Unified Forecast System Student Engagement Plan

Alekya Srinivasan
Undergraduate, The Pennsylvania State University
First Student Ambassador of the Unified Forecast System
2023 William M. Lapenta Intern

Jennifer Vogt, Krishna Kumar, Maoyi Huang, Neil Jacobs
NOAA/OAR/Weather Program Office/EPIC, UCAR

https://doi.org/10.25923/093y-sk56

Table of Contents

Acknowledgements

This Unified Forecast System Student Plan was established through the 2023 William M. Lapenta Student Internship Program and staff. I would like to express my deepest appreciation to Jennifer Vogt, Dr. Maoyi Huang, Dr. Krishna Kumar, and Dr. Neil Jacobs for mentoring and guiding me. I would also like to acknowledge Aaron Jones, Laura DeHaan, Dr. Gina Eosco, Dr. Hendrik Tolman, and the Earth Prediction Innovation Center Contract and Program teams for their continued support throughout this project. I was welcomed with open arms into NOAA's Weather Program Office, and I would like to extend my gratitude towards Dr. Dorothy Koch and Dr. John Ten Hoeve. This endeavor would not have been possible without Keven Blackman, Chief Engineer at Raytheon, and Kristopher Booker.

Introduction

In the world of atmospheric, climate, environmental, and oceanic science, data is consistently collected for analysis. As technology develops and advances, individuals often find themselves yearning to expand their knowledge. Meteorological, operational forecast systems are designed to predict severe, short-term, and long-term weather patterns to keep communities up-to-date. Technological efforts inform the improvement of efficiency, effectiveness, and lead time for these model forecasts. More recently, certain offices have begun to pave the way within the National Oceanic and Atmospheric Administration (NOAA) for community open-source collaboration, acceleration of Research to Operations (R2O), and broadening of diverse stakeholder engagement across all sectors of the Weather Enterprise.

NOAA's Weather Program Office (WPO) specializes in supporting innovative research and helping transition R2O. Within the WPO, several programs exist, with one of them being the Earth Prediction Innovation Center (EPIC). EPIC consists of both contractors and federal employees who serve as the catalyst for community research and modeling advances in our Nation's operational forecast modeling systems. The Unified Forecast System (UFS) was introduced to enhance open-source scientific collaboration both within and outside of NOAA.

The UFS is a coupled, community-based Earth modeling system that encourages community collaboration to accelerate and incentivize the R2O journey. Simplifying the multitude of operational forecast models into similar categories, each with the same standard data core, will provide ease to users and reduce confusion. The UFS' open-source format allows users throughout the Weather Enterprise to access GitHub repositories. Users can utilize infrastructure, forecast model source code, and public research programming for their own personal projects before submitting for funding and grants. The foundation of the UFS is built upon user support and technological transparency, and is supported by the framework of the EPIC community.

The position of the UFS Student Ambassador was designed with the intention of creating a UFS Student Engagement Plan that highlights suggestions for closing the gap between the UFS and academia. The Student Ambassador will advocate for community engagement and technological advancements for academia while providing an undergraduate student

perspective on typical academic curriculums and experiences as a meteorology and atmospheric science student. They will perform innovative methods of outreach to academia to gauge interest and viewpoints on the UFS and evaluate UFS tutorials/training materials. Ultimately, both community engagement and technological components will result in recommendations listed in this UFS Student Engagement Plan and Short Range Weather Application and Land Data Assimilation (DA) Undergraduate Student Evaluation and Technical Report.

The goal of this summer research project is to collect qualitative data to share with the UFS and EPIC teams. The UFS is beginning to rise in popularity amongst diverse sectors of the scientific community, and it is hopeful that this project will support future research and innovative initiatives. This UFS Student Engagement Plan will explore numerous student perspectives regarding academic curriculums, programming, Numerical Weather Prediction (NWP), recommendations to increase and enhance stakeholder engagement, an exploration of university/student outreach performed, and an evaluation of tutorials for UFS applications.

Student Ambassador Personal Experiences and Perspectives

When embedding oneself in an academically challenging environment, it is easy to lose sight of what is important. The consistent competition and the feeling of inferiority overpower the primary reason that students attend university: to learn. Taking a step backwards and absorbing the importance of patience and flexibility is essential to creating a nurturing learning environment for oneself and peers.

