Position Papers for the ASCR Workshop on Cybersecurity and Privacy for Scientific Computing Ecosystems

https://www.orau.gov/2021ascr-cybersecurity

Stacy Prowell[1], David Manz[2], Candace Culhane[3], Sheikh Ghafoor[4], Martine Kalke[5], Kate Keahey[6], Celeste Matarazzo[7], Chris Oehmen[2], Sean Peisert[8], and Ali Pinar[9] (eds.)

November 2021

[1]Oak Ridge National Laboratory
[2]Pacific Northwest National Laboratory
[3]Los Alamos National Laboratory
[4]Tennessee Tech University
[5]Massachusetts Institute of Technology, Lincoln Laboratory
[6]Argonne National Laboratory
[7]Lawrence Livermore National Laboratory
[8]Lawrence Berkeley National Laboratory
[9]Sandia National Laboratories

Disclaimer

U.S. Department of Energy Points of Contact:
Hal Finkel, hal.finkel@science.doe.gov, Margaret Lentz, Steven Lee, and Robinson Pino

https://doi.org/10.2172/1843573

Contents

Halappanavar, Mahantesh	Transformer-based Frameworks: Bridging the Gap from Machine Readable Data to Actionable Cyberdefense Insights
Johnson, Grant	Trusting 5G and edge components for controlling instrumentation
Johnstone, Patrick	Federated Scientific Machine Learning
Jung, Taeho	Secure and Private End-to-End Provenance Capture and Verification for Multi-party Data Workflow
Kantarcioglu, Murat	Secure, Robust, and Accountable Federated Learning using Trusted Execution Environments
Kellison, Ariel	Formal Methods-based Certification Frameworks for Scientific Computing Applications
Khairy, Sami	Reliable Privacy-Preserving Peer-2-Peer Distributed Reinforcement Learning for Science (RP4RL)
Kim, Kibaek	Challenges in Algorithm Design for Privacy-Preserving Federated Learning
Kotevska, Olivera	Challenges with Sensitive Data in Distributed Graph Settings
Lange, Jack	Secure Compartmentalization and Partitioning of System Software
Lawrence, David	Federated Analysis of Experimental Data
Lewis, Cannada	Federated Tensor Models for Data Integrity
Li, Hai	Towards Privacy-Preserving and Robust Federated Learning
Li, Mingyan	Toward Effective Security/Reliability Situational Awareness via Concurrent Security-or-Fault Analytics (SoFA)
Liao, Chunhua	Research Opportunities for Ensuring HPC Data Integrity and Provenance
Lin, Yuewei	Toward Automated Defense Against The Adversarial Attacks and Forgeries in National Security Video Surveillance
Lofstead, Jay	Fine-Grained Containerization for Data Authentication, Security, and Portable Workflows
Machanavajjhala, Ashwin	Negotiating Privacy Utility Trade Offs under Differential Privacy
Madireddy, Sandeep	An Information-Theoretic View of Learnable Privacy-Utility Trade-off for Scientific Data
Michael, J. Robert	Addressing the Limitations to Distributed Learning Containing Sensitive Data
Moriano, Pablo	Graph Algorithms for Quality and Security of Scientific Computing Ecosystems

Norman, Andrew	A Platform For Controlled Access to Heterogeneous Computing for High Energy Physics
Pirkelbauer, Peter	Trustworthy Computing with Edge Devices
Prasadan, Arvind	Sketching Algorithms in Distributed Systems
Ren, Yihui	Communication-free Secure Multi-Party Computation for Deep Neural Network Training using HPC
Shi, Xinghua	Secure and Trustworthy Machine Learning for Life Sciences: Opportunities and Challenges
Son, Seung Woo	Challenges Towards Efficient Anomaly Detection for Improving HPC Data Integrity
Stanley, Chris	Practical Privacy-Preserving Algorithms for Efficient Knowledge Dissemination at DOE Facilities
Teranishi, Keita	Leveraging Fault Tolerance for Secure Scientific Computing
Tombs, Vandy	Differential Privacy is not Privacy
Vanderbruggen, Tristan	Machine Learning for Identifying Vulnerable Libraries in Binary Code
Varshney, Lav	Blockchain and the Scientific Method
Wang, Bao	A Flexible Framework for Privacy-Preserving Scientific Computing
Wang, Dexin	Integrated Workflow Management for Scientific Research on Complex Systems
Wu, John	Data Integrity and Provenance in Self-Driving Facilities

