Technology Laboratories: Facilitating Instruction for Cyberinfrastructure Infused Data Sciences

Zhenhua He[1]
happidence@tamu.edu

Jian Tao[1]
jtao@tamu.edu

Lisa M. Perez[1]
perez@tamu.edu

Dhruva K. Chakravorty[1]
chakravorty@tamu.edu

ABSTRACT

While artificial intelligence and machine learning (AI/ML) frameworks gain prominence in science and engineering, most researchers face significant challenges in adopting complex AI/ML workflows to campus and national cyberinfrastructure (CI) environments. Data from the Texas A&M High Performance Computing (HPRC) researcher training program indicate that researchers increasingly want to learn how to migrate and work with their pre-existing AI/ML frameworks on large scale computing environments. Building on the continuing success of our work in developing innovative pedagogical approaches for CI-training approaches, we expand CI-infused pedagogical approaches to teach technology-based AI and data sciences. We revisit the pedagogical approaches used in the decades-old tradition of laboratories in the Physical Sciences that taught concepts via experiential learning. Here, we structure a series of exercises on interactive computing environments that give researchers immediate hands-on experience in AI/ML and data science technologies that they will use as they work on larger CI resources. These exercises, called "tech-labs," assume that participating researchers are familiar with AI/ML approaches and focus on hands-on exercises that teach researchers how to use these approaches on large-scale CI. The tech-labs offer four consecutive sessions, each introducing a learner to specific technologies offered in CI environments for AI/ML and data workflows. We report on our tech-lab offered for Python-based AI/ML approaches during which learners are introduced to Jupyter Notebooks followed by exercises using Pandas, Matplotlib, Scikit-learn, and Keras. The program includes a series of enhancements such as container support and easy launch of virtual environments in our Web-based computing interface. The approach is scalable to programs using a command line interface (CLI) as well. In all, the program offers a shift in focus from teaching AI/ML toward increasing adoption of AI/ML in large-scale CI.

[1] High Performance Research Computing, Texas A&M University, College Station, TX

DOI: https://doi.org/10.22369/issn.2153-4136/13/1/8

CCS CONCEPTS

•**CS**→Computer Science; •**Cybertraining**→training on using cyberinfrastructure; •**HPC**→high performance computing • interactive computing • training • containers

Keywords

Artificial intelligence, Machine learning, Cyberinfrastructure, Portal, Jupyter notebooks, Keras, Training, Scikit-learn, High performance computing, Pedagogy, HPRC[1]

1. INTRODUCTION

AI/ML and data science frameworks have been rapidly adopted in various fields of science and engineering. We find that most researchers, however, first interact with these technologies on their personal computing devices. As we continue to work through the impact of the COVID-19 pandemic and remote working scenarios, we see that researchers use these devices to simultaneously create reports and publications, attend workshops and conferences, and use teleconferencing techniques. These devices continue their routine roles of serving as a means of communication and entertainment as well. As we move toward larger community-shared data sets, researchers struggle to cope past the barriers of computing and managed storage on their devices. In a similar vein, the scientific community is moving toward workflows like Federated learning approaches in AI/ML and data science that allow for data to be kept local to a site while the model is trained and shared among the collaborating sites. With the data residing on-site, federated learning approaches meet the privacy and compliance needs of data but are currently not viable on the personal devices used by researchers. Taken together, we note that researchers' personal computing machines, regardless of the hardware specifications, are unable to accommodate AI and machine learning workloads at scale.

In this emerging scenario, researchers find their work limited by their choice of technology. To get more access to computing and storage, they are increasingly looking toward campus, regional, and national large-scale computing options. Over the last decade, and in particular in the last couple of years, the CI landscape has grown increasingly complex. Campus CI resources, commonly referred to as clusters, have evolved from operating traditional high-performance computing (HPC) environments to now simultaneously operating mixed environments that support batch schedulers, containers, virtual machines, cloud-bursting, composability via software (Kubernetes) and hardware (Liqid composability). Accelerators such as graphical processing units (GPUs) now run in half-, single -, mixed-, and double-precision modes. Data is stored in different formats and may be generated or streamed into systems. In such an environment, researchers new to

CI environments rapidly encounter an intimidating cyberinfrastructure (CI) landscape. It is perhaps ironic that these seemingly daunting technologies were developed to support AI/ML frameworks.

