Final Report of the

2024 NSF/DOE Workshop on NAIRR Software

May 1, 2025

Workshop Location:
Argonne National Laboratory
Lemont, IL

Workshop Dates:
December 3–4, 2024

Workshop Co-Chairs:
Dhabaleswar K. Panda, Ohio State University
Michael E. Papka, University of Illinois, Chicago and Argonne National Laboratory

NSF/DOE Contacts:
Sheikh Ghafoor, NSF/OAC
Saswata Hier-Majumder, DOE/ASCR
David Rabson, DOE/ASCR

This workshop was supported by the award OAC:2440948
from the National Science Foundation.

Suggested citation:

Final Report of the 2024 NSF/DOE Workshop on NAIRR Software, Michael E. Papka, Dhabaleswar K. Panda, Ilkay Altintas, Wahid Bhimji, Ewa Deelman, Murali Emani, Nicola Ferrier, Daniel S. Katz, Lois Curfman McInnes, Anita Nikolich, Feiyi Wang, and Jim Willenbring, US National Science Foundation Report, 2025.

Report Authors

Michael E. Papka	University of Illinois Chicago and Argonne National Laboratory
Dhabaleswar K. Panda	Ohio State University
Ilkay Altintas	University of California San Diego
Wahid Bhimji	Lawrence Berkeley National Laboratory
Ewa Deelman	University of Southern California
Murali Emani	Argonne National Laboratory
Nicola Ferrier	Northwestern University and Argonne National Laboratory
Daniel S. Katz	University of Illinois Urbana-Champaign
Lois Curfman McInnes	Argonne National Laboratory
Anita Nikolich	University of Illinois Urbana-Champaign
Feiyi Wang	Oak Ridge National Laboratory
Jim Willenbring	Sandia National Laboratories

Contents

Executive Summary

On behalf of the National Science Foundation's *Office of Advanced Cyberinfrastructure* (OAC) and the Department of Energy's *Advanced Scientific Research Computing* (ASCR) program, a National Artificial Intelligence Research Resource (NAIRR) software workshop was held at Argonne National Laboratory on December 3–4, 2024.

The workshop focused on identifying a feasible artificial intelligence (AI) software stack, or set of stacks, comprising computer programs, training and inference frameworks, libraries, user interfaces, data management, debuggers, and performance tools, to be made available to the broadest possible community. A key objective was determining the feasibility, essential components, and future research and development needed to sustain a long-term NAIRR effort. This effort, expected to launch in 1 to 1.5 years and target a 5-year horizon, will rely on robust software solutions that meet the evolving needs of the high-performance computing (HPC) and AI communities.

The workshop was organized by a technical steering committee with members from universities, national laboratories, and industry. More than 120 researchers, engineers, and educators from diverse organizations participated in the event. Over the course of two days, attendees took part in a keynote address, multiple invited talks, several breakout sessions, and a concluding panel discussion.

The key conclusions from the workshop are as follows:

1. The **NAIRR stack needs to leverage existing and emerging software solutions**, catering to a range of users (from novices to experts) across diverse hardware platforms and accelerators, including those used for education. Notably, the HPC community views these solutions as a *layered software stack* optimized for performance and scalability. In contrast, the AI community often refers to a *broader software ecosystem* that tightly integrates data, user support, and training frameworks. The NAIRR effort must bridge these perspectives to effectively serve all stakeholders.
2. The **NAIRR stack must respond to the evolving needs of the scientific and AI communities**, including real-time data analysis, privacy and security challenges, and portability across emerging AI hardware. Data management, which encompasses cleaning, curation, and annotation, must ensure that researchers can fully leverage the growing volume of diverse datasets.
3. The following components of the **NAIRR stack must remain flexible and extensible** to accommodate future technology advances: operating systems; middleware solutions for communication and resource management; languages and compiler support (with emphasis on Python, Julia, C, C++, and Fortran); workflow managers; and AI-related libraries, models, and frameworks, including HPC software that can be leveraged and/or enhanced through AI.
4. The **NAIRR stack must embrace open-source development** and ensure compatibility with new hardware to remain at the forefront of technological advancements.
5. The **NAIRR stack must deliver user-friendly interfaces** (e.g., Jupyter Notebooks, web-based platforms) to significantly lower barriers for newcomers to AI. Moreover, comprehensive training and robust ongoing user support must be seamlessly integrated into the software, highlighting the critical importance of accessible educational resources and dedicated guidance to empower new users in effectively navigating this technology.
6. The **NAIRR stack must address immediate user-support needs**; for instance, funding small supplements for current grantees during the NAIRR Pilot was proposed. Attendees also recommended creating intuitive *chatbot* interfaces to help users interact with the software stack, further reducing barriers to adoption and ensuring efficient troubleshooting and assistance.

1 Introduction

The National Artificial Intelligence Research Resource (NAIRR) Task Force identified the need to democratize access to artificial intelligence (AI) resources that have traditionally been limited to large organizations. The Task Force's 2023 report, ***Strengthening and Democratizing the U.S. Artificial Intelligence Innovation Ecosystem: An Implementation Plan for a National Artificial Intelligence Research Resource***, underscored the importance of creating a broad AI ecosystem that fosters innovation, enhances engagement for all, and ensures fair access to AI capabilities. The Task Force report emphasized the importance of participating in AI research. The NAIRR Pilot is a proof-of-concept for the eventual full-scale NAIRR. It will focus on supporting research and education efforts, including training students on the responsible use and development of AI technologies by providing access to infrastructure and training resources while gaining insights that will refine the design of a full NAIRR. The NAIRR pursues four key goals:

- spurring innovation,
- increasing the breadth of talent,
- improving capacity, and
- advancing trustworthy AI.

To achieve these goals, at the request of the National Science Foundation's *Office of Advanced Cyberinfrastructure* (OAC) and the Department of Energy's *Advanced Scientific Research Computing* (ASCR) program, we organized a NAIRR software workshop.

The workshop aimed to discuss an AI software stack, or a set of AI software stacks, which includes computer programs, training and inference frameworks, libraries, user interfaces, data management, curation, debuggers, and performance tools, and making them available to the broadest possible audience. The NAIRR software stack should strive to span scientific research domains, scales, and users, leveraging existing software stacks used in academia, national laboratories, and industry.

This workshop report presents the NAIRR Pilot's immediate needs and long-term goals, considers the composition of the NAIRR software stack over two to five years, and aims to address the evolving needs of the scientific community. These needs include real-time analysis of sensor/experimental data and decision-making using AI, privacy and security requirements in AI-based scientific applications, foundation models, mixed precision libraries, operating systems, programming environments, toolchains, storage needs for the ever-growing volume of training data, and portability of software across emerging novel AI hardware platforms.

The NAIRR Pilot's software stack will consist of existing software designed for a wide range of users, from beginners to experts, across multiple communities, including education, various hardware platforms, and both current and emerging accelerators. The report also aims to address immediate user support needs and suggests providing small supplements to current grantees during the NAIRR Pilot.

A key purpose of the workshop was to determine the feasibility and needs of the required software stack, along with the related research and development to sustain a long-term NAIRR effort, expected to begin in 1 to 1.5 years and aimed at a 5-year horizon. The workshop also explored procedures for integrating newly developed software into the NAIRR software stack and future funding requirements, specifically for NAIRR software. This inquiry differed from funding fundamental research in AI or AI applications in science, for which federal funding agencies already have funding opportunities.

The workshop report also emphasizes the creation of ethical, transparent (explainable), and trustworthy AI. The NAIRR's goal of utilizing AI for science differs from that of AI for industry, requiring a customized software stack. Moreover, the workshop included users who were not fluent in HPC or simulations, such as experimentalists managing a deluge of data. Lastly, the report aims to be a cohesive outcome of the workshop that ensures the seamless integration of inputs from all subgroups involved.

The NAIRR aims to spur innovation, increase breadth of talent, enhance capacity, and promote trustworthy AI. The NAIRR software workshop aligns with this vision and laid the foundation for a robust, democratized AI research infrastructure that empowers a broad user base and drives innovation throughout the U.S. AI ecosystem. Furthermore, it addressed the evolving needs of the research community, such as real-time data analysis, privacy and security in AI applications, and software portability across emerging AI hardware platforms. By focusing on the feasibility and needs of a longer-term NAIRR, this workshop explored procedures for incorporating new software developments and future funding requirements, ensuring that AI tools and frameworks remain cutting-edge, user-friendly, and adaptable to various research environments.

Specifically, the workshop examined the various components that should comprise the NAIRR software stack. The key elements under consideration included operating systems like Unix and Linux, which form the backbone of many AI applications. The discussion also covered middleware solutions for communication and resource management. Language support and compilers were another key focus, with special attention given to widely used languages such as Python, Julia, C, C++, and Fortran, which are essential for developing AI models and performing computational tasks. Additionally, the workshop investigated workflow managers and AI-related libraries, including machine learning (ML) and deep learning (DL) libraries, models, and frameworks, which are essential for developing and deploying AI applications. Finally, the NAIRR software stack must be compatible with researchers' existing software dependencies, including HPC libraries and tools.

The goal was to identify various software options that cater to users' diverse needs without endorsing a single supplier. Emphasis was placed on open-source options to ensure broad accessibility and adaptability. Further discussions addressed immediate user support needs for the NAIRR Pilot software stack, established priorities for the comprehensive NAIRR software stack, and identified future investment requirements. These discussions focused on open-source development and support for emerging hardware, ensuring that the NAIRR software stack remains cutting-edge and relevant to evolving technological advancements. The workshop organizers conducted a pre-workshop user needs survey to better understand the software and libraries required for AI research. Based on the pre-workshop survey, a second, refined survey was prepared for the *NAIRR Pilot meeting* (February 19–21, 2025), which is reported here, as it surveyed a wider audience with some overlap.

2 Case Studies

At the outset of the workshop, six domain-focused case studies were presented to illustrate how AI can accelerate discovery, facilitate novel analyses, and open new avenues for scientific exploration. These studies spanned x-ray science for real-time materials characterization, advanced biology for protein design, and the integration of AI in science education. They also addressed AI-driven methods for cosmology, regulatory challenges in medical imaging, and the application of machine learning architectures in weather forecasting. These examples demonstrated both the promise of AI workflows in transforming scientific research and the persistent gaps that must be bridged to realize that promise at scale. The following section details each case study, examining the opportunities for innovation and the specific challenges—ranging from data collection and foundational model development to infrastructure requirements and formal uncertainty quantification—that emerged in these various domains.

End-to-end **x-ray science** powered by HPC and AI will unlock new scientific capabilities from existing instruments used in materials characterization. For example, at the Advanced Photon Source, a large-scale experimental user facility, AI at the edge enables the real-time analysis of Gb/s data streams, producing results that are often more accurate and 100 times faster. It also facilitates self-driving experiments and instruments to maximize information gain in minimal time and learns material physics directly from measurements, thereby expanding the knowledge base. **Gaps:** There are challenges associated with AI-aided real-time data analysis, foundation models, and the curation of data and models. A key challenge is striking the right balance between model accuracy, physics-based insights, and the practical constraints of real-world hardware. This includes dealing with complex-valued data and developing computational methods that efficiently incorporate scientific principles. As we work towards a comprehensive scientific AI assistant for experiment planning, guidance, and operation, we face gaps such as the need for multimodal, scientific context-aware foundation models, standard interfaces for tool usage, agentic AI capabilities, seamless machine learning operations (MLOps), as well as seamless computational and experimental provenance tracking and meta-analysis for the evaluation and deployment of foundation models.