Throughout the years of undergraduate education, numerous methods of conveying information from professors, fellow peers, and other mentors are incorporated into students' curriculums. As different educators have varying ways of delivering knowledge, it is interesting to assess how they demonstrate programming and NWP in courses in particular. When analyzing the general suggested academic plan for the meteorology and atmospheric science major, programming skills were scattered throughout courses offered from a student's first to last year in the program. Along with the general scientific courses including atmospheric thermodynamics, atmospheric dynamics, physical meteorology, and weather system prediction, programming is an essential skill embedded in their own courses and within other courses. This

section will discuss the student ambassador's experiences with programming throughout their undergraduate career thus far.

(1) Introductory meteorology course in MATLAB
(2) Advanced data analysis course in Python
(3) Cryosphere and climate course in Jupyter Notebooks
(4) Statistical experimental methods in R-studio

The introductory programming course in MATLAB introduced the Student Ambassador to new feelings of intimidation, fear, and apprehension regarding programming even with reassurance from professors. As the course progressed, students were a first-hand witness to their technological growth. The consistent motivation and support from professors, teaching assistants, and peers allowed students to look past doubts and dive deeper into programming. Every individual has a different learning style, and it is important to acknowledge these differences when training students for the intensity of the workforce.

The Python course was offered during the Student Ambassador's second year of undergraduate studies. They now had the opportunity to use their previous knowledge gained from their first programming course. This Python course differed from the previous MATLAB course, as the majority was "self-guided". This entailed students choosing their preferred data set for analysis through Python, consulting with external resources to solve any issues, collaborate in small groups, and to complete the majority of assignments within the designated class time period. The infrastructure of this course allowed students to brainstorm collaboratively throughout the semester, another beneficial skill that can be utilized for the workforce. Along with a collaborative structure, students learned general skills that can be utilized for data analysis.

A cryosphere and climate course allowed exposure to Jupyter Notebooks. Students enrolled in this class were given the opportunity to analyze data collected through automated weather stations scattered across the Antarctic and Greenland ice sheets. Variables such as melt rate, Surface Energy Balance (SEB), incoming shortwave radiation, albedo, etc. were plotted and then depicted on maps of Greenland or Antarctica for a visual representation.

Aside from these meteorological/climatological programming courses, the Student Ambassador was required to complete a statistics course about general experimental methods. The majority of assignments were completed in R-studio. Students found that the programming language R was useful and straightforward when it came to data analysis. Code was easily locatable online for any additional assistance.

Allowing students to thrive in their personal learning environments is essential to making them feel a sense of belonging at their universities and workplaces. Having mentors or professors who genuinely understand the student perspective allow students to thrive in the spotlight and embrace their unique talents.

Outreach and Community Engagement

The significance of community outreach and engagement lies within the assurance that each sector of the Weather Enterprise/scientific community is being reached and acknowledged. Industry and private sector represent the commercial distribution of weather data to other organizations, as well as utilizing government-provided data to keep the community informed. The government (e.g. NOAA), including laboratories and centers, support research and development, and are responsible for supplying variable data to research and commercial facilities. Lastly, the academic sector consists of many research opportunities, universities, and labs that contribute a large portion of innovative studies, methods, and ideologies to the scientific community. Accessibility for the industry/private sector, public sector, and academia alike is crucial for broadening the realm of diverse stakeholder engagement.