While the development of scientific computing applications and analysis of scientific data continue to be done on the command line, a broad swath of researchers by and large prefer graphical interfaces where they can interactively develop their applications and visualize their data at the same time. To facilitate the transition of researchers to this seemingly daunting environment, we have adopted interactive graphical interfaces like Jupyter Notebooks [1] and Google Colab where researchers can develop their applications and visualize their data at the same time. Such interactive development environments can be paired with web-based portals such as the Design Safe portal and Open OnDemand that offer researchers the ability to interact with CI in application- and systems-driven approaches, respectively [2]. Adoption challenges around compatibility and usability remain in these approaches as well. Furthermore, "quality of life" features that are common in AI/ML toolkits on public cloud-based environments are often not available on these resources. We have previously reported on our work in advancing the state of CI-training at Texas A&M. Texas A&M High Performance Research Computing (HPRC) offers over sixty training seminars, courses, workshops, and week-long computing-camps that support CI aspirations ranging from middle school students to CI professionals [3–10]. During the COVID-19 pandemic-imposed workplace changes, we have learned that pedagogical approaches commonly adopted in in-person environments do not translate to a virtual learning setting. During late spring 2020, we adopted "peer-mentored" and "peer-led" learning approaches that were coupled to persistent chat (SWEETER Slack workspace, 970+ researchers) and class videos offered in short, intermediate, and longer formats [9]. Continuing in our quest to offer researchers scaffolded techniques in computing environments, we recently reported how support for features like containers, virtual environments, quota allocations, and easy buttons on a Web-based computing interface led to a new approach toward introducing containerization [10]. We anticipate that these practices will help improve the adoption of FAIR (Findability, Accessibility, Interoperability, and Reusability) [14] and FEAT (Fairness, Ethics, Accountability, and Transparency) [15] standards in research computing.

Data from the HPRC ticketing systems, short course participation trends, and user feedback show that researchers need a new form of AI/ML training that focuses on adoption of AI/ML practices on our CI environments rather than courses that merely focus on teaching the AI/ML technologies themselves. Engaging, accessible, and interactive computing environments offer new opportunities for CI-infused teaching while simultaneously improving user adoption of new technologies. These environments allow researchers to move away from the CLI and explore other avenues to learn and interact with popular AI/ML frameworks such as Keras, TensorFlow, and Torch. These avenues are explored in popular industry-sponsored courses like the NVIDIA Deep Learning Institute, Intel's offering on AI/ML, or campus CI offerings like our courses that introduce TensorFlow and Pytorch. In this study, we report on our approach toward advancing AI/ML training on CI resources in an approach named the Technology Laboratories or tech-labs (Figure 1).

2. METHODS

We take a leaf from the pedagogical approach used in the tradition of Physical Science laboratory classes, during which exercises were stacked and techniques simultaneously taught while elucidating the concepts covered during classroom lectures. In these programs, the study material was divided into two distinct groups. Students first learned a foundational approach, typically how to use a scientific instrument. Then they used that instrument to conduct a series of experiments, each geared toward understanding and exploring scientific concepts. The tech-labs teach learners in a similar vein. They first introduce researchers to the CI technologies and then show them how to effectively work with their existing research workflow in a large-scale CI environment. Much like physics laboratories, we structured a series of exercises that first helped the researcher gain familiarity with the "instrument" or mode of computing. This could be the command line, a graphical user interface (GUI) for a scientific application, or a Jupyter Notebook. Here, we explore the use of Jupyter Notebooks.