ASCR Workshop on Cybersecurity and Privacy of Scientific Computing Ecosystems, Nov 3-5, 2021

At the request of the Department of Energy's (DOE) Office of Advanced Scientific Computing Research (ASCR), this program committee has been tasked with organizing a workshop to identify basic research needs in cybersecurity and privacy to better support DOE's science and energy mission.

As part of the process, the program committee is soliciting community input in the form of position papers to help identify significant use cases, facility issues, and other barriers to enabling verifiably trustworthy computational science while preserving data confidentiality as appropriate for scientific workflows of interest to DOE. The program committee will review these position papers and based on the fit of their area of expertise and interest, selected contributors will have the opportunity to participate in the workshop currently planned as a virtual event November 3-5th, 2021.

ASCR is one of the six interdisciplinary scientific program offices within the Office of Science (http://science.energy.gov/) along with Basic Energy Sciences, Biological and Environmental Research, Fusion Energy Sciences, High Energy Physics and Nuclear Physics. ASCR's mission is to discover, develop, and deploy computational and networking capabilities to analyze, model, simulate, and predict complex phenomena important to the DOE (http://science.energy.gov/ascr/).

The thrust areas that will be explored by this workshop are the following:

- Algorithms for secure, scalable, privacy-enhancing technologies and frameworks, including
    - Federated AI/ML
    - Differential privacy
    - Randomized algorithms
    - Adversarial modeling & simulation
    - Graph algorithms
    - Formal methods
- Platforms to support the entire scientific-computing ecosystem, including edge computing for large-scale experiments, focusing on heterogeneous systems and distributed systems, including
    - Heterogeneous computing systems
    - Distributed computing systems
    - Secure data architectures
- Data workflows to allow agile use of data while preserving integrity and privacy, making the important properties verifiable either at runtime or post-computation, including
    - Integrity and provenance
    - Data management infrastructure

Topics that are out-of-scope for the workshop include discussing specific proposed solutions or areas that are clearly out of DOE's fundamental and applied-sciences mission scope, e.g., cryptography, enterprise security, and general-operations technology.

Requirements:

Position papers should identify a use case that is blocked by issues related to security and privacy, and identify future research directions that are relevant to the goals of the workshop. Each position paper should provide contact information (name, institution, email address) for a single corresponding author and should be no more than 2 pages in length (no smaller than 11-point font), including cited references.

Registration and Submission:

Registration information: ASCR Workshop on Cybersecurity and Privacy for Scientific Computing Ecosystems

Link for submission of position papers: Position Paper Submission

Structure:

The workshop will consist of an initial DOE/ASCR overview and invited talks followed by break-out discussion sessions. The break-out discussion sessions will be organized around the three thrust areas described above. Workshop discussions will help create a report that will be submitted to ASCR. The schedule will span three five-hour days.

Summary:

Position Paper Length and Format: Up to 2 pages (references included), at least 11-point font PDF file.
Due Date: **11:59 PM ET on October 14, 2021**
Notification of Selection: **October 20, 2021**
Workshop Date: **November 3-5, 2021**

Program Committee:

- David Manz, Pacific Northwest National Laboratory
- Stacy Prowell, Oak Ridge National Laboratory
- Kevin Barker, Pacific Northwest National Laboratory
- Diana Burley, American University
- Candy Culhane, Los Alamos National Laboratory
- Sheikh Ghafoor, Tennessee Technological University
- Martine Kalke, MIT Lincoln Laboratory
- Katarzyna Keahey, Argonne National Laboratory
- Celeste Matarazzo, Lawrence Livermore National Laboratory
- Angela Norbeck, Pacific Northwest National Laboratory
- Christopher Oehmen, Pacific Northwest National Laboratory
- Sean Peisert, Lawrence Berkeley National Laboratory
- Ali Pinar, Sandia National Laboratories

Sponsor:

Department of Energy, Office of Science, Advanced Scientific Computing Research
ASCR Points of Contact: Hal Finkel (Hal.Finkel@science.doe.gov), Steven Lee, Margaret Lentz, and Robinson Pino.