In **biology**, programmable protein design entails a framework that allows users to prescribe programmable design constraints via a natural language interface, providing ease and flexibility. A critical challenge in realizing such a framework is a lack of comprehensive multimodal protein design datasets that integrate text, protein/gene sequence, and structure/conformational modalities to build aligned representations for protein sequence-function mapping. Curating such a dataset requires LLM-assisted workflows to create rich narratives that can resolve potential issues, such as mode collapse. Additionally, we lack workflows effectively designed to integrate experimental observables with foundation models seamlessly. Moreover, many of these observations are qualitative and not always quantitative, and there is a lack of sufficient experimentally labeled datasets. **Gaps:** To train such multimodal foundation models, we require high-performing software stacks that are easy to customize. For instance, existing software stacks for text-vision models are not easily transferable to protein multimodal models. The current stacks require significant development time to incorporate custom changes. There is also a need for libraries that have ease of use while retaining scalability and performance. Finally, in the broader context of automated scientific discovery, there are software gaps in automated test beds that link foundation model outputs with self-driving laboratories. We require agentic frameworks to evaluate HPC resources and select the top-performing candidates for self-driving laboratory experiments. The framework must be customizable to implement agents with the sophistication to plan and execute fine-grained steps of robotic pipelines, thereby realizing self-driving experiments and recording experimental outputs to provide feedback for the foundation models and to scientists in an easily understandable manner.

As a data-rich science, **cosmology** is an excellent application domain for AI/ML methods. A convergence of data-intensive and high-performance computing pathways will accelerate adoption. AI methods have solved and will solve problems that could not be approached otherwise. These methods can and must be applied in many places due to the size and complexity of the data sets. However, the status of formal uncertainty quantification (UQ) applications in these areas remains relatively crude, primarily due to the complexity of the problem. The presence of bias due to several problems with measurements, modeling uncertainties, and assumptions remains a key issue, as techniques such as discrepancy modeling are not yet mature enough. A combination of both physical and modeling input into purely data-based methods is needed, as is widely recognized, but current approaches only represent a starting point. Large-scale models, combined with massive computing resources, can open up new avenues for exploration. **Gaps:** Historically, there was a gap between HPC and AI hardware; now, they are essentially the same, the consequences of which are still unknown. AI applications, primarily LLMs, are driving hardware evolution away from double precision, and modeling and simulation software tools must address the challenge of handling mixed precision. This presents an argument for having a somewhat unified toolchain for AI, modeling, and simulation. The diversity of the AI for Science (AI4S) application space is daunting; it is significantly more complex than the space with which HPC professionals are familiar. Nevertheless, the AI application ecosystem has a robust framework for addressing this, including deep learning (DL) frameworks, machine learning (ML) libraries, natural language processing (NLP) tools, notebooks, APIs, and other related technologies. However, the optimal strategies for combining HPC and AI approaches, such as modular design and workflow management systems, remain uncertain. A hybrid approach to integrating HPC and AI software stacks is probably best determined by pilot projects rather than a top-down approach. HPC facilities will need to become more cloud-like to support the AI toolchain, including containerization, orchestration, elastic scheduling, and support for hybrid workflows.

Weather forecasting is an excellent testbed for newly developed machine learning architectures, as the extensive observational data necessary to make accurate predictions pushes the limits of current hardware and software. ERA5, the fifth generation of climate reanalysis produced by the European Centre for Medium-Range Weather Forecasts (ECMWF), offers hourly data on various atmospheric, land-surface, and ocean-state parameters, along with uncertainty estimates. Tasks such as predicting the three-dimensional (3D) atmosphere for medium-range weather forecasting (up to 14-day lead time), creating emulators for climate research, and downscaling images for local-scale impacts of weather and climate rely on datasets of hundreds of terabytes to petabytes of data. The advent of scalable machine learning architectures (e.g., transformers), the availability of high-quality data, and access to numerous GPUs and TPUs are driving a paradigm shift in weather forecasting. Current software is primarily designed for traditional vision-based tasks, encompassing everything from data loading to readily available architectures. This limitation affects academic researchers and helps explain why most of these models are created by large technology companies. **Gaps:** Due to the large image sizes (721 x 1440) and the number of channels (potentially hundreds to thousands), substantial GPU memory is necessary for the activations alone. I/O is typically the limiting factor for training, so a significant challenge is adapting the current generation of hardware and software to these datasets. Custom I/O for training with node-local storage and improved caching/prefetching compared to native PyTorch Lightning has enhanced training time by 20-30 percent (DALI and DLIO for benchmarking and profiling); however, future generations will need petabytes of training data, and the model architecture along with deep learning packages are not optimized for models with O(100) channels. Nontrivial model parallelism techniques—such as gradient checkpointing, parameter sharding, and tensor parallelism—are essential for efficiently managing memory usage and enabling the training of large models. Moreover, PyTorch requires considerable customizations to minimize pre-processing and any nontrivial CPU-based pipelines, such as asynchronous operations.

Within **regulatory science**, research across numerous program areas treats AI and machine learning as major focal points for developing lifesaving medical devices. Tools are designed to help innovators assess the safety and effectiveness of emerging technologies at every stage of device development. Over the past decade, advancements in AI image processing have led to a marked increase in clearances and approvals across diverse product areas. However, the availability of and access to medical imaging data remain a persistent challenge. To address this, a recent collaborative initiative was launched to establish a medical imaging data exchange platform equipped with built-in tools for creating algorithms that meet regulatory standards, ultimately benefiting the broader imaging ecosystem. **Gaps:** Current generative AI research centers on formulating a case-agnostic approach to assessing factual accuracy, employing performance assessment strategies such as benchmarking, expert evaluation, and model-based evaluation. Key hurdles in medical AI, from a regulatory standpoint, include the limited availability of accessible, sustainable data platforms that meet stringent requirements, the need for advanced evaluation platforms incorporating new methodologies and performance metrics, and the complexities involved in determining the quality of synthetic data.

Recognizing the expanding **role of AI in education** across diverse disciplines, a large environmental science course integrated AI-driven tools to enhance programming support and tackle complex analytical problems. This integration opened new opportunities for students to explore large-scale geospatial and observational data, apply advanced machine learning techniques to ecosystem modeling, and conduct real-time analyses of climate variables. Although these capabilities significantly extended the scientific scope of the course, they also surfaced multiple areas in need of attention. **Gaps:** The course implementation revealed a requirement for specialized hardware and local storage to run large-scale AI applications, as well as the limitations of the initial configuration in handling complex tasks. A persistent challenge lay in developing an adaptable, scalable infrastructure capable of accommodating evolving AI tools. Another gap was the absence of streamlined mechanisms for rapid deployment and testing across diverse AI models. Furthermore, linking Jupyter-AI with the course's tools necessitated custom configurations for an online environment, highlighting the complexity of such setups. Ultimately, this experience underscored that truly transformative AI in educational settings requires robust technical foundations, seamless model integrations, and carefully preconfigured AI environments to enable broad-scale access and flexibility.

3 State of the Community

3.1 Current Software, Tools, and Gaps

A wealth of HPC and AI resources is available across training and inference, data management, models and datasets, accessibility and usability, and security and privacy. Yet, **integration** across these focus areas remains a persistent challenge. Researchers continue to seek end-to-end workflows—from initial data ingestion and curation to final model deployment—that are robust, reproducible, and secure. This section explores the community's growing **demand** for better incentives, privacy-aware frameworks, seamless hybrid HPC–Cloud–Edge infrastructures, and comprehensive education initiatives that lower the barrier to entry for domain experts. Finally, we discuss the critical role of a national-scale resource like the **NAIRR** in complementing, rather than duplicating, existing industry-driven and HPC solutions while prioritizing the **unique needs of science**.

3.1.1 Training & Inference (Tables: 1 & 2)

Training and inference remain central pillars of AI workflows. The research community, encompassing academia, government laboratories, and industry, predominantly relies on well-known deep learning frameworks such as **PyTorch** and **TensorFlow**. For large-scale model training, frameworks such as **DeepSpeed** and **Megatron-LM** enable efficient distributed training, particularly for large language models (LLMs). In some cases, existing LLMs need to be fine-tuned for a specific scientific purpose. Meanwhile, tools like **vLLM** and **TensorRT-LLM** optimize inference performance for production deployments.

Beyond the core frameworks, there is an increasing need for containerization (e.g., via **Docker** or **Singularity/Apptainer**) and for reproducible environments, especially in the HPC ecosystem. Researchers also see value in **JAX** for high-performance machine learning and differentiable programming, while **Hugging Face** Spaces and model repositories facilitate experimentation and community sharing.

Despite the rich ecosystem, multiple **gaps** remain. Users require streamlined pipeline management and better environment version control to avoid the pitfalls of inconsistent or brittle deployments. As large and multimodal datasets become more prevalent, resource reservation and job scheduling complexities (e.g., across distributed systems) can become bottlenecks. There are also mundane challenges that need to be addressed, such as submitting remote workloads to remote systems. Issues such as a lack of standard interfaces and the need for multi-factor authentication hinder the automation process. Another critical gap emerges in bridging HPC and emerging AI stacks: how to efficiently port PyTorch or other frameworks to new hardware, such as Cerebras, Graphcore, and AMD, while retaining reproducibility and performance.

Several participants emphasized the importance of privacy-preserving and secure infrastructure for training on proprietary or sensitive data, as well as the imperative to integrate *non-deep-learning* AI methods, domain-specific simulations, and trustworthy AI frameworks into mainstream training and inference pipelines. Finally, reproducibility—particularly the ability to checkpoint, convert, and retrain models—remains a significant challenge at scale.

Tool/Software	Description/Use Case
PyTorch & TensorFlow	Core deep learning frameworks for training & inference across multiple domains.
DeepSpeed & Megatron-LM	Distributed training of large-scale models (LLMs).
JAX	Differentiable programming, high-performance ML, often used in research.
Hugging Face Repos/Spaces	Hosting & sharing of pre-trained models; quick demos for domain use cases.
vLLM	Optimized LLM inference with low latency.
TensorRT-LLM	NVIDIA's high-performance inference engine for LLMs.
Containers (Docker, Singularity)	Portable, reproducible environments for AI workflows.
Scikit-Learn	Classic machine learning, widely used for simpler training & inference tasks.
QisKit, PennyLane, TorchQuantum	Quantum/ML frameworks (used in specialized research).
Llama, llama.cpp, ollama	Foundation model & inference frameworks for LLM experimentation.
Lightning AI	Wrappers for PyTorch/TF enabling easier training & scaling.

Table 1: Commonly used tools and software for training and inference.

Gap	Description/Need
Pipeline Management & Monitoring	End-to-end training/inference pipelines are complex; need integrated workflow & provenance tools.
Resource Reservation & Usage	Allocating and scaling HPC/cloud resources as well as workload submission and monitoring remain challenging for workflow systems and end-users.
Software Version & Environment Control	Tools like Spack/E4S exist, but consistent environment management is still burdensome.
Conversion of Training Checkpoints	Converting checkpoints for cross-framework inference or edge deployment is error-prone.
AI on Private/Proprietary Data	Confidentiality concerns require robust privacy-preserving training & inference methods.
Performance Portability & Scaling	Users want their code to seamlessly run on GPUs, specialized HW (Cerebras, Graphcore), HPC clusters, etc.
Incentives & Reproducibility	Lack of tools and clear incentives to share reproducible pipelines; reproducibility remains a cultural & technical gap.