The tech-labs are geared toward improving adoption of better CI practices in research environments. As such, they assume that a researcher is proficient in the AI/ML approaches that will be used. During these labs researchers learn how to use these techniques on clusters. Pre-requisites and learning objectives were identified to ensure that researchers do not misunderstand the purpose of these courses and manage participant expectations. Interested learners who were new to AI/ML techniques were directed toward community learning resources such as our short courses that covered these topics at an introductory level [16]. Toward facilitating a flipped classroom approach, curricular materials are pre-staged on the HPRC website along with relevant Git

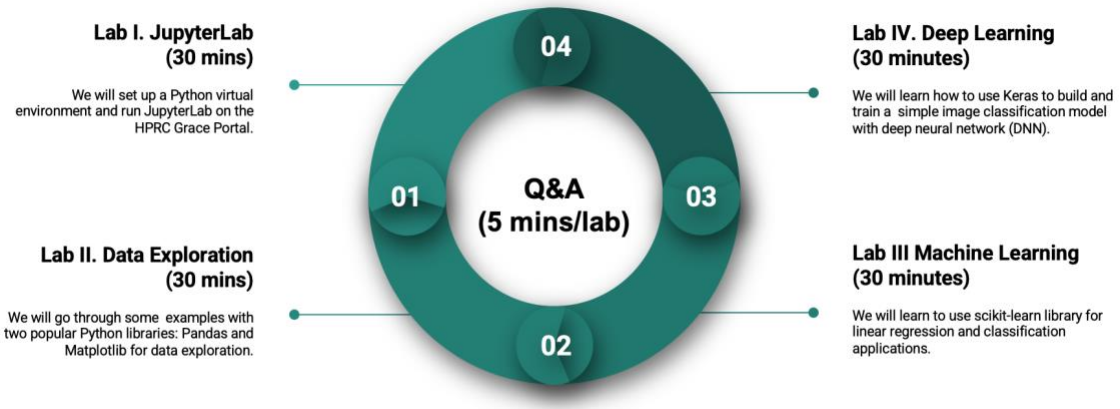

Figure 1. Structure of the AI/ML and Data Science Technology Laboratories.

repositories. Delivery is performed entirely using hands-on practices on CI resources, with no time spent exclusively on lectures introducing the AI/ML topics.

The class size is typically limited to 20 researchers to ensure that the instructor can assist the learners as they work through the exercises. Employing teaching assistants offers the option to build out the classroom. The tech-labs are divided into four components. This is described in the schematic presented in Figure 1. In the first session, researchers learn about the platform that will be used to access or interact with the CI resources. This session lasts for 15 minutes and is followed by three sessions of equal length that cover the AI/ML areas of interest. Each session is separated with a five-minute session during which participants can take a break or interact with the instructor.

2.1 Using the HPRC Open OnDemand (OOD) Web Portal

Noting the appeal of interactive computing approaches, especially to researchers who are not familiar with large-scale CI practices, we have developed a special version of the Open OnDemand platform for researchers at Texas A&M HPRC [10]. The adoption of this portal by researchers and its use in CI-training have been reported elsewhere. During the tech-labs, we first teach researchers how to use CI resources via the portal. Figure 2 shows the various applications available on the Texas A&M portal on the Terra supercomputer. Researchers learn that the home and scratch directories can be accessed, and files can be uploaded and downloaded easily. Researchers can view the status of their active jobs from the Jobs tab and access the shell from the Clusters tab, offering convenient access to the command line interface. Under the Interactive Apps tab, there is ready access to interactive applications such as Bio apps, GUI apps, JupyterLab, Jupyter Notebook, etc., for users to use.

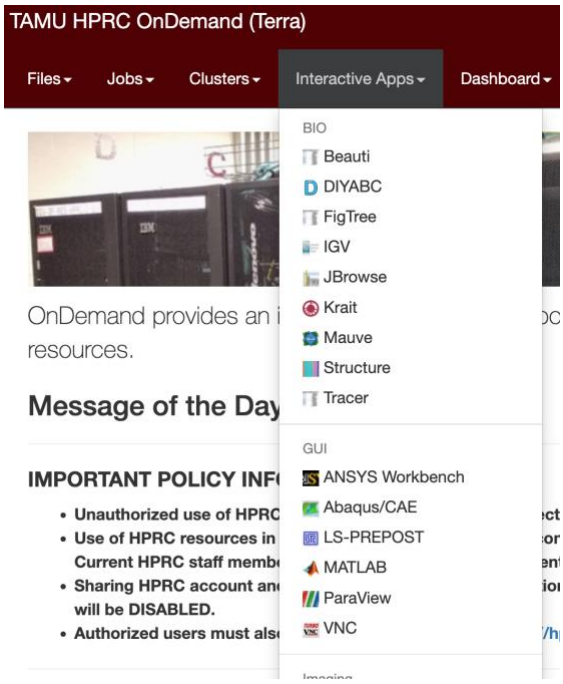

Figure 2. Some software applications built into the Texas A&M HPRC Open OnDemand (OOD) Portal for the Terra computing cluster.