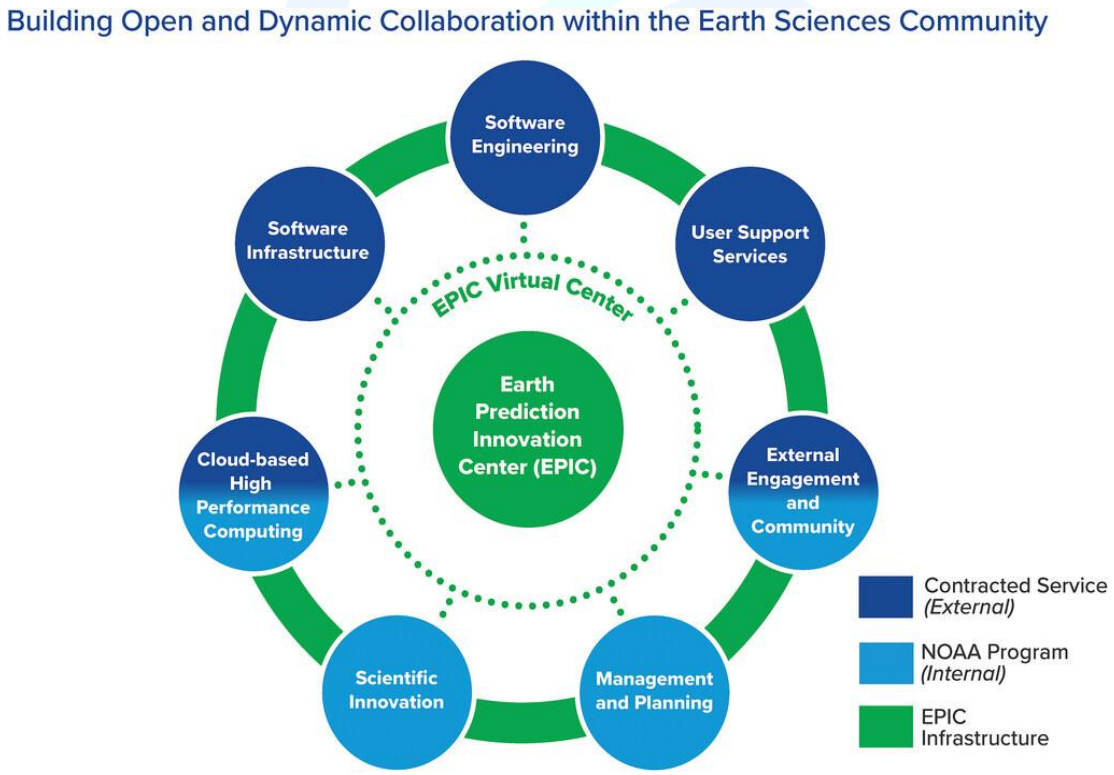
Project Inspiration

In last year's Unifying Innovations and Forecasting Capabilities Workshop (UIFCW) 2022 Report, a post-workshop survey displayed an imbalance of users in the UFS community. Academia had much less representation at UIFCW 2022 in comparison to the private sector or industry. In order to uphold stakeholder engagement and needs, the UFS and EPIC teams are working towards altering typical procedures to attract younger generations, fund greater amounts of low Readiness Level (RL) research, and prepare university-level students for the

workforce. This project ventures two prominent paths: (1) utilizing innovative outreach methods and community engagement strategies to increase stakeholder engagement, and (2) evaluate/measure the usability and accessibility of UFS application tutorials for a seamless potential integration into academic lesson plans/curriculums.

Construction and Collaboration

Before initiating any outreach methods, background details, guidance, and input were compiled and taken into consideration. This background information was obtained through many discussions with the EPIC Program and Contract teams along with leaders in the UFS community. The figure below is taken from Uccellini et al.'s BAMS paper (linked in References) and highlights EPIC's seven building blocks.



In order to preserve the prominent building blocks of EPIC (successfully providing user support, community outreach, supporting innovative research, adaptable software and

technological resources, and maintaining a diverse user database), the community engagement design will consist of six main components :

(1) Designing email drafts
(2) Creating list of universities and UIFCW 2023 student attendees
(3) Creating 'How the UFS Benefits Academia' document
(4) Creation of questions in Google Form format
(5) Sending emails to universities and UIFCW 2023 student attendees
(6) Receiving innovative feedback

University Outreach

After much deliberation regarding outreach, the decision was made to move forward with contacting universities through email. Each email sent to professors and department heads consisted of general information about the project, why this outreach was being performed, and how it would be mutually beneficial for their university program as well as the EPIC and UFS communities. Emails were sent to each university representative individually so as to maintain a personal connection. A document titled 'How the UFS Benefits Academia' was embedded within each email that contained general information about the UFS, how it can benefit students at any phase of their undergraduate or graduate career, and how it assists with preparing students for the workforce. This document was created so that university representatives would not be required to search extensively for a general overview of the UFS.

University selection proved to be a challenging task, as this project considered both atmospheric science and computer science students. Ensuring the involvement of both Minority Serving Institutions (MSI) and Historically Black Colleges and Universities (HBCU) expanded the field of accessibility to a wider range of underrepresented undergraduate and graduate students. While considering many factors, it is important to create the foundation for students in a variety of situations to be acknowledged and supported through this outreach. The fifteen schools selected consisted of renowned atmospheric science/meteorology programs, renowned computer science programs, MSI and HBCU universities, both undergraduate and graduate programs, and NOAA partnerships.