Table 2: Top gaps in training and inference workflows.

3.1.2 Data Management & Storage (Tables: 3 & 4)

Effective data management and storage solutions underpin the success of AI projects, yet much of the existing storage stack (e.g., parallel filesystems such as **Lustre**, **DAOS**, **Spectrum Scale**) was initially engineered for HPC workloads with large sequential I/O. AI training often involves many small random reads, dynamic data augmentation, and the need to quickly load large batches of unstructured data (e.g., images, text, videos).

Consequently, frameworks like **Globus** for data transfer and **MinIO** for object-based storage are increasingly important, complemented by standard scientific formats such as **HDF5**. Multiple workshop participants emphasized the importance of **FAIR principles** for data (Findable, Accessible, Interoperable, Reusable) as well as **FAIR4ML** considerations. Tools that integrate metadata (including domain ontologies) are necessary to handle the diversity and complexity of emerging AI datasets.

Key **gaps** include the lack of user-friendly data lifecycle management (especially for large, ever-growing datasets), the difficulty of integrating domain-specific metadata and provenance, and bridging HPC systems' node-local storage with shared parallel filesystems in a way that is easy for non-expert users. The community is also requesting a "data commons" approach, which enables the sharing of curated, domain-relevant datasets, provided that privacy and licensing constraints are respected.

Tool/Software	Description/Use Case
HDF5 & NetCDF	Common scientific data formats for array-based data.
Globus	Data access, sharing, & transfers across different sites.
MinIO	Distributed object storage is frequently used for AI data.
Parallel Filesystems (Lustre, DAOS, Spectrum Scale)	HPC-grade storage often used for large-scale training.
Data Version Control (DVC)	Versioning of data & experiment tracking.
Metadata & Ontologies (HPC-FAIR)	Tools/ontologies for describing data, e.g., HPC-FAIR, domain-specific ontologies.
Tools for AI Model Management	Tools such as MLFlow that allow model and experiment tracking

Table 3: Common tools for data management and storage in AI.

Gap	Description/Need
AI-ready Metadata Layer	Tools to systematically capture domain-specific metadata, provenance, and semantics.
Unified Ontologies & Standards	Lack of consistent data schemas across scientific domains hinders reusability.
Multi-tiered Storage Integration	Need seamless bridging of node-local and shared filesystems (burst buffers, HPC, cloud).
Lifecycle & Provenance Management	End-to-end policies for data creation, curation, archival, and potential unlearning/removal.
Data Privacy & Confidentiality	Mechanisms for secure storage & controlled access, especially in regulated domains.
Scalability of Data Movement	Transferring multi-terabyte datasets from distributed locations is expensive & time-consuming.
Data Discovery Services	Capabilities to discover datasets relevant to a particular scientific discovery.

Table 4: Top gaps in data management and storage.

3.1.3 Current Models, Datasets, and Gaps (Tables: 5 & 6)

The ecosystem of AI **models** and **datasets** is increasingly diverse. Public model repositories (e.g., **Hugging Face** and **OpenMined**, as well as domain-specific repositories) have enabled the proliferation of pre-trained models, particularly large language models (LLMs) and foundational vision models. Domain researchers in agriculture, climate science, health/medicine, and materials science have begun adopting these models, often requiring specialized datasets such as **PlantVillage**, **Phenobench**, or curated medical datasets subject to Health Insurance Portability and Accountability Act (HIPAA) compliance.

Crucial **gaps** include the need for better data curation (e.g., removing or correcting "bad" data), robust annotation tools, and the ability to **update** or **untrain** models without retraining from scratch. Synthetic data generation is gaining traction as a means to address proprietary or sparse datasets, but best practices for verifying data authenticity and fidelity are still in development. Researchers desire more explicit **FAIR4ML schemas** to describe models and data, enabling more consistent and transparent sharing across different platforms.

Model/Dataset/Software	Description/Use Case
Hugging Face Repositories	Hosting & serving pre-trained models (transformers, LLMs, etc.).
PlantVillage, Weed Detection, CropAndWeed, Fruits-360	Public agriculture datasets used for plant disease/weed detection tasks.
Corn/Soybean Disease, Growth Stages (private)	Proprietary agriculture datasets used in industry or specialized research.
All of Us (NIH)	Confidential medical dataset, subject to stringent privacy & IRB rules.
Kaggle, Data.gov, NASA/CDF	Wide variety of open datasets for AI & data science competitions.
Domain-Specific Repositories	E.g., `Phenobench` in crop research, `Flatiron` multimodal cosmology data.

Table 5: Common models and datasets in use across various domains.

Gap	Description/Need
FAIR Model Schemas (FAIR4ML)	Standardized ways to describe, discover, and reuse models & their training data.
Model Unlearning & Continual Training	Efficient removal of problematic data or incremental updates without full retraining.
Synthetic Data Generation	Tools for generating domain-specific synthetic data while preserving statistical fidelity.
Benchmarking & Rigorous Evaluation	Need standardized metrics for comparing models across tasks and domains.
Data Mobility & Federated Learning	"Moving compute to the data" to address connectivity or privacy constraints.
Multi-modal Integration	Merging images, text, sensor data, and simulation outputs into cohesive models.

Table 6: Top gaps in models and datasets for AI.

3.1.4 Current Accessibility & Usability and Gaps (Tables: 7 & 8)

A recurring theme is that accessibility in AI is about more than raw computing power. Researchers and practitioners seek frictionless environments—commonly, **Jupyter Notebooks** and web-based platforms—that minimize overhead for domain scientists who may not be HPC experts. Tools such as **Open OnDemand**, **Globus Compute**, or discipline-specific **Science Gateways** have made strides. Yet, the complexity of large-scale AI remains a barrier to new entrants.

Workshop participants also cited the importance of **facilitators** (akin to Campus Champions, XSEDE's ECSS or "AI research facilitators") who provide hands-on support. Standardizing data collection and annotation practices would help ensure that domain experts can more easily **contribute** as well as **consume** AI resources. The "digital divide" extends to AI in many domains, with limited connectivity in rural or resource-poor regions, making data ingestion and model inference difficult.

Tool/Software	**Description/Use Case**
JupyterHub, Jupyter Notebooks	Interactive computing environment for prototyping, teaching, & collaboration.
Open OnDemand, Globus Compute	Simplified web portals for HPC/AI resource access & job management.
Science Gateways	Domain-focused portals (GUI-based) for specialized AI/ML tasks.
TAPIS, DIAMOND, TACC interfaces	HPC/AI abstraction layers and workflow engines.

Table 7: Common accessibility and usability tools for AI.

Gap	**Description/Need**
Ease of Environment Setup	Containerization & environment mismatch hamper broad adoption; simpler solutions needed.
Onboarding & Education	Many new AI users lack HPC experience; guided learning paths or "AI facilitators" would help.
Data Movement & Management	Users struggle with multi-step workflows to ingest large data for training or analysis.
Computational Resource Heterogeneity	Each cluster has different scheduling, container, or library constraints.
Low-code/No-code Interfaces	Domain experts want GUIs or chat-like interfaces for model exploration without heavy coding.
Community Practices	Users want to be able to follow community practices for data and computation management.
Digital Divide & Connectivity	Rural or under-resourced communities have limited network bandwidth for data transfer.

Table 8: Top gaps in accessibility and usability for AI workflows.

3.1.5 Security & Privacy (Tables: 9 & 10)

Security and privacy considerations become increasingly important as AI permeates sensitive domains, including healthcare, finance, agriculture, and those involving proprietary genetic data. A variety of **Privacy Enhancing Technologies (PETs)** exist, including **Homomorphic Encryption**, **Differential Privacy**, **Secure Multi-Party Computation**, and **Federated Learning**; however, many remain complex to implement at scale.

Workshop participants also pointed to the growing need for adversarial robustness and **red-teaming** tools (e.g., IBM's Adversarial Robustness Toolbox and Microsoft's Counterfit). However, **practical** best practices around model security (model theft, data leakage, etc.) are still lacking. End-to-end security, from data ingestion to model deployment, rarely has a single blueprint, especially in interdisciplinary, multi-institutional collaborations. Education, reproducibility, and the notion of *trustworthy AI* are further cross-cutting challenges.

Tool/Software	**Description/Use Case**
Privacy Enhancing Crypto (PECs)	Homomorphic encryption (HE/FHE/PHE), ZKPs, secure MPC, differential privacy.
Adversarial AI Evaluation (ART, Counterfit)	Toolkits to test AI models against adversarial attacks.
Trusted Execution Environments (TEEs)	Hardware-based enclaves (Intel SGX, AMD SEV) for secure computation.
Federated Learning Frameworks	Often built atop PyTorch or TensorFlow to train models without centralizing data.
Confidential Computing (NVIDIA, Intel TDX)	Industry solutions to protect data and models in hardware-based secure enclaves.

Table 9: Common security and privacy tools for AI.

Gap	**Description/Need**
Practical PET Integration	Many privacy-enhancing cryptographic methods remain difficult to deploy & scale.
Standardized Security/Privacy Blueprints	Researchers lack reference architectures for end-to-end secure AI.
Data Governance & Ownership	Clear policies for who owns the data/models, especially in multi-institution consortia.
Adversarial Robustness Testing	Tools exist but remain underused; best practices for systematic red-teaming are lacking.
Federated Identity & Access Management	Need robust solutions for cross-institution authentication (e.g., InCommon, NIH login).
Regulatory & Ethical Gaps	AI regulations, fairness, bias, and explainability remain unaddressed in many domains.
Model Unlearning & Data Removal	Mechanisms to remove or anonymize data points after model training are still in their infancy.

Table 10: Top gaps in security and privacy for AI.

3.1.6 Summary

Across all five focus areas—training/inference, data management, models/datasets, accessibility/usability, and security/privacy—the community has developed a substantial collection of tools. However, **integration** remains a recurring challenge: researchers consistently request end-to-end workflows, from data ingestion and curation to final model deployment, that remain robust, reproducible, and secure. Integration is also essential for smoothly incorporating the NAIRR stack into researchers' existing software ecosystems, increasing the likelihood that the NAIRR software will be widely adopted and used.

There is a growing **demand** for:

- **Better incentives and tools** to share, document, and maintain reproducible solutions.

- **Robust frameworks** for privacy-aware and secure AI, including unlearning and continuous model updates.
- **Hybrid HPC–Cloud–Edge workflows with underlying robust workflow management systems** that meet domain-specific needs (e.g., agriculture, medical imaging).
- **Comprehensive education & facilitation** that lowers barriers to entry for domain experts.

Finally, participants frequently highlighted that the **NAIRR** (or any national-scale resource) should capitalize on existing industry-driven software and HPC solutions while focusing on the ***unique needs of science***, such as long-term data curation, specialized domain workflows, or novel compute architectures for emerging AI paradigms.

