming language by practice, accordingly, almost a bi-weekly assignment is posted on CANVAS. Assignments are designed to include self-directed learning opportunities. This is to give students the opportunity to explore and experiment with a variety of Python libraries and datasets of interest. The course also includes a semester-long project to perform data analytics with data of each team's choice to gain insights on an engineering or related system of interest to the team. Details of the course project are outlined in the following sections.

4. Term Project

4.1 Project Description

Students are expected to conduct a project using Python on a water or environmental issue of their choice. A research question or problem statement that guides the students' analysis is first developed. Students need to use Python and its various libraries to process, analyze, and visualize data to answer their research questions or solve an industry-or community-oriented problem. Students need to work in a group of two to four. Throughout the project, they can apply and deepen their skills in data analysis and visualization, critical thinking, and independent learning, in alignment with the project objectives.

4.2 Project Objectives

The project should demonstrate the ability to perform the following tasks.

1. Data analysis and visualization: Conduct a project using Python, encompassing data analysis, and visualization to address a water or environmental issue of your choice
2. Research question/problem statement: Develop a research question or problem statement that guides your analysis
3. Data acquisition and processing: Skillfully access, manage, wrangle, and analyze complex datasets with diverse data formats and from diverse sources such as Google Earth Engine, USGS ScienceBasePy, NOAA, Data.Gov, peer-reviewed articles, among others
4. Python libraries and tools: Apply appropriate Python libraries and tools for data manipulation, analysis, and visualization, such as Jupyter, Pandas, NumPy, Matplotlib, Xarray, Cartopy, Google Earth engine, Geemap, FloPy, and TensorFlow

5. Result evaluation and interpretation: Critically evaluate and interpret the results of your analysis, and draw meaningful conclusions that address your original research question or problem statement
6. Code documentation and best practices: Deliver clear and well-documented Jupyter notebook with Python codes following the best coding practices
7. Data communication: Effectively communicate your findings by documenting, visualizing, sharing and presenting your data insights in a clear and concise manner utilizing tools such as Jupyter notebook, GitHub, and Binder
8. Team collaboration: If applicable, contribute meaningfully to the project, and demonstrate effective teamwork skills including communication, collaboration, and conflict resolution
9. Reporting: Deliver a technically proficient final report following recognized academic or professional standards.
10. Presentation: Deliver a clear, organized, and effective class presentation to disseminate the project's findings to a wider audience

4.3 Schedule and Assessment

To ensure the project is feasible, teams schedule a meeting with the instructor to discuss their proposed project and obtain approval. Meetings are typically around 15 minutes but may be extended as needed. Teams need to obtain approval before they submit their project abstract.

The project is graded according to the following schedule:

- Project Abstract (10%): Two-page abstract outlining the proposed research question or industry-oriented problem, method, plan for implementation, and expected outcomes
- Interim Report (10%): Progress report detailing the work done so far, preliminary results, and any challenges encountered
- Final Report (50%): Comprehensive report detailing your method, analysis, results, and conclusions with the following outline
    - Cover page (with specific and informative project title)
    - Abstract (less than 200 words)
    - 1. Introduction (brief)
    - 2. Methods
    - 3. Results and discussion
    - 4. Conclusions
    - Data availability statement (containing link to your code, presentation and report repository)
    - References
    - Appendix (if needed)
- Project Presentation (20%): In-class or Eagle X presentation of the project
    - For in-class presentation, plan for 10-minute presentation that includes two minutes for questions and transition.
    - Students have the option to present at Eagle X for extra credit.
    - Winning an Eagle X Research Award guarantees an "A" grade (if you get a passing grade of 50% in the final).

- Peer Assessment (10%): Evaluation of students' contribution and their team members' contributions.

4.4 Project plan for implementation

It is recommended to focus on a question or hypothesis that does not require extensive data collection or complex modeling, which can be time-consuming. Given the available number of hours to spend on this project per week, teams need to have a realistic plan to complete their project on time. Once the team members decide on their research/management/business question or hypothesis, they can outline a plan for implementation, guided by the following example.