2.2 Setting up Virtual Environments

Virtual environments offer a convenient mechanism for each project to have its own isolated environment on CI resources where its required dependencies can be installed regardless of what dependencies other projects require. We next teach the researchers how to use Anaconda commands to create virtual environments. These instructions are shown in Figure 3. For this project, we install scikit-learn and tensorflow packages into the virtual environment.

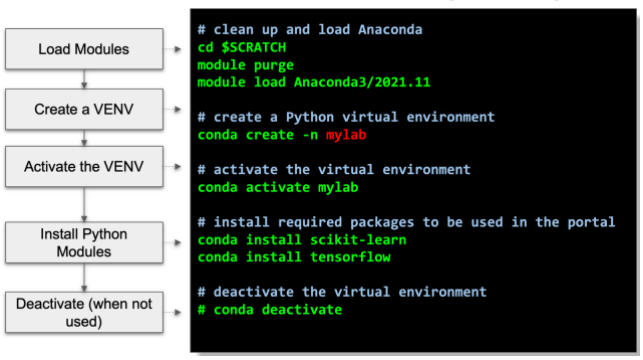

Figure 3. Creating a virtual environment with Anaconda commands.

2.3 Data Exploration Laboratory

In this section, we introduce some data science problems and two popular python libraries: Pandas and Matplotlib. Pandas is a Python library for data manipulation and analysis while Matplotlib is used for data visualization. These are taught using the JupyterLab interface as shown in Figure 4. During this session, researchers learn data manipulation skills. They are introduced to Pandas via examples of the two Pandas data structures — series and dataframe — and operations are provided such as retrieving and dropping entries, indexing, selecting, filtering, sorting, and ranking (based on the positions after sorting). The skills learned will be used in the exercises in the next session. Also, examples of using the Matplotlib object-oriented API to create figures and plots are taught. The advantage of using an object-oriented API becomes apparent when more than one figure is created or when a figure contains more than one subplot. Colormap, contour figures, surface plots, wire-frame plot, and contour plots with projections are also introduced. Colormaps and contour figures can be used to plot functions with two variables with the third dimension encoded. Passing a projection='3d' keyword argument to the add_axes or add_subplot methods can enable plotting 3D figures for better visualization.

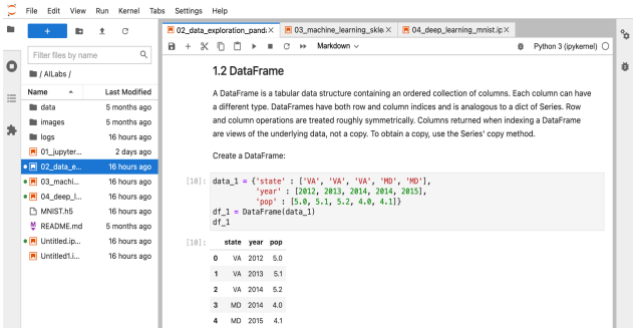

Figure 4. JupyterLab interface for the Data Exploration Lab.

2.4 Machine Learning Laboratory

In the machine learning session, we show the relationship between machine learning and artificial intelligence. [12]. During this lab, researchers learn how to use a machine learning library named Scikit-learn to work on regression, classification, and clustering problems with different algorithms. Linear regression is used to estimate the relationship among variables and predict a continuous-valued attribute of an object. It fits a linear model to minimize the sum of squares between the observations and predictions. In these exercises, researchers import the required libraries and models, generate an x-dataset with the numpy.linspace() function and a y-dataset, create an instance of the LinearRegression() model, and fit the model with x- and y-datasets. Researchers next check the coefficients for the linear regression model and the determination coefficient R2 with score() function, and visualize the data points and the best fit line. They finally work on a polynomial fitting exercise with linear regression by modifying the x-dataset with more polynomial terms with numpy.hstack().