3.2 Workshop Survey

A pre-workshop survey was distributed to gain informal insight into participants' perspectives on the software workshop, identify key areas of interest, and guide meaningful discussions. Rather than being a formal survey, its primary purpose was to gauge the community's current state and help shape both this workshop and future follow-on workshops. Feedback from the survey also informed the design of break-out sessions. The survey broadly covered topics such as the AI software ecosystem, applications, deployment strategies, data challenges, infrastructure, and associated hurdles. It included a mix of multiple-choice, Likert scale, and open-ended questions to collect quantitative and qualitative insights. Graphs and a detailed summary of these initial survey results are provided in Appendix C. Building on insights from this preliminary feedback, a refined version of the survey was distributed at the NAIRR Pilot's Annual Meeting in February 2025, receiving three times as many responses. Those expanded results are presented below. The refined survey focused on gathering information about attendees' research domains involving AI, specific AI applications used, challenges encountered, expertise levels, computational resource usage, and motivations driving their AI research. Analysis of these responses highlights several key insights that characterize the current landscape of AI research within the NAIRR community.

The NAIRR Pilot's Annual Meeting responses demonstrated a broad use base, with computer science representing the dominant discipline (Figure 1). Engineering was the second most represented field, followed by biological and physical sciences. This distribution highlights the interdisciplinary nature of AI research, though with a strong technical foundation.

In terms of expertise levels (Figure 2), the majority of respondents identified as experts in AI, with a significant number also reporting advanced competency. This suggests that the respondents were experienced AI practitioners. However, some were also intermediate, beginner, or basic-level users, so an accurate mapping of potential NAIRR users is not captured in this context.

Attendees reported various motivations for their AI research activities (Figure 3), with research advancement being the primary driver. Innovation ranked second, followed by operational efficiency and cost reduction. Notably, healthcare improvement and commercialization received minimal attention, suggesting that the NAIRR community is more focused on fundamental research than on commercial applications.

Model development is the predominant AI application among attendees (Figure 4), followed by model optimization and inference tasks. Data curation and evaluation metrics also received substantial attention, while generative AI, education, data analysis, and model analysis were less frequently reported. This distribution indicates a research community primarily focused on foundational AI development rather than applied use cases.

PyTorch emerged as the dominant framework for AI development (Figure 5), used by nearly twice as many

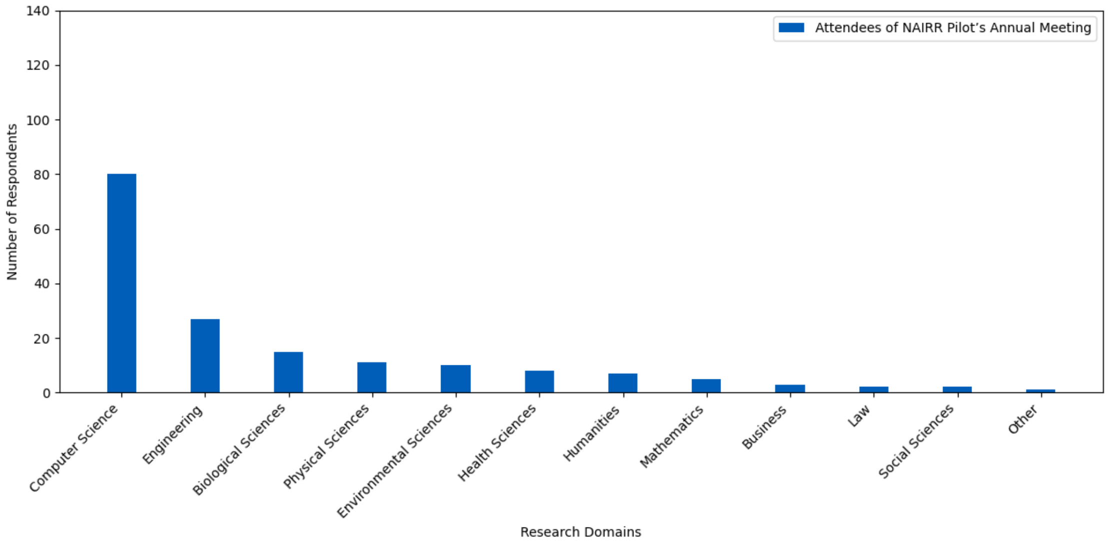

Figure 1: Research domains in which attendees were using AI

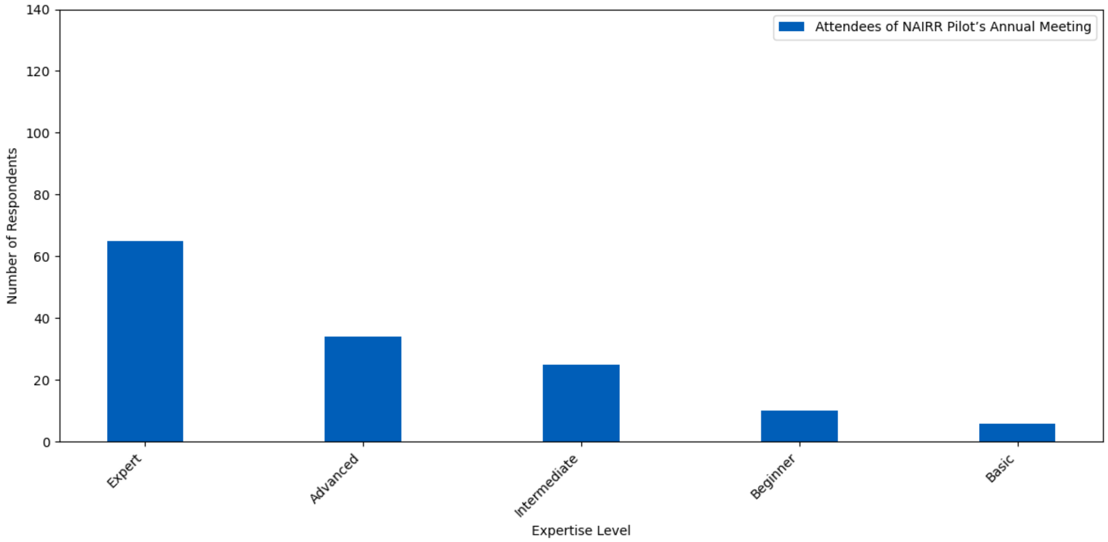

Figure 2: Self-reported AI expertise levels of attendees

respondents as the second-most popular framework, Hugging Face. Scikit-learn, TensorFlow, and various APIs also had substantial adoption, while specialized tools had more limited usage.

Regarding hardware infrastructure (Figure 6), GPUs were the most widely utilized computing resource, followed by CPUs. AI accelerators, edge devices, and other specialized hardware were used by a relatively small number of respondents, highlighting the continued dominance of traditional GPU computing in AI research and likely increasing the need for training around AI accelerators.

Most respondents reported working with datasets in the range of 1 GB to 1 TB (Figure 7), with the 1 GB to 100s of GB category being the most common. Fewer researchers worked with datasets of either very small size (less than 1 GB) or very large size (greater than 100 TB). This distribution suggests that while big data

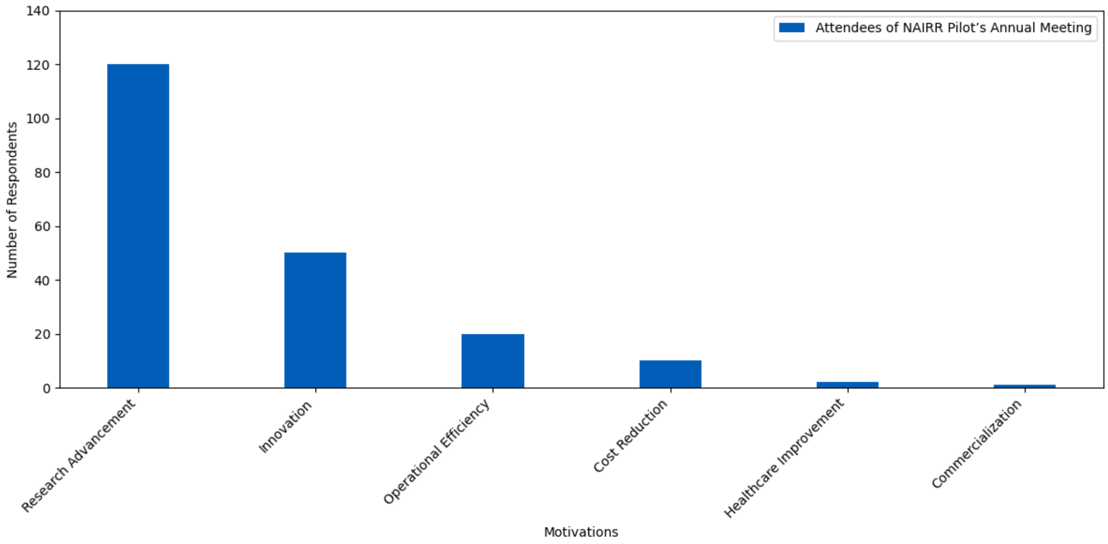

Figure 3: Primary motivations for AI research among attendees

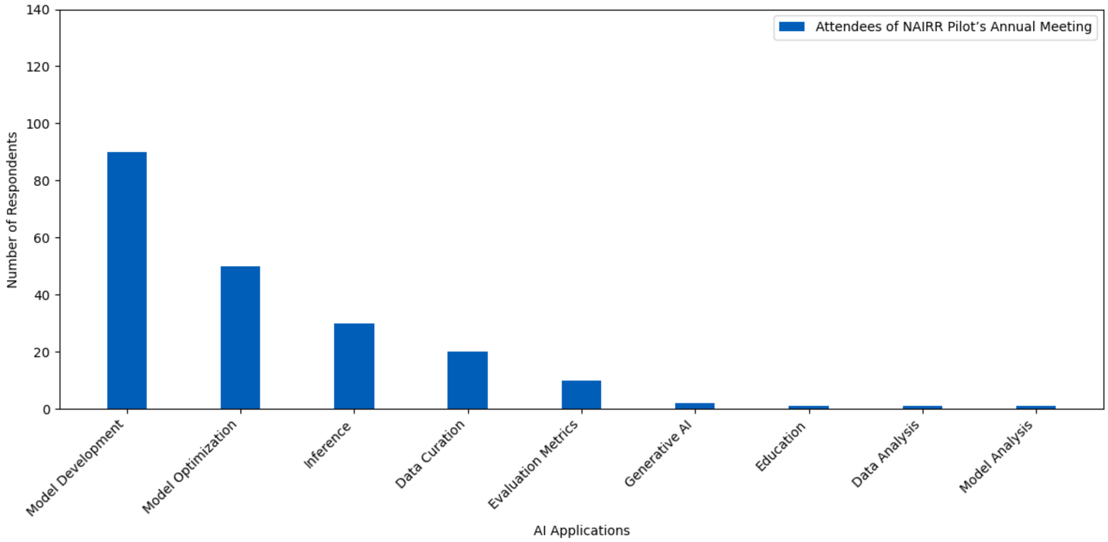

Figure 4: AI applications pursued by attendees

is vital in AI research, truly massive datasets remain relatively uncommon.

The vast majority of attendees reported currently using AI in their research (Figure 8), with only small fractions either planning to use AI or not using it at all. This high adoption rate is expected given the nature of the NAIRR Pilot program and confirms that the community consists primarily of active AI practitioners.