- Week 1: Project planning and data source identification. Define the scope and goals clearly and start searching for and gathering the required data.
- Week 2: Data acquisition and initial exploration. Download the needed datasets and conduct preliminary analysis to understand the data.
- Week 3: Data cleaning and preparation. Handle missing values, outliers, and any data inconsistencies.
- Week 4: In-depth data analysis. Use statistical methods or machine learning models if applicable to test your hypothesis or answer your research question.
- Week 5: Interpretation of results and report drafting. Begin to draw conclusions from your analysis and start writing your findings.
- Week 6: Finalizing the report/presentation. Refine your report and prepare the presentation and final report to share your project results.

4.5 Service learning

The students have the option to count service-learning hours while working on your term project. All undergraduate students at FGCU are required to complete 80 hours of service-learning as a graduation requirement. Service learning is a graduation requirement at FGCU. To qualify, the student term project should address a direct or research need for a community partner that is not-for-profit as detailed in the project assignment (https://aselshall.github.io/eds/HW/project#3-service-learning-optional). Students can search for a community partner that aligns with their interests. The students can use the FGCU Community Partners Database to identify Potential partners including the Florida Department of Environmental Protection, U.S. Army Corps of Engineers, FGCU, or similar organizations. Alternatively, students can contribute to an FGCU actively funded research projects such as projects related to groundwater climate resiliency, contaminant fate and transport, and water quality modeling using machine learning funded by U.S. Environmental Protection Agency and Florida Department of Environmental Projection. The students could continue working on their projects as research assistants.

4.6 Student projects

Project topics selected by students in Spring 2024 and 2025 varied in accordance with students' interests, and are shown below:

1. Evaluating downscaled precipitation data in Collier County (service learning)
2. Dynamic Interplay: Hurricanes and surface water salinity levels

3. Seasonal and spatial variations of nutrient concentrations in the Sanibel Slough: Implications for Watershed Management
4. Machine Learning of algal blooms prediction due to Lake Okeechobee discharge
5. Investigating the groundwater drawdown using FloPy package for MODFLOW groundwater modeling
6. Contaminant transport in complex groundwater regime
7. Aligning plant biodiversity data of the Naples Botanical Garden with Global Biodiversity Information Facility using pandas and ArcGIS Online Notebooks (service learning)
8. Plasma proteomics of loggerhead sea turtles (Caretta caretta) to establish diagnostic biomarkers for brevetoxicosis
9. Coral bleaching of the Floridian coral reefs (service learning)
10. Identification and analysis of upwelling events in the Gulf of Mexico and their connection to K. brevis blooms on the West Florida shelf using remote sensing data
11. Exploring machine learning models for red tide prediction in the Gulf of Mexico
12. Predictive modeling of red tide events using machine learning (service learning)

The data and codes of these projects are available on GitHub. The project assignment (https://aselshall.github.io/eds/HW/project) includes links to GitHub repositories of the students' projects. Several students are continuing their project during the summer as research assistants.

5. Student Feedback and Lessons Learned

A pre-semester survey was conducted at the beginning of the semester. The survey showed that 93% of the students had no previous knowledge of Python, 57% preferred to learn everything, including low level computer tasks like installing a library, and 86% of the students preferred active learning (more hands-on exercises). Students' perception about the EDS course was assessed by a formal anonymous survey administered towards the end of the semester. Students' responses to questions regarding the course curriculum are shown in Appendix B. The appendix lists students' responses to open-ended questions: (1) "*what did you like best about the environmental data science course*?" and (2) "*what did you like least about the environmental data science course*?" Most of the students liked to learn how to code and to discover the different usage of Python. Few students disliked running into errors while using the code and that some elements of Python are not intuitive, (3) "*Considering the time allocations and the content importance, how would you recommend adding, removing, merging, or extending some topics based on your experience*?" Students had a wide range of recommendations, varying from not changing anything in the course to extending class hours into Matplotlib, Xarray, Numpy, Pandas, Machine learning and GeeMap API. One student wished the course was over two semesters, and (4) *What are other thing(s) you would prefer/like to change about the way the course has been taught? What would you like the instructor to do differently? Additional comments/concerns/suggestions/compliments, etc*." One student wished for longer class time, two hours dedicated to each lesson, and adding steps to troubleshoot the code with respect to common errors. Based on the students' response, it would be essential to hire a TA and/or a Learning Assistant (LA) in this course.