Researchers are next shown how classification is used to identify the category an object belongs to based on a training dataset in which the membership of the objects is known. They are introduced to three concepts here: (i) Support vector machine (SVM) aims to find the hyperplane that separates binary sets with maximum margin to both classes. (ii) K-Nearest Neighbors (KNN) works based on the assumption that every data point belongs to the same class with the majority of its surrounding data points. In other words, it classifies a new data point based on similarity. (iii) Clustering is the task of separating the data points into groups such that the data points in the same groups are more similar than the data points in other groups. SVM and KNN classifiers are introduced in the lab's exercises. Students worked on a K-Means clustering exercise as well. In this exercise, the K-Means algorithm starts with a set of randomly selected centroids that are the beginning centers for every cluster and performs iterations to optimize the centroids' locations.

2.5 Deep Learning Laboratory

Deep Learning (DL) is included in the tech-labs since it finds applications in research. Because of its capability to handle high-dimensional data, DL is good for automatic feature extraction as well. During this session, researchers learn that deep learning is a subset of machine learning methods that are based on neural networks to improve algorithms by data and the relationship among AI, ML, and DL. This relationship is described in Figure 5. Three different DL methods including supervised learning, unsupervised learning, and reinforcement learning are explained. In supervised learning, the models are trained with labeled datasets. Regression and classification problems belong to this learning type. In unsupervised learning, the models are trained with unlabeled datasets. Clustering problems are in this category. In reinforcement learning, there are no training datasets, and it is about how agents take actions in an environment to maximize the reward.

During this session, researchers are taught to distinguish between traditional modeling that utilizes different numerical methods to solve the governing equations, and ML modeling that trains models with datasets and predicts unknown data with the trained models.

In the final session, researchers are introduced to Keras [17], the popular open-source neural network library. They perform an exercise during which they build a handwritten digits classifier [18]. The components include how to import the required libraries; load, split, and normalize the MNIST (Modified National Institute of Standards and Technology) [13] dataset; build a multi-layered neural network model with Sequential class; compile the model with an optimizer and a loss function; train the model with fit function on the train datasets; evaluate the model on test dataset; and predict. Finally, they study the images that were not correctly predicted and understand the potential reasons leading to these erroneous results.

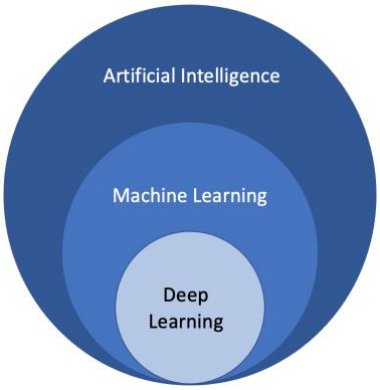

Figure 5. Relationship between AI, ML and DL.

The topics and in-class exercises covered during the tech-labs are summarized in Table 1. We have offered these technology laboratories every semester since 2020. These have been offered in hybrid (in-person and remote) and in remote (virtual, Zoom) formats.

Table 1a. Covered topics, in-class examples and exercises.

Topics and exercises covered	Topics and exercises covered
Create a virtual environment*	Write a dataframe to a file
Launch JupyterLab on OOD portal*	Matplotlib — line plot
Create a Pandas series	Matplotlib — subplots
Get the index and values of a series	Matplotlib — color map
Series indexing	Matplotlib — contour figures
Series filtering	Matplotlib — 3D figures
Series sorting	Case study — house market data*
Series mathematical operations	Linear regression
Create a Pandas dataframe	Polynomial fitting with linear regression
Specify the sequence of dataframe columns	SVM classification
Add a column to a dataframe	KNN classification
Retrieve a row from a dataframe	K-Means clustering
Retrieve a column from a dataframe	Principal component analysis

Table 1b (continuation of Table 1a).