Attendees reported various challenges in implementing AI solutions (Figure 9), with access to infrastructure being the most significant barrier to implementation. Documentation issues ranked second, followed by difficulties in scaling models and optimizing performance. Other notable concerns included software setup, data management, security and privacy, and the interpretation of results. These findings underscore the ongoing need to enhance AI infrastructure and documentation to further advance research capabilities.

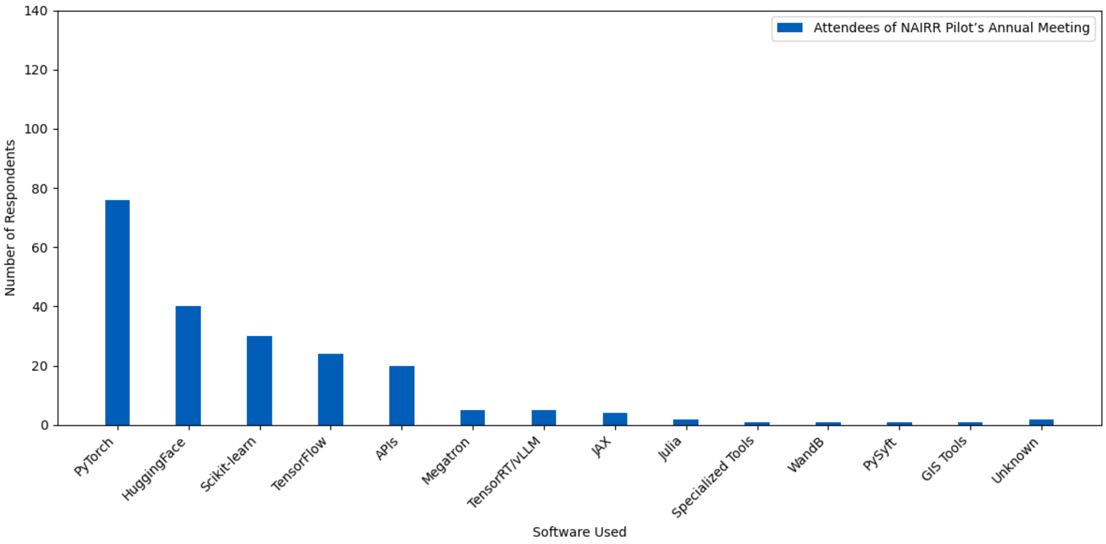

Figure 5: Software frameworks and tools used by attendees

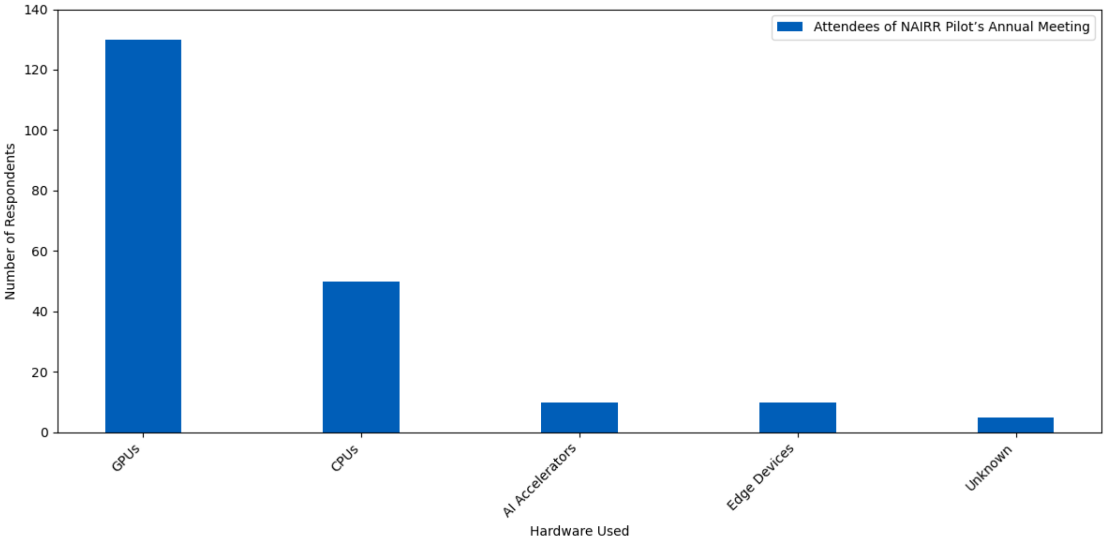

Figure 6: Hardware resources used for AI research

The survey results paint a picture of the NAIRR Pilot community as predominantly comprising expert and advanced AI practitioners from computer science and engineering fields, primarily motivated by research advancements and innovation. Their work focuses on model development and optimization, mainly using PyTorch on GPU infrastructure with medium-sized datasets. Infrastructure access remains the most significant challenge, followed by documentation issues and technical scaling challenges.

These insights can inform the future development of the NAIRR program to better support the AI research community's needs, particularly in addressing infrastructure barriers and improving documentation resources. Additionally, the relatively limited representation of fields outside computer science and engineering suggests an opportunity to broaden participation across a broader range of research domains.

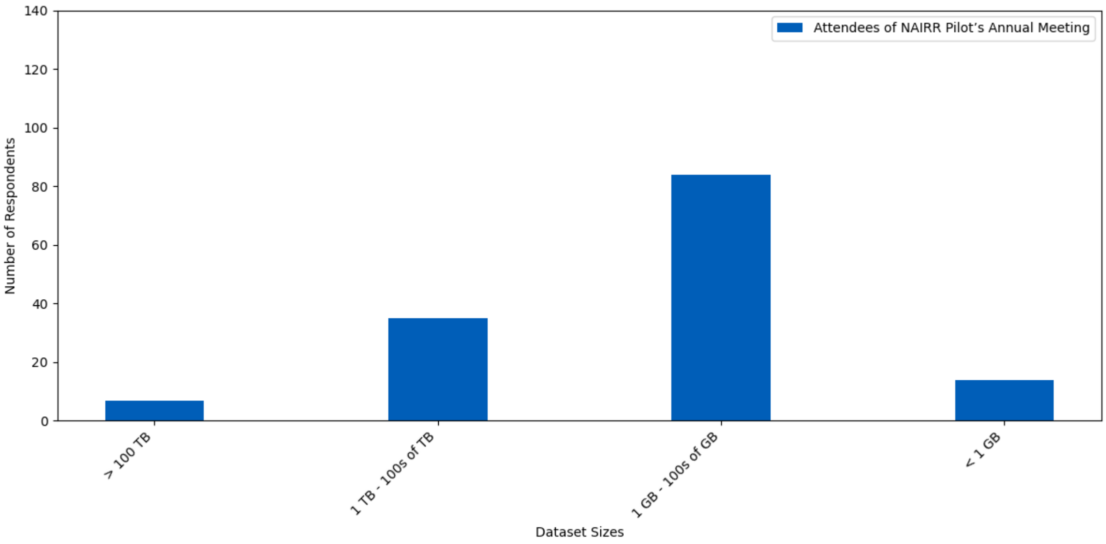

Figure 7: Size distribution of datasets used by attendees

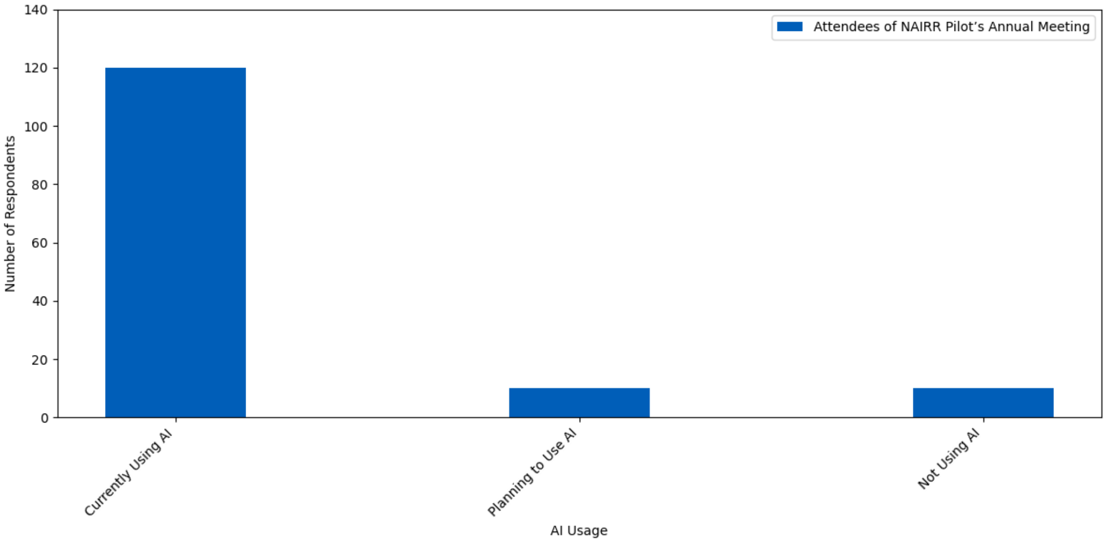

Figure 8: Current status of AI usage among attendees

4 Critical Themes for Advancing an AI Ecosystem

As AI continues to reshape scientific discovery and education, designing a robust and inclusive AI ecosystem becomes ever more pressing. This ecosystem must reconcile state-of-the-art capabilities for large-scale research with user-friendly tools that empower specialists in diverse fields, ranging from computational physics to the social sciences. While cutting-edge methods are necessary to drive innovation, a practical focus on accessibility, standardized workflows, and community-driven development is equally important. The following subsections examine five interrelated areas that collectively form the foundation of a next-generation AI software stack. They highlight common pain points, emerging solutions, and actionable steps for fostering greater collaboration and efficiency.

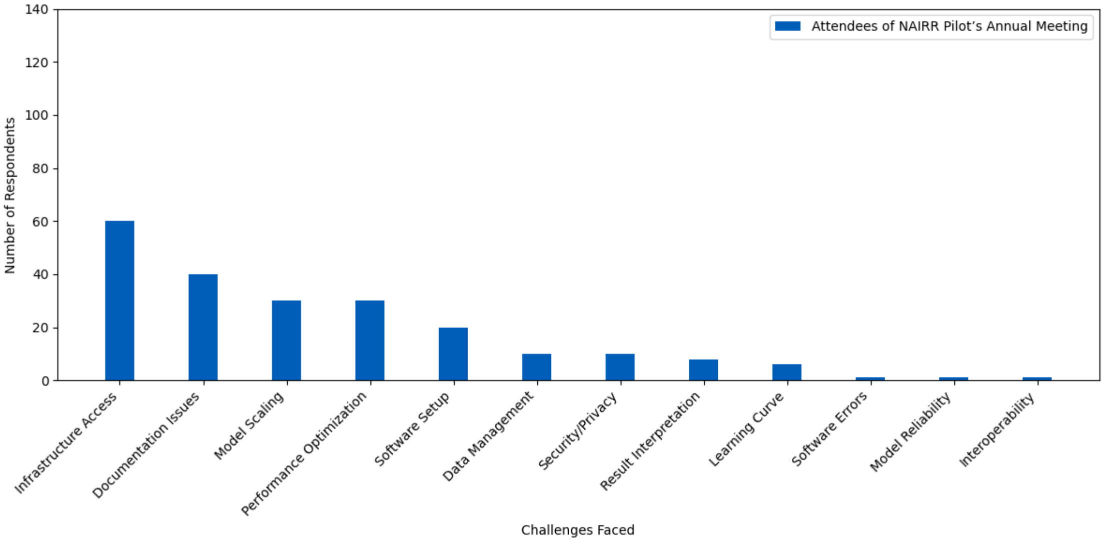

Figure 9: Challenges faced by attendees in implementing AI solutions

4.1 Adoption Barriers

The widespread adoption of next-generation AI software faces a broad range of barriers stemming from rapid technological evolution, heterogeneous computing environments, and the contrasting requirements of training and inference. Developers note that while simple tasks can be easily accomplished with existing frameworks, more advanced needs often prove to be prohibitively complex. Model portability remains a significant challenge, as diverse file formats, checkpoint structures, and hardware-optimized libraries make it difficult to transfer models across platforms, particularly when transitioning from central computing clusters to edge devices. This lack of standardization extends to performance monitoring and debugging, where current tools and methodologies often fail to provide consistent or insightful feedback across varied systems. Additionally, AI stacks emphasize training workflows over inference, resulting in limited support for production-level model deployment and real-time predictions. Another obstacle is the disparity between the approaches favored by commercial cloud providers and the specialized requirements of HPC environments, forcing research teams to manage multiple software versions and bespoke integration layers.