Conclusions

Environmental data science is an emerging field that encompasses several STEM domains and offers exciting career prospects in a wide range of engineering applications. This paper presents

the unique components of a recently integrated Environmental Data Science course for environmental engineering junior students, including a description of its assignments and associated semester project. In addition, the paper provides a course map outlining how the existing undergraduate environmental engineering curriculum can be improved to include data science courses and application domains. Statistical methods and computational tools courses serve as prerequisites and are expected to provide students with essential knowledge and skills to succeed in the data science class. The data science class empowers the students with valuable tools to be used in their subsequent senior courses, especially the capstone senior design project.

It is the hope of the authors that the course presented would be beneficial to other programs envisioning introducing environmental data science into their curriculum. It is anticipated that a thorough examination of the course's features, and course synergy-enhancing factors would help to develop a guideline for environmental data science curriculum development, implementation, and evaluation in environmental engineering. Future research on the impact of data science on students' performance in subsequent courses, ABET student outcomes, graduation rates, and employability is warranted.

Funding statement

This work is funded by U.S. Environmental Protection Agency (EPA) Award Number AWD-03D09024.

References

[1] S. Scher, "Toward data-driven weather and climate forecasting: Approximating a simple general circulation model with deep learning," *Geophys. Res. Lett.*, vol. 45, pp. 12616–12622, 2018.

[2] Q. Liu, R. Klucik, C. Chen, G. Grant, D. Gallaher, Q. Lv, and L. Shang, "Unsupervised detection of contextual anomaly in remotely sensed data," *Remote Sens. Environ.*, vol. 202, pp. 75–87, 2017.

[3] V. Nourani, E. Foroumandi, E. Sharghi, and D. Dąbrowska, "Ecological-environmental quality estimation using remote sensing and combined artificial intelligence techniques," *J. Hydroinf.*, vol. 23, no. 1, pp. 47–65, 2021.

[4] C. Shen and K. Lawson, "Applications of deep learning in hydrology," in *Deep Learning for the Earth Sciences*, G. Camps-Valls, D. Tuia, X. X. Zhu, and M. Reichstein, Eds. 2021, doi: 10.1002/9781119646181.ch19.

[5] M. S. U. Chowdury, T. B. Emran, S. Ghosh, A. Pathak, M. M. Alam, N. Absar, K. Andersson, and M. S. Hossain, "IoT-based real-time river water quality monitoring system," *Procedia Comput. Sci.*, vol. 155, pp. 161–168, 2019.

[6] S. Afrifa, T. Zhang, P. Appiahene, and V. Varadarajan, "Mathematical and machine learning models for groundwater level changes: A systematic review and bibliographic analysis," *Future Internet*, vol. 14, no. 259, 2022, doi: 10.3390/fi14090259.

[7] A. A. Saim, *Machine Learning & Big Data Analyses for Wildfire & Air Pollution Incorporating GIS & Google Earth Engine*. Fayetteville, AR, USA: Univ. Arkansas, 2021.

[8] S. B. Sonu and A. Suyampulingam, "Linear regression-based air quality data analysis and prediction using Python," in *2021 IEEE Madras Section Conf. (MASCON)*, Chennai, India, 2021, pp. 1–7, doi: 10.1109/MASCON51689.2021.9563432.