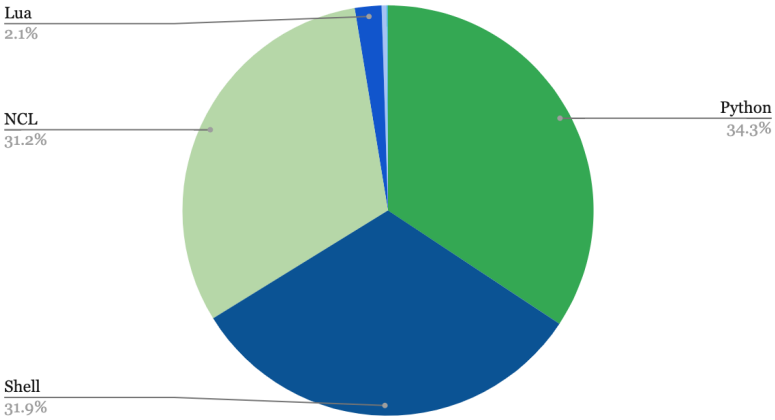
In order to receive responses and feedback from these university representatives, a list of questions was designed to inform the EPIC and UFS teams regarding adapting to the community's needs. The questions asked pertained to the availability and accessibility of programming/NWP courses to students, which software(s) their respective departments are using, if they have encountered any issues with their current software, and if they would be willing to participate in a "UFS Roadshow" (referring to in-person UFS demonstrations/online live tutorials). These questions, once compiled into a Google Form, were embedded into the email to spark an interactive experience. Innovative feedback began arriving shortly after the delivery of emails. Following the examination of responses, there appeared to be common themes noticed amongst the responses.

There is interest in live UFS demonstrations from the majority of respondents, as they stated their willingness to participate in innovative outreach activities. This will ultimately open many doorways toward greater stakeholder engagement. Throughout the responses, there were mentions of both undergraduate and graduate programs having programming/NWP courses available. However, some courses were only available in the form of optional electives or they were not offered every semester. It was also stated that the most common programming language used in a classroom setting is Python, as universities are beginning to stray away from foundational languages.

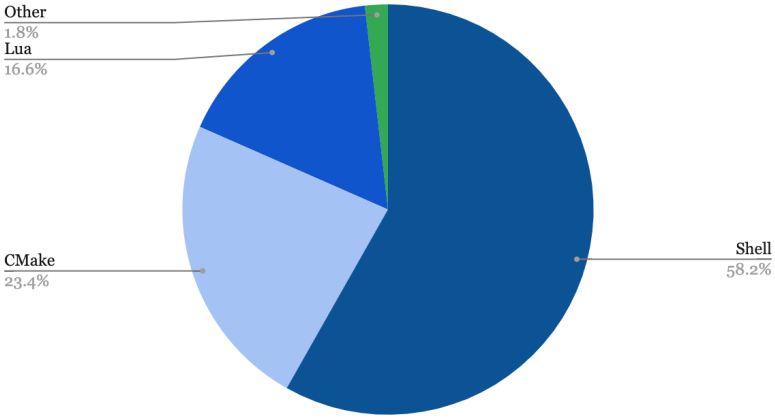
The spread of programming languages used in the UFS Weather Model, the Short Range Weather application, and Land DA were analyzed and charted below. This information was taken from the following sources:

- UFS Weather Model (https://github.com/ufs-community/ufs-weather-model/)
- Short Range Weather Application (https://github.com/ufs-community/ufs-srweather-app
- Land DA (https://github.com/ufs-community/land-DA_workflow)

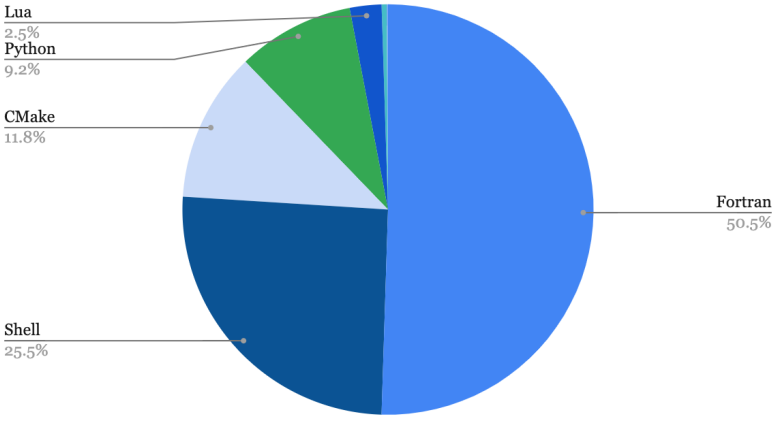
SRW Programming Languages Distribution



Land DA Programming Languages Distribution



UFS Weather Model Programming Languages Distribution



One response stood out from the others. A suggestion made towards the EPIC Program and Contract teams was submitted regarding the alignment of new, innovative research with ongoing software advancements. The anonymous response is included below:

"...EPIC can help by focusing more effort on bridging the gap between new science and software developments. Training students and scientists in academia how to apply the UFS for research is helpful for most users, but fails to reach the broad pool of potential developers. To provide one example, data assimilation is often viewed as an area of operational development in the US that lags other national centers. At the same time, there exists virtually no support for data assimilation developers to bring new methodology into JEDI. The DTC used to provide this level of support for GSI, I don't believe EPIC (or anyone else) has taken on this responsibility for JEDI." - Anonymous

Along with focusing on UFS integration into academia, it is recommended that this disconnect be mended so both endeavors can move forward together. It is critical that these two components of the modeling community align, as it will also benefit UFS community engagement efforts.