Potential Directions and Approaches: Strategies for overcoming these barriers emphasize balancing industry-standard frameworks with science-specific optimizations. A robust software ecosystem may include recipe-based package management, enhanced performance profiling tools, and standardized checkpoint formats to simplify model transfer and reuse. Providing managed "co-lab" environments—where models, data, and computing resources are integrated—could reduce the overhead of manually configuring complex AI workflows. In parallel, cross-platform abstractions and hardware-agnostic libraries would help mitigate edge-to-cloud heterogeneity, ensuring that both training and inference can be seamlessly scaled. Finally, community-led best practices, reference implementations, and potentially even AI-driven chat interfaces for software configuration and debugging could accelerate productivity. By focusing on well-defined interoperability standards and collaborative open-source culture, the research community can promote broader and more efficient adoption of AI technologies across the full spectrum of scientific applications.

4.2 Deployment Challenges

Developing and deploying next-generation AI software stacks for scientific research involves complex challenges arising from the need to integrate diverse hardware platforms, accommodate varied user skill levels, and meet evolving research demands. A core issue is the variability in computing environments—ranging from commercial cloud platforms to on-premises HPC clusters—each with its own containerization, orchestration, and resource allocation requirements. Successfully orchestrating AI services and infrastructure across these heterogeneous environments necessitates robust workflow frameworks and standards for interoperability. Moreover, while AI training has received significant attention, many researchers now emphasize that efficient inference resources are equally critical. Balancing the requirements of training and inference is complicated by differences in resource availability, deployment costs, and organizational policies.

Another significant challenge is establishing a cohesive software ecosystem that evolves in tandem with the rapidly advancing AI methods. Researchers typically rely on multiple deep learning frameworks, specialized libraries, and domain-specific tools. Ensuring these components remain compatible and secure demands a systematic approach to versioning, vulnerability management, and extensibility. In practice, the "big tent" vision—offering standard AI services across multiple providers—runs up against vendor-specific optimizations and proprietary tools. As a result, selecting a baseline or "minimum viable" technology stack to unify deployments becomes a pressing concern, particularly if hardware and AI frameworks are expected to change over time.

Such difficulties underscore the importance of workflow-centric solutions and flexible resource management strategies that allow users to compose multi-step AI pipelines. Robust support for data federation and secure enclaves is also essential, enabling researchers to move large volumes of data without compromising performance or privacy. Although container-based approaches (e.g., Kubernetes) and HPC schedulers (e.g., SLURM) offer partial solutions, aligning them within a single operational model requires advanced scheduling, orchestration, and packaging techniques that are not yet standardized.

Potential Directions and Approaches: Numerous community-driven efforts aim to address these deployment hurdles. Container-based packaging solutions, coupled with automated build pipelines, can streamline software releases and reduce overhead for researchers. Tools that bundle model repositories, data services, and remote computation frameworks (for instance, systems leveraging model-as-a-service approaches) have shown promise in simplifying model deployment. Additionally, flexible data management and access solutions, combined with HPC-like scheduling for AI tasks, help unify the user experience across traditional supercomputing and cloud-based environments. Projects that leverage open-source libraries and container repositories (e.g., the Extreme-scale Scientific Software Stack (E4S)) can serve as a baseline for institutions to build domain-specific enhancements.

Ultimately, establishing a seamless environment for AI deployment will require striking a balance between standardization and the need for ongoing innovation. Clear documentation, robust APIs, and user-centered design principles—alongside support for new model architectures, platform abstractions, and scalable resource management—can help close critical gaps. These efforts must be guided by researchers' practical experiences, ensuring that the resulting AI ecosystem remains adaptable, interoperable, and responsive to current and future scientific demands.

4.3 Design Challenges

A central challenge in creating an open AI software stack for the NAIRR is addressing the diverse needs of researchers and educators across multiple disciplines. On one hand, experts require advanced capabilities for large-scale data processing and model training, including support for distributed computing, specialized accelerators, and robust performance monitoring. On the other hand, non-expert users in fields such as the social sciences or humanities often need lower technical barriers, minimal setup overhead, and intuitive interfaces that facilitate a quick transition into AI-driven workflows. Balancing these requirements demands careful consideration of mature technologies (e.g., containerization, established deep-learning frameworks) and emerging methods (e.g., federated learning, privacy-preserving training) that make AI more accessible and transparent.

Equally important is the need for reliable data management and provenance, particularly given the increasing complexity of modern AI systems. Researchers must be able to discover, share, and reuse datasets without violating privacy or ownership constraints. Integrating FAIR principles (Findable, Accessible, Interoperable, and Reusable) is essential for sustaining open, collaborative work. At the same time, specialized concerns, such as explainability, safety checks, and auditing, remain inconsistently supported by current software tools. Combining reproducibility and ethical oversight with fast-paced innovation presents a significant design challenge, underscoring the need for modular and adaptable architectures.

Potential Directions and Approaches: A promising avenue for the NAIRR is to build upon robust open-source platforms, integrating them into a cohesive environment tailored for research and education. Collaboration with projects like Tapis, KNIME, and OpenMined can help streamline workflow management, visual programming, and secure data handling. Meanwhile, industry-standard frameworks like PyTorch and Hugging Face provide robust, well-documented foundations for model development, training, and deployment. Bringing these elements together in a "NAIRR Collab"-style platform with preconfigured containers, curated datasets, and user-friendly tooling could significantly reduce the overhead for classroom adoption and early-stage research.

Another beneficial strategy involves designing standardized pipelines for everyday tasks, such as canonical workflows for text classification, image analysis, or multimodal modeling. By abstracting away repetitive setup steps, these pipelines can improve reproducibility, lower entry barriers, and enable domain experts to focus on generating insights rather than wrangling code. Metrics for success may include the breadth of user adoption across various fields, the volume of reproducible publications generated from the system, and the degree to which educational programs incorporate NAIRR-supported resources. Ultimately, an adaptable, transparent, well-documented AI software stack complements industry-led innovations and fulfills the NAIRR mandate to democratize AI for the broader scientific and educational community.

4.4 Outreach

A common barrier to realizing the full potential of AI research resources is a general lack of awareness and guidance within the broader research community. Many prospective users, particularly those outside the AI and HPC domains, remain unaware of the available opportunities or how to navigate the application processes. Even those with preliminary knowledge often struggle with inconsistent documentation, varying facility policies, and the shortage of straightforward, unified "getting started" materials. Educational institutions and research groups note that while short courses and training sessions exist, they are not always effectively publicized or tailored to the needs of newcomers. New users may feel overwhelmed by unfamiliar terminology and competing resource portals in this environment, leading to underutilized allocations and missed research opportunities. Further complicating matters, tight deadlines for proposals and fixed-term allocation periods can deter or disadvantage investigators who require additional time to fully engage with advanced AI infrastructure.

Potential Directions and Approaches: Effective outreach strategies combine structured training opportunities, ongoing community engagement, and transparent, consistent communication. In-person workshops, virtual tutorials, and open-house-style events can help demystify resource usage and reduce the perceived barriers for those new to large-scale computing. Partnering with established conferences, open-source communities, and educational nonprofits provides a direct channel to broader and more diverse audiences, particularly those whose needs differ from those of seasoned HPC and AI specialists. Reference projects and working examples that showcase real-world benefits can guide users through replicable demonstrations, providing a clear path to success. At the same time, a well-organized support network—featuring both online forums and dedicated staff—can provide timely assistance. Clear incentives, such as certificates, scholarships, or recognition for completing training modules, bolster participation and highlight the personal and career benefits. Finally, building user-friendly websites and central resource directories, with intuitive *getting-started* guides and active community channels, ensures that researchers discover the right tools and readily apply them.

4.5 Software Carpentry

Software Carpentry and its sister initiatives in Data Carpentry and Library Carpentry provide foundational coding and data science skills that help researchers use computational methods more effectively worldwide. As AI methods gain prominence in fields ranging from biology to cosmology, there is a growing need to adapt traditional Software Carpentry pedagogies to include AI-focused content and best practices for large-scale and heterogeneous environments. This involves introducing domain scientists and newcomers to concepts such as scripting, version control, reproducible research workflows, and more advanced tools for distributed AI training and inference. Although many researchers are proficient in basic programming, translating these skills to AI-specific tasks—particularly in HPC or large-scale cloud settings—can be daunting without a structured, community-driven curriculum.

Despite broad enthusiasm for leveraging AI in scientific research, multiple challenges persist in integrating Software Carpentry approaches with AI pipelines. Before tackling more advanced AI libraries or specialized frameworks, Beginners require foundational knowledge of the command line, version control, and Python or R. Moreover, domain scientists seeking to incorporate AI into modeling and simulation workflows frequently confront a steep learning curve when debugging complex training jobs, understanding resource allocation, or setting up containerized environments. Meanwhile, advanced users may struggle to make their software compatible with NAIRR's evolving ecosystem or follow consistent standards for documentation, testing, and reproducibility. These gaps point to the need for well-defined community policies covering everything from security practices to interoperability requirements and innovative ways of "badging" or certifying software that aligns with NAIRR's principles.

Potential Directions and Approaches: Building on the community spirit and open-access model that characterize The Carpentries, a tailored AI-focused curriculum could provide structured tutorials and resources, guiding learners from basic coding skills to advanced HPC-based AI workflows. Such content would include example-driven lessons covering AI model lifecycle management, containerization, and collaborative tools for versioning and sharing. Hands-on workshops—co-hosted by NAIRR resource providers, HPC centers, and software development experts—would impart these skills and showcase reference projects and success stories demonstrating how AI can accelerate and enrich scientific discovery. In parallel, a shared repository of best practices and modular training assets (e.g., scripts, containers, notebooks) can help domain scientists adapt lessons to their specific fields. By encouraging community contributions and maintaining an open, iterative development process, Software Carpentry for AI can evolve alongside rapidly changing technologies while focusing on core competencies and reproducible methodologies.