Shrirang Abhyankar, Henry Huang
Pacific Northwest National Laboratory
shri@pnnl.gov, zhenyu.huang@pnnl.gov

Secure Heterogenous Algorithms for Energy System Optimization: Needs and Challenges

The electrical power grid is at the heart of a complex system of interdependent critical infrastructures. Any disruption to it leads to dire consequences to the functioning of the society and the economy. With increasing threats from natural disasters, such as hurricanes, wildfires, etc., and human-initiated cyber events, the electrical grid is facing increasing operational time. At the same time, the electric power grid is undergoing a renaissance spurred by the deeper penetration of cleaner renewable energy sources, increased adoption of electric vehicles, proliferation of distributed sensing and control devices (PMUs, Smart Meters), and at-home electricity production through solar panels, storage devices, etc. The incorporation of these new technologies also brings with additional planning and operational complexities.

Grid planners and engineers heavily rely on computational analysis and models to assess and plan the grid operations. Various applications require fast and accurate solutions to computational problems ranging from nonlinear solves to mixed integer optimization problems. Moreover, with the increasing interdependence between the critical infrastructures (for e.g., reliance of the electrical power grid on gas network), this computational problem is expanding to a complex system of systems problem requiring solution to larger multi-physics problems.

While the technological advancements are making the grid cleaner and more efficient, it is also burgeoning the data and computing requirements needed for the analysis. The use of desktop computers and/or small local clusters is becoming insufficient for the computational problem needs, and the increasing data storage needs are pushing the boundaries.

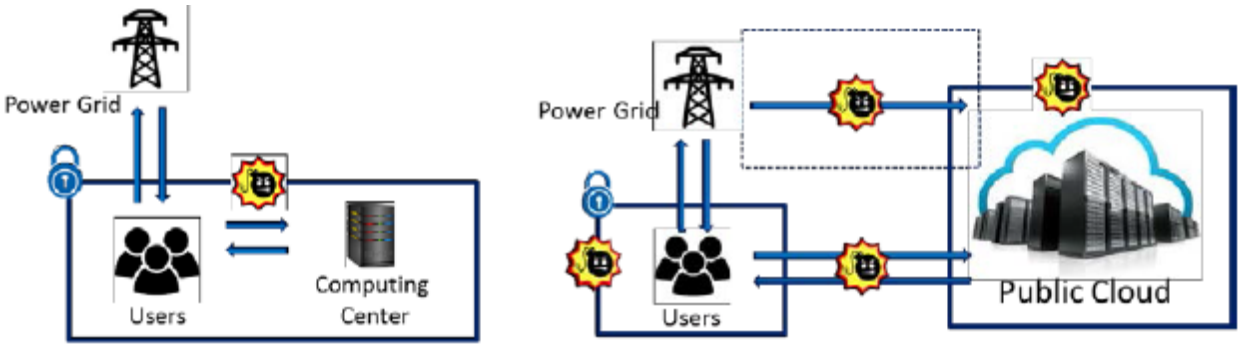


Figure 1: Traditional computing platform paradigm (left) versus next-generation computing platform paradigm. Source [1]

A promising avenue for scaling the computations is using large-scale computing facilities, such AWS cloud or DOE leadership computing facilities. The increasing use of heterogenous systems with CPU-GPU architectures is another promising avenue for further speeding up the

computations. In the ECP project ExaSGD [2], we are working on developing scalable heterogenous algorithms for power grid optimization. However, for reliably using these algorithms in an operational environment, additional layer of security needs to be embedded to ensure the trustworthiness of these algorithms. Below, we outline several of challenges in introducing security while adhering to the computational requirements.