Student Outreach

The idea of student outreach came to light when the university outreach survey did not yield much information about UFS involvement as originally expected. University outreach and student outreach work hand-in-hand, as student outreach contributes insight into what students need, which can be compared side by side with what universities provide. This project provides outreach towards students with different levels of education who were already registered to attend UIFCW 2023. Now, outreach would be directed towards students who already have an interest in the UFS community.

After analyzing the list of students attending the workshop, the list went through the process of stratification where graduate and undergraduate students were identified. The drafted email consisted of information about the project itself, 'How the UFS Benefits Academia' document, and a Google Form composed of a different set of questions geared towards students. These questions included topics such as the student's involvement level with the UFS, if the student would like the UFS to become incorporated within their academic studies, if there

were any programs or resources that the student wished their university offered, and what the student hoped to accomplish by representing academia at this workshop. Some responses regarding what students wished their university offered are included below:

"It is currently difficult for students specifically in my department to attend conferences abroad even if it means partial funding." - Anonymous Student

"UFS training courses." - Anonymous Student

"An overview of the UFS, explaining its types and different applications and a brief training on how to run the UFS would be useful." - Anonymous Student

"More classes on scientific programming and architecture." - Anonymous Student

After reviewing student responses, it became more apparent that the UFS has not entirely made its way into the academic community. Aside from the UFS, most students acknowledged have little to no experience regarding NWP in general. A few mentions of the WRF (Weather Research and Forecasting) model were noted. There was interest in having the UFS become more deeply incorporated within academia through curricula and developed lesson plans. Along with this, requests were made for UFS training and introductory courses. Many students stated that they were attending UIFCW 2023 to learn about current research in the field, to network with individuals across the Weather Enterprise, to present research, and to familiarize themselves with different modeling frameworks.

When asking students, both undergraduate and graduate, about how they hoped to represent academia at this workshop, the responses were quite uplifting. Some anonymous responses are listed below.

"Increasing students' voice in what models are used in universities." - Anonymous Student

"Implement recent advances in academic research." - Anonymous Student

"I hope to learn new innovations that I could implement in my research and equally pass the knowledge and skills to fellow researchers in my institute." - Anonymous Student

"I hope to understand more on the applications of the UFS, the challenges associated with it and the ongoing research related to UFS S2S, SRW and MRW." - Anonymous Student

Student Ambassador Insights

After analyzing responses received through both the university outreach and student outreach portions, a list of insights and observations were made. These insights are listed below.

- Students of all educational backgrounds need to be heard and represented
- Students should be provided hands-on UFS learning experiences
- Allowing students to hear about other student experiences through UFS success stories and fellowship/internship information
    - Increases confidence when students are provided reassurance and motivation from others
- Creating a separate training series geared towards students (mentioned in more detail later)

Recommendations, Action Items, and Key Deliverables

Listed below are some recommendations geared towards improving and enhancing engagement and outreach as well as UFS training materials. These recommendations are meant to provide the UFS and EPIC teams with ideas and insights that can be taken into consideration to expand and improve community engagement.

Engagement and Outreach

- Prioritizing student outreach as they are the target demographic when expanding the UFS community
- Younger generations can be reached through social media platforms

- Establish a UFS Ambassador Program, with students and/or professors located at different universities, serving as a champion for UFS
- Collaboration with universities, AMS Education, NOAA education, COMET, and Met-ed to design and develop UFS lesson plans

UFS Training

- Curated training courses
    - Molded and designed to be used specifically by university-level students with little to no NWP experience
    - Platform agnostic training materials for UFS applications
    - Packaged scripts to set up modeling environment (so more time can be spent using the model code than setting up the environment it runs on)
    - Develop pre-requisite training materials and share with students prior to training session
    - Tutorials delivered in sequential format
- See Appendix A for additional recommendations

Engaging with Modelers and Developers

- Creating an inclusive environment for all platforms (Mac, PC, etc.)
- Creating more tutorials for different computing environments and SSH clients
- Distribution of pre-tutorial materials
    - Provides users with any experience level background knowledge on tutorial
    - User will know what to expect before watching the tutorial
- Implementing explanation of documentation commands directly into infrastructure and source code
- Short Range Weather (and future UFS applications) reconfiguration to Jupyter Notebooks
    - Appeals to younger programmers
    - Can be downloaded and accessed on a browser
    - Assists in the shift towards a highly utilized program language, Python

Summary

This UFS Student Engagement Plan was designed through collaboration efforts between the First Student Ambassador of the UFS, the UFS and EPIC teams, and Raytheon Technologies. The community engagement and technological components of this endeavor will ultimately assist the UFS community with academic integration. The insights gained from universities and students alike will be taken into consideration to implement improvements within the foundation of the UFS.