Metrics and Community Engagement Meaningful progress can be tracked through multiple lenses: the number and breadth of workshop participants, the extent to which researchers reuse and extend provided training materials, and the frequency with which lessons and toolkits are cited or adapted in scientific publications. Success might also be measured by creating new *badged* software packages that follow NAIRR-compliant guidelines or by the growth of an online forum where users can discuss challenges and share solutions. Ultimately, a collaborative approach—rooted in open-source ethos and grounded in real-world research needs—will empower scientists at all skill levels to harness the transformative potential of AI tools consistently and sustainably.

4.6 Summary

These five focus areas highlight the breadth and depth of considerations necessary to develop a robust, inclusive AI software stack. Addressing *Adoption Barriers* calls for simplified workflows and standardized checkpoints; managing *Deployment Challenges* requires flexible, interoperable infrastructure; resolving *Design Challenges* depends on striking a balance between cutting-edge capabilities and broad accessibility; prioritizing *Outreach* ensures that new users can discover and leverage available resources; and embracing *Software Carpentry* principles empowers scientists at every skill level to adopt and adapt AI methods confidently. These subsections underscore the importance of harmonizing technical innovation, training, and community engagement to realize AI's transformative potential across the scientific landscape.

5 Insights and Emerging Ideas from Workshop

The workshop concluded by bringing together leading experts in AI, HPC, and data science to explore software-related strategies and governance considerations for the NAIRR Pilot's AI software stack. The panel included specialists with backgrounds in security and governance, data lifecycle and accessibility, AI-driven interfaces, and HPC-AI integration. Their collective goal was to determine best practices, key priorities, and actionable steps for advancing the NAIRR's software ecosystem.

A central theme of the session was the importance of **user-centric design**. Panelists emphasized that the NAIRR's software stack must lower barriers to AI adoption for diverse communities, particularly those without extensive technical backgrounds. Several participants proposed the rapid development of a **NAIRR-GPT**—a chatbot interface leveraging LLM technology to guide new users. This capability should be lightweight and agile to quickly adapt to the rapid progress made by industry in foundational models. This interface would enable researchers to articulate scientific problems in plain language and receive tailored recommendations for relevant models, data sources, and computing resources.

However, panelists cautioned against a ”one-size-fits-all” approach. Given AI's rapid evolution, the NAIRR must remain flexible and agile, allowing domain experts to integrate emerging tools without overhauling the entire infrastructure. Participants agreed that the NAIRR should initially focus on a smaller, well-defined scope—such as educational pilots and foundational user support—before scaling to more complex features or broader user bases.

Governance and security emerged as paramount considerations. One panelist emphasized the importance of defining clear roles, responsibilities, and oversight mechanisms, particularly in relation to software development pipelines, hardware and software supply chains, and data usage agreements. From the beginning, the NAIRR must incorporate robust security measures to protect software integrity, user data, and the research process.

The panel also explored the **intersection of HPC and AI**, noting that traditional HPC workloads differ significantly from AI inference and training tasks. Experts underscored that software systems must accommodate both domains, potentially requiring new instrumentation, monitoring, and resource management layers. Attendees further discussed **community-driven innovation**, pointing to hackathons, targeted workshops, and pilot programs as effective ways to foster collaboration among HPC experts, AI researchers, and domain scientists.

Throughout the discussion, the panelists emphasized the value of **leveraging existing software tools** rather than reinventing them. One contributor recommended identifying tried-and-true solutions in data management, model training, and collaborative analytics, and refining these tools to meet the NAIRR's unique needs. In the process, the NAIRR could fill gaps by supporting new features or ensuring compatibility across a broad spectrum of user requirements.

The panel converged on several core recommendations. First, **prioritize user accessibility and rapid onboarding** through initiatives like NAIRR-GPT. Second, **structure governance and security** from the ground up, clarifying vendor responsibilities and ensuring robust software pipelines. Third, **focus on a manageable initial scope**, then scale the available resources as AI capabilities and community needs evolve. Finally, **promote continuous community engagement**—through hackathons, surveys, and cross-domain collaborations—to keep pace with the fast-changing AI ecosystem. These guiding principles will be the foundation for an inclusive, sustainable, and forward-looking NAIRR software environment.

6 Conclusions and Recommendations

Some of the main conclusions and recommendations of the workshop (in no particular order) are as follows:

1. The NAIRR Pilot's software stack will leverage existing and emerging software solutions, catering to a range of users (from novices to experts) across diverse hardware platforms and accelerators, including those used for education. Notably, the HPC community views these solutions as a layered ***software stack*** optimized for performance and scalability. In contrast, the AI community often refers to a broader ***software ecosystem*** that tightly integrates data, user support, and training frameworks. The NAIRR effort must bridge these perspectives to effectively serve all stakeholders.
2. The stack must respond to the evolving needs of the scientific and AI communities, including real-time data analysis, privacy and security challenges, and portability across emerging AI hardware. Effective data management, encompassing cleaning, curation, and annotation, ensures that researchers can fully leverage the growing volume of diverse datasets.
3. The following software components for the NAIRR stack must remain flexible and extensible to accommodate future technology advances:
    - Operating systems,
    - Middleware solutions for communication and resource management,
    - Languages and compiler support (with emphasis on Python, Julia, C, C++, and Fortran),
    - Workflow managers, and
    - AI-related libraries/models/frameworks, including HPC software that can be leveraged and/or enhanced through AI.
4. Embracing open-source development and ensuring support for new hardware will be essential for keeping the NAIRR stack at the forefront of technological advances.
5. The stack should offer easy-to-use interfaces (e.g., Jupyter Notebooks, web-based platforms) to lower the barrier for newcomers to AI. The stack should also include services that leverage AI technologies and can improve user experience along the computational/experimental lifecycle, including computation/experiment setup, monitoring, debugging, provenance tracking, and result analysis and interpretation. At the same time, training and continuous user support cannot be separated from the software itself, underscoring the need for educational resources and dedicated guidance to help new users navigate this technology.

6. The stack must address near-term user-support needs; for instance, funding small supplements for current grantees during the NAIRR Pilot was proposed. Attendees also recommended creating intuitive *chatbot* interfaces to help users interact with the software stack, further reducing barriers to adoption and ensuring efficient troubleshooting and assistance.

Acknowledgments

This work was supported by the National Science Foundation under award number OAC:2440948. The opinions, findings, and conclusions expressed herein are those of the authors and do not necessarily reflect the views of the National Science Foundation.

A Workshop Participants

Names in **bold** are **technical committee members**, session leads*, speakers$^+$ and session scribes$^\#$.

- Ryan Adamson - Oak Ridge National Laboratory
- Marian Adly - United States Department of Veterans Affairs
- **Ilkay Altintas*** - San Diego Supercomputer Center
- Katie Antypas - National Science Foundation
- Troy Arcomano$^+$ - Argonne National Laboratory
- Aldo Badano$^+$ - United States Food and Drug Administration
- David Balenson - University of Southern California
- Purushotham Bangalore - National Science Foundation
- Shivam Barwey$^\#$ - Argonne National Laboratory
- Nate Bastian - Defense Advanced Research Projects Agency
- Karlo Berket - Lawrence Berkeley National Laboratory
- **Wahid Bhimji*** - Lawrence Berkeley National Laboratory
- Anoushka Bhutani - University of Michigan
- Brian Bockelman - Morgridge Institute for Research
- Nicolae Bogdan - Argonne National Laboratory
- Carl Boettiger$^+$ - University of California, Berkeley
- Zechun Cao - Texas A&M University - San Antonio
- Giuseppe Cerati - Fermi National Accelerator Laboratory
- Dhruv Chakravorty - Texas A&M University
- Kyle Chard - University of Chicago
- Vipin Chaudhary - Case Western Reserve University
- Haipeng Chen - William & Mary
- Yiran Chen - Duke University
- Matthew Cherukara$^+$ - Argonne National Laboratory
- Krishna Teja Chitty-Venkata$^\#$ - Argonne National Laboratory
- Sajal Dash - Oak Ridge National Laboratory
- Ewa Deelman - University of Southern California
- Daniel DeFreez - Southern Oregon University
- Gautham Dharuman$^+$ - Argonne National Laboratory
- **Murali Emani*** - Argonne National Laboratory
- **Nicola Ferrier*** - Argonne National Laboratory / Northwestern University
- Sam Foreman - Argonne National Laboratory
- Sheikh Ghafoor - National Science Foundation
- Josh Greenberg - Sloan Foundation
- Anju Gupta - University of Toledo
- Salman Habib$^+$ - Argonne National Laboratory
- Shawn Haag - University of Minnesota
- Ben Hawks - Fermi National Accelerator Laboratory
- Amr Hilal - Tennessee Technological University
- Khalid Hossain$^\#$ - Argonne National Laboratory
- Shu Hu - Purdue Polytechnic Institute
- Xiaolei Huang - University of Memphis
- Tanzima Islam - Texas State University
- Ali Jannesari - Iowa State University
- **Shantenu Jha** - Rutgers University
- Krishna Kant - Temple University
- Anuj Karpatne - Virginia Tech
- Ian Kash* - University of Illinois Chicago
- Daniel S. Katz - University of Illinois Urbana-Champaign
- Kristopher Keipert - NVIDIA
- Duckbong Kim - Tennessee Tech University
- Farinaz Koushanfar - University of California, San Diego
- Ho-Joon Lee - Yale School of Medicine
- Juan Li - National Science Foundation
- Frank Y. Liu - Old Dominion University
- Miron Livny - University of Wisconsin - Madison
- Raghu Machiraju - Ohio State University
- Mahnaz Maddah - Broad Institute
- Amit Majumdar - San Diego Supercomputer Center
- Manil Maskey - National Aeronautics and Space Administration
- Kenton McHenry - University of Illinois Urbana-Champaign
- Lois Curfman McInnes* - Argonne National Laboratory
- **Diana McSpadden*** - Jefferson National Laboratory
- Bill Miller - National Science Foundation
- Pratik Mukherjee - University of California, San Francisco
- **Anita Nikolich*** - University of Illinois Urbana-Champaign
- Nwamaka Okafor$^\#$ - Argonne National Laboratory
- **Dhabaleswar K. Panda*** - Ohio State University
- **Michael E. Papka*** - University of Illinois Chicago / Argonne National Laboratory
- Jasmin Phua - Datavant

- Marlon Pierce - National Science Foundation
- Elena Pourmal - LifeBoat LLC
- Amina Qutub - University of Texas San Antonio
- David Rabson - Department of Energy
- Nick Rahimi - University of Southern Mississippi
- Subhashini Ramkumar - OpenMined
- Benedikt Riedel - University of Wisconsin
- Mike Ringenburg - Microsoft / Azure HPC & AI
- Jesse Roberts - Tennessee Tech University
- James Rondinelli - Northwestern University
- Varuni Sastry - Argonne National Laboratory
- Ranga Setlur - State University New York - Buffalo
- Shilpika# - Argonne National Laboratory
- Leah Silen - NumFOCUS
- Carol Song - Purdue University
- Biplav Srivastava - University of South Carolina
- Hari Subramoni - Ohio State University
- Al Suarez - National Science Foundation
- **Nathan Tallent*** - Pacific Northwest National Laboratory
- Alastair Thomson - Advanced Research Projects Agency for Health
- Jiachuan Tian - Energy Sciences Network (ESnet)
- Mikhail Titov - Brookhaven National Laboratory
- Karen Tomko - Ohio Supercomputer Center
- Wen-Wen Tung - National Science Foundation
- Archit Vasan - Argonne National Laboratory
- Alex Wadell - University of Michigan
- **Feiyi Wang*** - Oak Ridge National Laboratory
- Yingfeng Wang - University of Tennessee at Chattanooga
- Jim Willenbring - Sandia National Laboratories
- Chandi Witharana - University of Connecticut
- Jianjun Xu - Amazon (AWS)
- Shinjae Yoo - Brookhaven National Laboratory
- Zhao Zhang - Rutgers University
- Huihuo Zheng - Argonne National Laboratory
- Michael Zink - University of Massachusetts Amherst
- Houlong Zhuang - Arizona State University