Topics and exercises covered	Topics and exercises covered
Drop a row from a dataframe	Linear regression with a neural network library Keras
Drop a column from a dataframe	Build a handwritten digit classification model
Dataframe filtering	Train the model
Dataframe sorting	Evaluate the model performance
Load a file to a dataframe	Make predictions with the trained model

3. RESULTS

The tech-labs were offered by Texas A&M HPRC from fall 2020. Each tech-lab session ran for a duration of two-and-a-half hours. The labs assumed that researchers have prerequisite knowledge of the AI/ML and data frameworks. This marked a departure from our regular course of instruction where all materials were covered at an introductory level. The tech-labs were offered in hybrid (in-person and virtual) and virtual-only settings. Virtual classrooms were offered on the Zoom video-conferencing program, with dedicated support facilitated by the breakout room functionality. In spring 2022, the classes returned to a hybrid format. Table 2 lists the teaching modality and the number of registered students.

We have previously found that researchers are comfortable with this duration in sessions that include hands-on exercises. During the sessions, students created teams to complete the in-class exercises. Creating such teams offered the researchers opportunities for peer-learning as well as continued discussions after the classroom. To accommodate the taxing demands of learning via a "live" virtual session, we included several breaks. The Zoom "class" included a main session with several breakout rooms for teams' projects. With a view toward supporting researchers at different levels of learning, a competition-based approach was not adopted.

Table 2. Technology laboratories offered at Texas A&M.

Date offered	Modality	Registered attendees
2022-03-11	In-person	20*
2021-10-29	Hybrid	44
2021-06-02	Virtual	17
2020-10-30	Virtual	55

* registration for hybrid modality but course was held in an in-person modality.

During these tech-labs, learners were introduced to Jupyter Notebooks. This was followed by exercises in data exploration using Pandas and Matplotlib, machine learning using Scikit-learn for linear regression and classification applications, and Deep Learning using Keras to create and train a simple image classification model with a deep neural network (DNN). This is made possible by introducing a series of enhancements such as container support and easy launch of virtual environments in our Web-based computing interface. The approach can be readily expanded to support CI-adoption of Python-based AI/ML frameworks on the command line, AI/ML in Matlab, and other data science approaches. In all, the program offers a shift in focus from teaching AI/ML toward increasing adoption of AI/ML in large-scale CI.

4. CHALLENGES FACED AND LESSONS LEARNED

Owing to its format, the tech-lab encourages discussions between the instructor and participating researchers. It is not surprising that the tech-labs are a demanding teaching experience during which participant researchers ask several questions. This is particularly problematic during virtual sessions when using breakout rooms to have students communicate with each other. The instructor can join the breakout rooms to help answer their 'big' questions that they cannot solve together in a breakout room. It is, however, challenging for a single instructor to handle several breakout rooms. More teaching assistants should be trained for the short course to answer questions if the breakout room feature is used. Learning from our challenges in supporting all participating researchers, in summer 2021, we moved from a single instructor-supported instruction model to one that included two teaching assistants.

5. SUPPORTING INFORMATION

All training materials used in this study are available to the community via the Texas A&M HPRC website [16]. Videos and course recordings are available at the Texas A&M HPRC channel on YouTube. The community is invited to join the SWEETER slack workspace at https://hprc.tamu.edu/sweeter. Surveys and review exercises that will be developed as part of this longitudinal study may be requested from the author. Please send us feedback about your adoption experience via an email to help@hprc.tamu.edu.

6. ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation (NSF) award number 1925764, "CC* Cyberteam SWEETER," and NSF award number 2019129, "MRI:FASTER," NSF award number 1730695, "CyberTraining: CIP: CiSE-ProS: Cyberinfrastructure Security Education for Professionals and Students", NSF award number 2019136, "CC* BRICCs: Building Research Innovation at Community Colleges," and NSF award number 1829799, "Cybertraining: CMS3."

7. REFERENCES

[1] Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B. E., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J. B., Grout, J., Corlay, S., Ivanov, P., Avila, D., Abdalla, S., Willing, C. and Jupyter development team. 2016. Jupyter Notebooks — a publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. F. Loizides & B. Schmidt (Eds.), 87–90. DOI: https://doi.org/10.3233/978-1-61499-649-1-87

[2] Hudak, D., Johnson, D., Chalker, A., Nicklas, J., Franz, E., Dockendorf, T. and McMichael, B. L. 2018. Open OnDemand: A web-based client portal for HPC centers. *Journal of Open Source Software* 3, 25 (May 2018), 622. DOI: https://doi.org/10.21105/joss.00622