References

Alves, J., Tolman, H., Roland, A., Abdolali, A., Ardhuin, F., Mann, G., Chawla, A., Smith, J. (2023). NOAA's Great Lakes Wave Prediction System: A Successful Framework for Accelerating the Transition of Innovations to Operations. *Bull. Amer. Meteor. Soc.,* 104, E837-E850. https://doi.org/10.1175/BAMS-D-22-0094.1.

Jacobs, N. (2021). Open Innovation and the Case for Community Model Development. *Bull. Amer. Meteor. Soc.,* 102, E2002-E2011. https://doi.org/10.1175/BAMS-D-21-0030.1.

Land DA: https://github.com/ufs-community/land-DA_workflow

Short Range Weather: https://github.com/ufs-community/ufs-srweather-app

Stagnari, A., Dubots, L., Vogt, Jennifer., Huang, M. (2022). Working with Stakeholders: Engaging Sectors to Influence the Future of EPIC and UFS. Unifying Innovations in Forecasting Capabilities Workshop Report.

Uccellini, L., Spinrad, R., Koch, D., McLean, C., Lapenta, W. (2022). EPIC as a Catalyst for NOAA's Future Earth Prediction System. *Bull. Amer. Meteor. Soc.,* 103, E2246-E2264. https://doi.org/10.1175/BAMS-D-21-0061.1.

UFS Weather Model: https://github.com/ufs-community/ufs-weather-model

Appendix A

Short Range Weather Application and Land DA: Undergraduate Student Evaluation and Technical Report

Introduction

When adapting to generational change, observing the needs of beginner student programmers can build the foundation for innovation. One of the most important roles any mentor must take on is to visualize a career-development plan for their student. Providing personal experience and training materials are two of the most essential components to a successful learning experience/mentorship. Preparation for the future workforce is a common activity embarked upon during undergraduate studies. Now, more than ever, software engineers, data analysts, computer scientists, and programmers are in-demand for abundant technological advancements.

For both undergraduate and graduate students, sharing numerous frustrating technological experiences along with peers regarding programming is common. As a computer science requirement for meteorology and atmospheric science, students must complete introductory programming courses which could include MATLAB and/or Python. These undergraduate courses provide background information to apply technological knowledge to UFS tutorials. This evaluation of Raytheon tutorials and training materials is written from an undergraduate student perspective, including suggestions, minor critiques, personal experiences, and advice on adapting to beginner student programmers. The evaluated training materials will predominantly entail the Short Range Weather (SRW) application infrastructure code tutorial, the virtual Packer-SRW and Amazon Web Services (AWS) sandbox training, and the Land Data Assimilation (Land DA) CodeFest training.

SRW Application Infrastructure Code Training: Virtual Training Evaluation

The SRW application captures the capability to display atmospheric data with a lead time of minutes to days. Having hands-on experience to actually build the infrastructure required to run source code was surreal. After having an AWS account created, the ability to run any application at my disposal in the Cloud environment was unlocked.

Link for training:
https://epic-sandbox-srw.s3.amazonaws.com/Packer_SRW_2June2023.mov

While following the Packer-SRW pre-recorded tutorial, a list of notes and suggestions was compiled that will make this experience more personalized and beginner-friendly for undergraduate students. The general content of this tutorial consisted of creating and building the infrastructure to run the SRW application on the Cloud utilizing the framework packer. There were approximately twelve commands that were required to build the infrastructure.

Conciseness and efficiency of this training was encapsulated in an approximate time frame that successfully kept the viewer's attention throughout the duration of the recording. Raytheon's Chief Engineer led the discussion and did an impressive job tying together AWS Cloud services and SRW code. This tutorial created excitement and enthusiasm for students about building the background for a large, operational forecast system. AWS Sandbox basics and how it can be useful with accessing applications on personal electronic devices was showcased throughout the session. In the future, this virtual training has the ability to be used in an academic setting where students are taking programming and/or Numerical Weather Prediction (NWP) - integrated courses.

The constructive criticism portion of this evaluation consists of some general improvements to adjust to the needs of academia and beginner student programmers.