- **Adverserial attacks:** This is perhaps a common thread to many domains, but it is a very important concern for power grid applications due to the sensitivity and confidentiality of the data. Attackers trying to manipulate the data or inject false data could cause failure of the underlying algorithm or an incorrect result computed causing mis-operation when the result is realized on the physical grid.
- **Time-criticality:** Power grid applications are time-critical needing the computations done in a specific time-frame. For instance, the security-constrained economic dispatch (SCED) application is run every five or fifteen minutes, and contingency analysis is run every few minutes. Any added security-layers need to be light-weight, minimally impacting the time-criticality aspect.
- **High overhead:** Some security algorithms, such as encryption methods, are difficult to apply and are computationally intensive.
- **Algorithmic complexities due to architecture:** While there has been work on secure algorithms based on CPU architectures, it is unclear whether the same algorithms will have the same accuracy and scalability on heterogenous systems. The SIMD architecture of GPUs slow-down some algorithms that run fast on CPU, for example algorithms that require matrix factorizations.

[1] A Resilient and Trustworthy Cloud and Outsourcing Security Framework for Power Grid Applications. Accessed at https://www.energy.gov/sites/default/files/2021-04/A%20Resilient%20and%20Trustworthy%20Cloud%20and%20Outsourcing%20Security%20Framework%20for%20Power%20Grid%20Applications%20-%20ANL.pdf
[2] Exascale Grid Dynamics (ExaSGD) project, accessed at https://www.exascaleproject.org/research-project/exasgd/
[3] Trustworthy cyber infrastructure for the power grid, accessed at https://tcipg.org/about-us.html
[4] U.S. government steps-up cyber security to safeguard power grids and software from malicious hacker attacks, accesed at https://www.militaryaerospace.com/trusted-computing/article/14200740/cyber-power-grid-software
[5] Workshop on research directions for security and networking in critical real-time and embedded systems, accessed at https://arcb.csc.ncsu.edu/~mueller/crtes06/ https://arcb.csc.ncsu.edu/~mueller/crtes06/papers/007-final.pdf

Cybersecurity and Privacy for Instrument-to-Edge-to-Center Scientific Computing Ecosystems

Ryan Adamson (adamsonrm@ornl.gov), Christian Engelmann (engelmannc@ornl.gov)
Oak Ridge National Laboratory, Oak Ridge, TN, USA

Challenge: The DOE's Artificial Intelligence (AI) for Science report [1] outlines the need for intelligent systems, instruments, and facilities to enable science breakthroughs with autonomous experiments, "self-driving" laboratories, smart manufacturing, and AI-driven design, discovery and evaluation. The DOE's Computational Facilities Research Workshop report [2] identifies intelligent systems/facilities as a challenge with enabling automation and eliminating human-in-the-loop needs as a cross-cutting theme. Autonomous experiments, "self-driving" laboratories and smart manufacturing employ machine-in-the-loop intelligence for decision-making. Human-in-the-loop needs are reduced by an autonomous online control that collects experiment data, analyzes it, and takes appropriate operational actions in real time to steer an ongoing or plan the next experiment. DOE laboratories are currently in the process of developing and deploying federated hardware/software architectures for connecting instruments with edge and center computing resources to autonomously collect, transfer, store, process, curate, and archive scientific data. These new instrument-to-edge-to-center scientific ecosystems face several cybersecurity and privacy challenges.

Computing systems from different administrative domains with different cyber security policies are interconnected with each other. This may involve instrument control systems, laboratory robots/automation/sensors, edge computing devices for real-time processing, Cloud-like computing for design of experiments and DOE's Leadership computing systems for scientific data analyses and digital twins. Scientific data, including experiment setup data, control data, and results, is then transferred between and processed in these different administrative domains. Resource orchestration across administrative domains ensures that required resources, such as network and compute, are available when needed and at the required capacity. The involved administrative domains may be different network enclaves within a DOE laboratory, or involve the experimental facilities of outside organizations such as other laboratories, universities, and industry. Some of these instrument-to-edge-to-center scientific ecosystems may even cross country boundaries to connect unique experimental facilities with unique computing capabilities.

The specific cybersecurity and privacy challenges for such instrument-to-edge-to-center scientific ecosystems are multifold and include bridging the differences in cyber security policies of several administrative domains, ensuring operational safety and security of experimental facilities and guarding the privacy and integrity of scientific data. Ecosystem computing can be represented as a set of the composable building blocks (system of systems) used by scientific workflows. Unfortunately, no single organization has the authority, responsibility, or capability to secure multi-organizational interconnected systems. ***Thus, the wholistic application of cybersecurity and privacy to these interconnected systems must ultimately be owned by the scientific workflow operators themselves***. The workflow itself is the only layer of this model that interacts with the complete set of systems, which often do not expose security, trust, and assurance primitives to scientists.