B Workshop Agenda

Day 1: December 3, 2024

- **08:00 - 09:00** *Breakfast and Check-in (TCS Conference Center)*
- **09:00 - 09:30** *Workshop Overview and Objectives (Room 1416, TCS Conference Center)*
  - 09:00 - 09:05 Introductions (DK Panda and Michael Papka)
  - 09:05 - 09:10 NAIRR Software Workshop Goals and Outcomes (Sheikh Ghafoor)
  - 09:10 - 09:30 NAIRR Overview (Katie Antypas)
- **09:30 - 10:30** *Domain AI Talks: AI in Practice, AI What is Missing (Room 1416)*
  - 09:30 - 09:40 Climate (Troy Arcomano)
  - 09:42 - 09:52 Medical (Aldo Badano)
  - 09:54 - 10:04 Instruments/Experiments (Mathew Cherukara)
  - 10:06 - 10:16 Biology (Gautham Dharuman)
  - 10:18 - 10:28 Environment (Carl Boettiger)
- **10:30 - 10:50** *Surveys (Room 1416)*
  - 10:30 - 10:40 NERSC Survey Results (Wahid Bhimji)
  - 10:40 - 10:50 NAIRR Software Survey Results (Murali Emani)
- **10:50 - 11:15** *Break*
- **11:15 - 12:30** *Breakout Session 1: Software Needs for NAIRR Pilot (Rooms 1416, 1404, 1405, 1406, 1407)*
  - Current Software, Tools, and Gaps - Training and Inference
  - Current Software, Tools and Gaps - Data Management and Storage
  - Current Models, Datasets and Gaps
  - Current Accessibility & Usability and Gaps
  - Current Software, Tools, and Gaps - Security & Privacy
- **12:30 - 13:00** *Summary of Breakout 1 (Room 1416)*
- **13:00 - 14:00** *Lunch (Networking)*
- **14:00 - 15:00** *Cosmology Meets AI: Roadmapping the Final Frontier (Speaker: Salman Habib, Room 1416)*
- **15:00 - 16:00** *Parallel Breakout Session 2: Key Software Features (Rooms 1416, 1404, 1405, 1406, 1407)*
- **16:00 - 16:15** *Break*
- **16:15 - 16:45** *Summary of Breakout 2 (Room 1416)*
- **16:45 - 17:00** *Day 1 Recap and Discussion (Room 1416)*
- **17:00 - 18:00** *Return Shuttles to Crowne Plaza*

Day 2: December 4, 2024

- **08:00 - 09:00** *Breakfast and Networking (TCS Conference Center)*
- **09:00 - 09:15** *Welcome Back and Recap of Day 1 (Room 1416)*
- **09:15 - 10:45** *Parallel Breakout Session 3: Software Adoption & Deployment Challenges (Rooms 1416, 1404, 1405, 1406, 1407)*
- **10:45 - 11:15** *Break*
- **11:15 - 11:45** *Summary of Breakout 3 (Room 1416)*
- **11:45 - 12:45** *Parallel Breakout Session 4: Addressing Short- and Long-Term Objectives (Rooms 1416, 1404, 1405, 1406, 1407)*
- **12:45 - 13:15** *Summary of Breakout 4 (Room 1416)*
- **13:15 - 14:15** *Lunch (Networking)*
- **14:15 - 15:30** *Action Plan for Post-Workshop Process (Room 1416)*
  Panelists:
  - Ilkay Altintas
  - Wahid Bhimji
  - Nicola Ferrier
  - Anita Nikolich
- **15:30 - 16:00** *Break*
- **16:00 - 16:30** *Final Thoughts and Next Steps (Room 1416)*
- **16:30 - 17:00** *Closing Remarks and Adjournment (Room 1416)*
- **17:00 - 18:00** *Return Shuttles to Crowne Plaza*

C Survey Results NAIRR Software Workshop

December 2024

The survey results from the December NAIRR Software Workshop highlight several insightful trends regarding attendee interests and experiences.

Examining the research domains represented at the workshop (Figure 10), computer science emerges as the dominant field, followed by physical sciences, indicating these disciplines are heavily engaged with AI software. Biological and environmental sciences, along with engineering, represent smaller yet significant user bases.

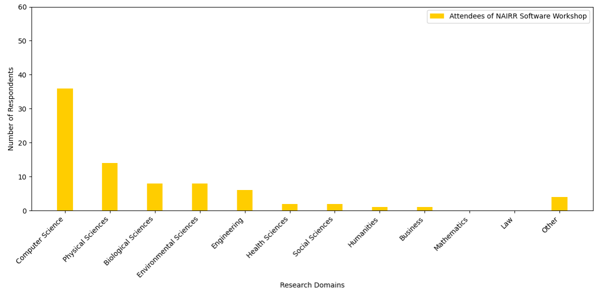

Figure 10: Research domains

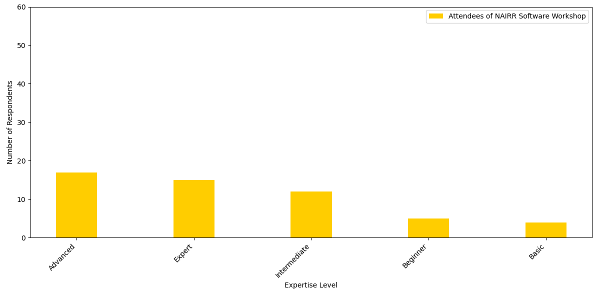

Figure 11: Expertise level

In terms of expertise (Figure 11), the workshop audience primarily consists of advanced and expert-level users, illustrating the depth of knowledge and skill within the community. Intermediate users also comprise a notable segment, with beginners and participants at the basic level constituting a smaller fraction.

The challenges identified by participants (Figure 12) primarily include access to infrastructure, data management, and model scaling. Software setup and documentation issues also represent significant hurdles, pointing to areas where improved tools and resources could benefit the community.

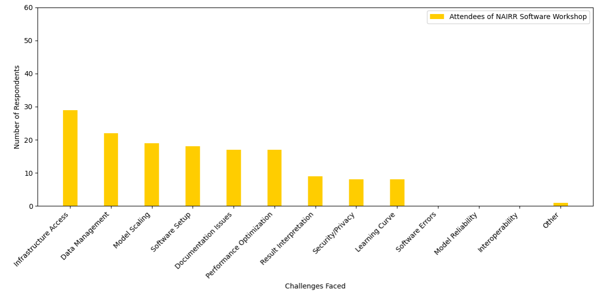

Figure 12: Challenges faced in using AI tools and infrastructure

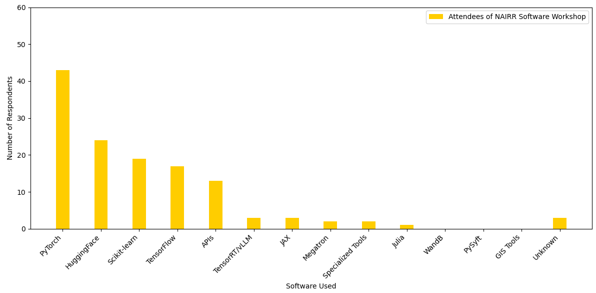

Figure 13: Software used

The analysis of software tools utilized by attendees (Figure 13) reveals a strong preference for PyTorch, HuggingFace, and Scikit-learn, reflecting their prominence in AI and machine learning communities. TensorFlow and various APIs are also notably used, though to a lesser extent.

When considering hardware preferences (Figure 14), GPUs stand out as the most widely used hardware, aligning with their suitability for computationally intensive AI tasks. CPUs and edge devices also play substantial roles, though specialized AI accelerators remain less common.

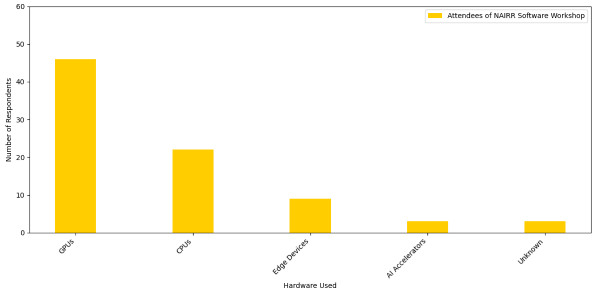

Figure 14: Hardware used

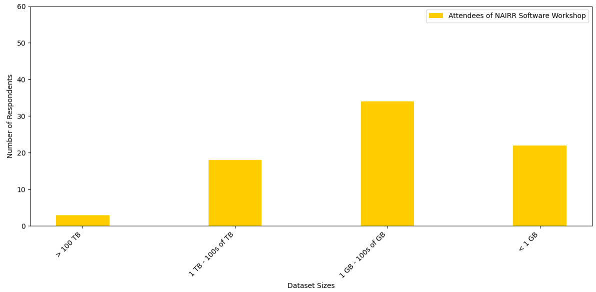

Figure 15: Dataset sizes

Dataset sizes utilized by attendees (Figure 15) tend to cluster predominantly between gigabytes and hundreds of gigabytes, with fewer attendees working with terabyte-scale datasets. This suggests moderate-to-large-scale data analysis is standard among participants.

AI usage status (Figure 16) indicates nearly universal engagement, with most attendees actively using AI technologies in their work. Only a minimal fraction is currently not using AI but is planning to adopt it.

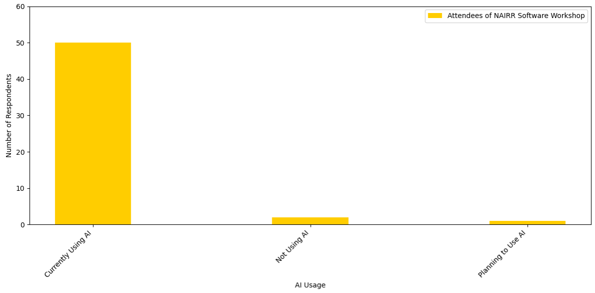

Figure 16: AI usage

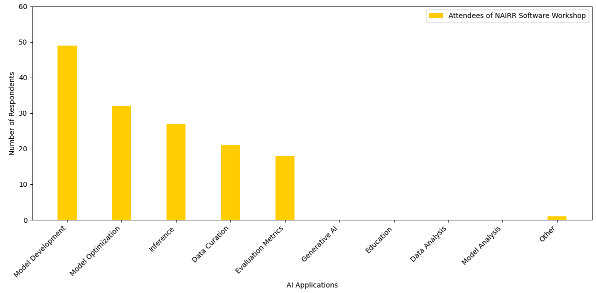

Figure 17: AI applications

AI applications among attendees (Figure 17) are most prominent in model development and optimization, with inference and data curation closely followed. Evaluation metrics are also an important focus, emphasizing robust and validated AI solutions within the community.