- The distinction of Mac versus PC servers is not made clear, as the tutorial is shown on a Mac.
- When the instance connection options are mentioned, the Mac terminal is associated with Mac users and puTTy is associated with PC users.
- When assessing the connection to an SSH client from an EC2 instance, it is not mentioned that the EC2 Instance Connect online browser is an option to run the infrastructure code.
- Providing an embedded or separate tutorial for puTTy configuration to steer the training's direction to be user-friendly for both Mac and PC
- Platform specific environments should have more detailed instructions.

There were twelve documentation commands included in the SRW pre-recorded tutorial. Including definitions and detailed explanations for these commands would be helpful to a user with limited programming knowledge. If additional details were provided with the code, it would increase the user's knowledge, understanding, and confidence in programming. This would also prove useful for students involved in research who are not as advanced or familiar with cloud services. When moving towards a more inclusive approach, detailed instructions and definitions would adhere to a user-friendly environment for the academic community. One of the commands used opens the file packer-srwcluster. When attempting to edit this file, it is necessary to mention clicking "i" to access insert mode. Without addressing this, the user becomes stuck in the file.

From a student perspective, coding is a generally intimidating skill to many undergraduate peers who have not been introduced to programming techniques and Cloud resources. Although the tutorial kept my attention and enthusiasm, many details needed to be explained further. A future goal for my summer project consists of SRW source code reconfiguration to Jupyter Notebooks to adapt to the academic community's accessibility levels. Not only does Jupyter Notebooks appeal to younger generations, it advocates for interactive computing, it emphasizes the ability to program and run code on a web browser, it resembles Python (commands and downloadable packages), and it allows for the user to organize and run notebooks (individual scripts) with titles and subheadings.

Aside from this tutorial, a separate virtual Packer-SRW and AWS sandbox training conducted by Raytheon's Chief Engineer. This virtual training also had attendance from experienced programmers located at other universities. Watching the aforementioned tutorial before attending this training allowed background knowledge to be used on AWS sandbox, educated questions to be asked, and the gradual familiarization with GitHub repositories. Keven did a fantastic job of answering questions with patience and composure. However, the live tutorial became fast-paced and rushed. It became difficult to keep pace with the speed of the demonstration, as it was not pre-recorded. Experienced programmers should update their training, as inexperienced students might work at a more deliberate pace than others if these presentations were to be incorporated into an academic curriculum.

CodeFest Land DA Training Evaluation

During the week of CodeFest, registration for the Land Data Assimilation (DA) training opened so that attendees could gain a better understanding of another UFS application aside from SRW. This tutorial had a larger attendance, attended by both mid-career professionals and university-level students. The goal of this training was to educate individuals with any level of programming experience about running Land DA infrastructure and source code through Cloud computing on any platform.

Some issues were encountered throughout this training, and these suggestions can be used to improve and inform future virtual or in-person CodeFest events. The speaker reiterated that the Land DA infrastructure and source code can be run on Mac and PC. An attempt to utilize the EC2 Instance Connect online browser resulted in malfunction. After inquiring about this with the demonstrator, they stated that the EC2 Instance Connect online browser was blocked for use with the Land DA application head node. By the time the set-up was reconfigured to a Mac terminal, the training had quickly progressed multiple steps beyond what could be accomplished. The speed at which this demonstration was administered was designed for mid-career, experienced professionals. The total time reserved for this training resembled approximately four hours, but was completed in less than a third of that time. Please see Appendix B for notes taken during these tutorials/training sessions.

To improve a user's experience during any training, the instructor must ensure that no attendee is left behind. Even if attendees have differing experience levels with the material, it is important that every attendee has a grasp on the topic before the end of the tutorial. To make a training successful, the tutorial needs to adapt to any and all experience levels.

Summary

Overall, the speakers chosen for these training sessions were extremely knowledgeable about the subject at hand. It is inspiring to see individuals so passionate and enthusiastic about future technology and sharing with others. In summary, there are numerous recommendations that are encouraged towards the progress and improvement of UFS tutorials and training materials. These action items are as follows:

(1) Create an inclusive environment for both Mac and PC users. Ensure that every attendee, no matter their platform of choice, is following along and is aware of the options available to them (EC2 Instance Connect not mentioned as an option).
    (a) Create embedded/separate tutorials, or packaged scripts for puTTy configuration, EC2 Instance Connect, and other SSH clients/methods of connecting to instance).
(2) Distribute a list of pre-tutorial materials to attendees before the scheduled training. This will inform them of what to expect and prepare for before the demonstration.
    (a) Potential pre-tutorial materials: puTTy configuration, creation of AWS account, items a user might encounter on AWS, general terms included in glossary, explanations/descriptions of each infrastructure construction documentation command, etc.
(3) Ensure that each detail is included in the tutorial. Excluding important information may create issues for the user.
(4) Pre-recorded and in-person tutorials might be more impactful than virtual tutorials. Tutorials that are fast-paced and have quick delivery can become frustrating for inexperienced users. This impacts the tutorials' usability in an academic setting.