[9] M. Ghoneim and S. M. Hamed, “Towards a smart sustainable city: Air pollution detection and control using Internet of Things,” in *2019 5th Int. Conf. Optimization and Applications (ICOA)*, Kenitra, Morocco, 2019, pp. 1–6, doi: 10.1109/ICOA.2019.8727690.
[10] C. S. de Morais, T. R. P. Ramos, M. Lopes, and A. Barbosa-Póvoa, “A data-driven optimization approach to plan smart waste collection operations,” *Int. Trans. Oper. Res.*, vol. 31, pp. 2178–2208, 2024, doi: 10.1111/itor.13235.
[11] M. Reza and M. Hassan, “AI-driven solutions for enhanced waste management and recycling in urban areas,” *IJSICS*, vol. 8, no. 2, pp. 1–13, Jun. 2023.
[12] N. Kroell, X. Chen, K. Greiff, and A. Feil, “Optical sensors and machine learning algorithms in sensor-based material flow characterization for mechanical recycling processes: A systematic literature review,” *Waste Manage.*, vol. 149, pp. 259–290, 2022.
[13] S. van der Walt, S. C. Colbert, and G. Varoquaux, “The NumPy array: A structure for efficient numerical computation,” *Comput. Sci. Eng.*, vol. 13, no. 2, pp. 22–30, 2011.
[14] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
[15] T. D. Bezabih, M. G. Glaety, D. A. Wako, and S. G. Worku, “Geospatial data analysis: A comprehensive overview of Python libraries and implications,” in *Ethics, Machine Learning, and Python in Geospatial Analysis*, 2024, pp. 22, doi: 10.4018/979-8-3693-6381-2.ch004.
[16] L. A. Rossman, *Storm Water Management Model User’s Manual*, 2010.
[17] M. Rocklin, “Dask: Parallel computation with blocked algorithms and task scheduling,” in *Proc. 14th Python in Sci. Conf.*, 2015, pp. 126–132.
[18] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2009.
[19] B-E. B. Semlali et al., “Hadoop paradigm for satellite environmental big data processing,” *Int. J. Appl. Environ. Inf. Syst.*, vol. 11, no. 1, pp. 23–47, 2020.
[20] W. McKinney, *Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter*, 3rd ed., 2022.

APPENDIX A

Topics covered in the course

0. Good Coding Practices
- GCP1 - Selecting “good” variable names
- Formatting “good” variable names
- Describing Code Readable
- Code Writing Scripts Assertions

1. Introduction to Environmental Data Science with Python
- Lesson 1 - Introduction to Environmental Data Science with Python
- Lesson 2 Getting Started

2 Python Basics
- Lesson 3 - Python Basics 1
- Lesson 4 - Python Basics 2 - Lists
- Lesson 5 - Python Basics 3 - Formatting

3 Python Programming
- Lesson 6 - Loops
- Lesson 7 - Conditional Statements
- Lesson 8 - Functions
- Lesson 9 - Script Files and Modules
- Exercise 3 - Python Programming

4 Pandas for Tabular Data
- Lessons 10 - 14 - Pandas Primer
- Exercise 4 - Pandas

5 Artificial intelligence (AI) Coding Assistance
- Lesson 15 AI coding assistance
- Exercise 5 - API-Key Module (Optional)

6 Data Science Workflow
- Lessons 16 – 17 Data Science Workflow
- Exercise 6 - AQI Data Preparation

7 NumPy for Scientific Computing
- Lessons 18-19 - NumPy Basics
- Exercise 7 - NumPy - Air quality data analysis

8 Matplotlib for Visualization
- Lesson 20 - 22 - Matplotlib Basics
- Exercise 8 Matplotlib - Air quality data visualization

9 Xarray and CartoPy for Labeled and Gridded N-dimensional Arrays
- Lesson 23 - Climate Data - CMIP6 and Remote Sensing Data
- Lesson 24-26 - Xarray basics
- Lesson 27 - CartoPy basics
- Exercise 9 - Sea surface height and red tides (Optional, Advanced)

10 Introduction to Google Earth Engine (optional)
- Lesson 28 - Intro to Google Earth Engine
- Exercise 10 - GeeMap, GEE, and Sentinel-2 Data

APPENDIX B

Students' perception about the EDS course was assessed by a formal anonymous survey administered towards the end of the semester. Below are a list of questions and students' responses.

What did you like best about the Environmental Data Science course?

"I liked to learn about pandas.
Learning how to answer my own questions in python. We may not know all of python but we know where to look.
I really liked learning a new skill in Python. I am glad that I was able to learn how to code and use such valuable skill.
Learning how to code using python and discovering different uses
I really enjoyed the instruction of this course, as well as the course material. I appreciated the lecture style, as well as the use of lecture/example problems.
I liked seeing the future of data management of engineering and the various tools utilized.
That Python makes long tasks much quicker.
What I like the most about this course is how I learned that python can do more things and have way more variety of packages than R.
This class was very enjoyable overall, and I enjoyed most the content on xarray because of the power and utility of accessing nc files and their relevance to my field of study.
Nothing.
I enjoyed learning about so many different packages in Python
Interaction with the content! Being able to follow along on Notebooks."