[3] Texas A&M High Performance Research Computing Website. Retrieved from https://hprc.tamu.edu

[4] Chakravorty, D. K., Pennings, M., Liu, H., Wei, Z., Rodriguez, D. M., Jordan, L. T., McMullen, D., Ghaffari, N. and Le, S. D. 2019. Effectively Extending Computational Training Using Informal Means at Larger Institutions. *Journal of Computational Science Education* 10, 1 (Jan. 2019), 40–47. DOI: https://doi.org/10.22369/issn.2153-4136/10/1/7

[5] Chakravorty, D. K., Pennings, M., Liu, H., Wei, Z., Rodriguez, D. M., Jordan, L. T., McMullen, D., Ghaffari, N., Le, S. D., Rodriquez, D. and Buchanan, C. 2019. Evaluating Active Learning Approaches for Teaching Intermediate Programming at an Early Undergraduate Level. *Journal of Computational Science Education* 10, 1 (Jan. 2019), 61–66. DOI: https://doi.org/10.22369/issn.2153-4136/10/1/10

[6] Seo, J. H., Bruner, M., Payne, A., Gober, N., McMullen, D. and Chakravorty, D. K. 2019. Using Virtual Reality to Enforce Principles of Cybersecurity. *Journal of Computational Science Education* 10, 1 (Jan. 2019), 81–87. DOI: https://doi.org/10.22369/issn.2153-4136/10/1/13

[7] Chakravorty, D. K., Pennings, M., Liu, H., Thomas, X., Rodriguez, D. and Perez, M. 2020. Incorporating Complexity in Computing Camps for High School Students — A Report on the Summer Computing Academy Program at Texas A&M University. *Journal of Computational Science Education* 11, 1 (Jan. 2020), 12–20. DOI: https://doi.org/10.22369/issn.2153-4136/11/1/3

[8] Chakravorty, D. K. and Pham, M. T. 2020. Evaluating the Effectiveness of an Online Learning Platform in Transitioning from High Performance Computing to a Commercial Cloud Computing Environment. *Journal of Computational Science Education* 11, 1 (Jan. 2020), 93–99. DOI: https://doi.org/10.22369/issn.2153-4136/11/1/15

[9] Chakravorty, D. K., Perez, L. M., Liu, H., Yosko, B., Jackson, K., Rodriguez, D., Trivedi, S. H., Jordan, L. and Le, S. 2021. Exploring Remote Learning Methods for User Training in Research Computing. *Journal of Computational Science Education* 12, 2 (Feb. 2021), 11–17. DOI: https://doi.org/10.22369/issn.2153-4136/12/2/2

[10] Lawrence, R., Pham, T. M., Au, P. T., Yang, X., Hsu, K., Trivedi, S. H., Perez, L. M. and Chakravorty, D. M. In press. Expanding Interactive Computing to Facilitate Informal Instruction in Research Computing. *Journal of Computational Science Education*.

[11] Dirk Merkel. 2014. Docker: lightweight linux containers for consistent development and deployment. *Linux Journal* 2014, 239 (Mar. 2014), 2. Retrieved from https://dl.acm.org/doi/10.5555/2600239.2600241

[12] Arthur L. Samuel. 1959. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development* 3, 3 (Jul. 1959), 210–229. DOI: https://doi.org/10.1147/rd.33.0210

[13] Yann LeCun, Corinna Cortes and Christopher J. C. Burges. MNIST handwritten digit database. Retrieved from http://yann.lecun.com/exdb/mnist/

[14] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* 3 (Mar. 2019), 160018. DOI: https://doi.org/10.1038/sdata.2016.18

[15] Ayanna Howard, Jason Borenstein and Kinnis Gosha. 2019. NSF-funded Fairness, Ethics, Accountability, and Transparency (FEAT) Workshop Report. In *NSF Workshop Reports* (Oct. 2019). Retrieved from https://par.nsf.gov/servlets/purl/10139705

[16] Texas A&M High Performance Research Computing. Short Courses. Retrieved from https://hprc.tamu.edu/training/

[17] Keras: the Python deep learning API. Retrieved from https://keras.io/

[18] GitHub - happidence1/AILabs. Retrieved from https://github.com/happidence1/AILabs