When it comes to a diverse community of users, it is important to adhere to all sectors of the Weather Enterprise (industry, private sector, and academia) for successful outreach. The duties of the First Student Ambassador of the UFS are to ensure that undergraduate students receive training, that they are exposed to technological developments and advancements, and that the academic community is advocated for. These training materials and detailed tutorials offer a phenomenal opportunity for students to become more prepared for the executive workforce. It is the hope of many that students will become further involved with the UFS in their academic and professional experiences.

Appendix B: Notes from Training Sessions

Short Range Weather (SRW) Infrastructure Code Tutorial:

- Infrastructure program incorporates Packer framework
- Tutorial is built to run on the Cloud
- Using Amazon Linux Browser Instance
    - Step 8: vim srw - cluster.pkr.hcl
    - Opens a separate file within the browser instance
    - Issue: Amazon Linux Browser Instance will not allow editing in file
    - Issue Override: Enter “i” to be able to edit - this is not mentioned
- Transition from copying command to pasting the SSH into the code is confusing and rushed
- Platform specific environments should have more detailed instructions
- Providing definitions for documentation commands will assist students involved in research who are less experienced than AWS professionals
- Can use “man” command to provide definitions for each infrastructure construction command
- Tutorial kept attention span, but lacked many important details

Virtual Packer-SRW and AWS Sandbox Training:

- AWS sandbox allows user to perform any task outside the NOAA firewall
- Through packer, the user can download any program to run on any cloud service provider
- Makes point that user does not need puTTy to run this code
    - Need to mention this in the infrastructure building tutorial
    - Process is doable on the browser instance
- Jupyter Notebooks reconfiguration - Recommended that sagemaker is ideal for creating notebooks, but running code in Linux box is better
    - Cloud agnostic - supports three service providers
    - Sagemaker is only through AWS
    - Create notebooks in EC2 to create a Linux version

Land DA CodeFest Training:

- JEDI framework for data assimilation
- Source code is “containerized” and repeatable
- Can run on any system including the cloud, which is cost-effective
- Must change region and instance type for Land DA to run
    - Gives a module command error when running in the browser
- Once changed to head node instance - browser failed to connect to instance
    - Browser has been blocked from connecting to head node instance
    - Have to use SSH in terminal, puTTy, Mac, or others
- Switched to Mac terminal to continue running Land DA infrastructure, but demonstrator had already moved on
- Tutorial lasted less than a third of the reserved time - fast-paced and quick delivery meant for mid-career professionals

Appendix C: Glossary

AWS: Amazon Web Services, a cloud service provider.

AWS sandbox: Program testing environment, allows the ability to run code in isolation from other programs.

Cloud Computing: Allows users to work in virtual environments with no required maintenance.

Cloud Service: Any provider that offers Cloud computing services.

CodeFest: Collaboration or gathering of individuals to discuss or design an innovative computational topic.

EC2 Instance Connect: Using AWS Identity to run code instead of sharing a personal SSH key.

EPIC: Earth Prediction Innovation Center

HBCU: Historically Black Colleges and Universities.

Head Node: Connector between internal and external computing environment.

Instance: virtual machines which have the ability to run on the Cloud.

Jupyter Notebooks: Browser-based, interactive, collaborative programming language.

Land DA: Land Data Assimilation.

MATLAB: Programming language developed by MathWorks.

MSI: Minority Serving Institutions.

NOAA: National Oceanic and Atmospheric Administration.

NWP: Numerical Weather Prediction

Packer: Open-source tool. Allows users to run any program on any Cloud service provider outside of the NOAA firewall.

PuTTy: Open-source terminal capable of file transfer.

Python: High-functioning programming language. Capable of producing quality graphics and running any general programs.

Raytheon: Aerospace, intelligence, and defense organization.

RL: Readiness Level.

R-studio: Data analysis and programming tool.

R20: Research to Operations

SRW: Short Range Weather application, operational forecast model.

SSH: Secure Shell. Grants secure user access to programs.

UFS: Unified Forecast System. A coupled, community-based system.

UIFCW: Unifying Innovations in Forecasting Capabilities Workshop.

Weather Enterprise: Government, private sector, and academia.

WPO: Weather Program Office