What did you like least about the Environmental Data Science course?

"There weren't any pre-requirements for this course and it is an advanced course. There should be more basic
programming classes in order to take this course.
Some elements of python are not intuitive
I wish that we had gotten through more material, and I wish it was a little more interactive. I also wish that things
were a little more organized, like with a study guide in Microsoft Word as opposed to being on Jupyter/Discord/GitHub because I am not as familiar with those platforms, and they were hard to access
and study and make my own.
the trial and error in trying to code
I wish there was more time to work on our final projects/more time in class. I appreciated that the last HW was
removed to permit more time to dedicate to the project.
I disliked how slow some of the lessons were.
Running into error after error when working with codes.
The only thing I liked the least is how hard python can be with very complex code.

The least enjoyable aspect of the course was the first few lessons on python basics, which went quite slow and could have been studied more efficiently at home.
Everything.
I feel like it was hard to cram all into one semester, it would be nice to break it up into two.
The fact that we had a final project and final exam."

Considering the time allocations and the content importance, how would you recommend adding, removing, merging, or extending some topics based on your experience.

"This is a good organization for the course. I would not change it.
I think it moved pretty well
I do not think Xarray was very helpful, so I would have removed that. I have found with my project I am using scipy a lot of statistical analysis, so I wish we could have covered that in this course. The introduction could have been accomplished in 1 lesson likely. I wish this course was over two semesters so we could have covered everything and a few things like Pandas, NumPy and Matplotlib in more depth.
I think the course moved at a good pace but I do wish there was more time allotted to learn about google earth
I would suggest less on Pandas, and more time allotted towards the special topics (Machine learning and GeeMap API).
I would shorten the panda lesson and python programming lesson by one lesson each to hit another subject in there.
It moved well.
I would say extend class hours in general for subjects like Matplotlib, Xarray, Numpy, and Pandas so we could cover more topics in the future.
I feel that a lesson just on AI coding assistance is not entirely necessary, especially because much of the utility of having an integrated AI API key in the jupyter notebook can be replicate by simplying accessing ChatGPT, Copilot, or any other LLM online. Additionally, it may be worth reducing the lessons on Matplotlib and instead teaching it alongside other lessons.
None.
I wish we were able to dive into Machine Learning
More time to work on Individual Project with professor guidance. Maybe about 2 classes allocated for that. AI coding assistance maybe removed or merged with another class.
Maintain the current course structure and offer a second course dedicated to machine learning, Google Earth engine, and case studies.
Maintain the current Python-centric approach but streamline topics to allocate more time for machine learning and Google Earth engine integration.;
Maintain the current course structure and offer a second course dedicated to machine learning, Google Earth engine, and case studies."

What are other things you would prefer/like to change about the way the course has been taught? What would you like the instructor to do differently? Additional comments/concerns/suggestions/compliments, etc.

"The instructor should stick to basics

Keeping troubleshooting until after class and only going over elementary topics one time

I would love the course to be more interactive. Maybe if the course was smaller, this would be easier, but I understand

that a lot of students want to take this course. As wonderful as Python is, I wish we also had a course specific to R,

since I feel that it is being used more widely especially for ecological studies.

Maybe we could have a lecture day and then a workshop day where we spend half the class walking through

How to do a code and the second half working with a partner to try to apply it to a case study.

I think more should be taught on the steps to troubleshoot the code with respect to the more common errors

I think how the instructor teaches the class is perfectly fine and shouldn't be changed.

I would prefer it if the lecture was more hands-on.

Only for the class hours to be 2 hours long, or one hour long if it is a MWF class.

The instructor’s enthusiasm is very welcomed and helps convey the importance of data science skills and programming. I would prefer a more self-learning focused class for the basics of learning packages and the basics of python in general, and more advanced topics/case studies to be covered in class.

Simpler lessons.