Today, the individual systems that make up autonomous experiments, "self-driving" laboratories, and smart manufacturing are *already* secured to appropriate standards set by their managing organizations. Yet, security and privacy concerns are not being wholistically addressed for scientific workflows. The NIST risk management framework [3] is commonly used within DOE for this purpose, but the resulting Confidentiality, Integrity, and Availability assurance levels are incredibly coarse; systems receive just one of three ratings: Low, Moderate, or High! These categories are insufficient since organizations rated at the same level are rarely able to interconnect systems without some normalization. What can be done to facilitate trust between various experimental facilities, networks, edge devices, and supercomputer centers?

The most effective security tools available today are inadequate for a multi-organization system of systems. Mandatory Access Control (MAC) systems such as AppArmor and SELinux only operate at the single node level within a single organization. Firewall rules are largely implemented to police IP addresses instead of data content because the very nature of inter-organization communication is endpoint-based and not data-centric. Recent research in Zero Trust Architectures (ZTA) has led to successful implementation and best practice development, but implementation of ZTA is still typically limited in scope to single applications, protocols, or organizations. ***How can we ensure privacy and security of distributed systems when tools are fundamentally based on a localized security model?***

Opportunity: Several research areas exist regarding the security of scientific computing ecosystems:

1) Security and privacy risk bounding techniques such as Uncertainty Quantification (UQ) have not yet been applied wholistically to systems, such as edge sensors, network interconnects, backing storage, and computational systems. ***Fine-grained UQ may enable scientific workflow operators to make strategic decisions about which systems to use and which ones to avoid in order to attain an acceptable measure of scientific data privacy and integrity assurance***.

2) Mandatory access control primitives are not present in distributed, highly-scalable ecosystem computing. Research and development is needed to discover, define, and integrate these primitives into scientific workflows and systems. ***Distributed mandatory access control methods may need to be integrated by workflow operators and subsequently enforced by experimental facilities to the satisfaction of all organizations spanned by autonomous workflows.***

3) Network security techniques have developed around the fact that communication *endpoints* are easier to protect than the data itself, but *data* protection at the network layer is needed. New data-centric communication protocols such as Content Centric Networking (CCN) and Named Data Networking (NDN) promote data sets to first class citizen status and replace the Internet Protocol (IP) layer of the network stack. NDN in particular requires cryptographically signed data packets at the lowest levels of the networking stack. ***When data trustworthiness is provided at the network layer, the application layers above become much more secure and much less complex***.

Timeliness or maturity: The instrument-to-edge-to-center scientific ecosystem is extremely complex. Today, there are already 300 independently maintained workflow solutions [4]. Tools to reduce and bound complexity and risk are urgently needed to meet the privacy and cybersecurity requirements of all the stakeholders that participate in this new scientific ecosystem. Without techniques to assess and trust the security of distributed workflows, many organizations will not be able to provide resources to enable "self-driving" laboratories and machine-in-the-loop workflows. Research into these opportunities is timely; UQ research has been applied recently to quantify AI model accuracy, and a recent Executive Order [5] has highlighted the need for implementing Zero Trust networking models across the federal government. MAC is a key component of Zero Trust, and implementation of NDN can essentially enable Zero Trust networking for free (for all layers of the networking stack) since data provenance is inherently necessary to the protocol.

References:

[1] **AI for Science Report.** March 2020. URL https://www.anl.gov/ai-for-science-report
[2] **DOE National Laboratories' Computational Facilities – Research Workshop Report.** ANL/MCS-TM-388. February 2020. URL https://publications.anl.gov/anlpubs/2020/02/158604.pdf
[3] **NIST Risk Management Framework.** 2018. URL https://doi.org/10.6028/NIST.SP.800-37r2
[4] **List of Computational Data Analysis Workflow Systems**. URL https://s.apache.org/existing-workflow-systems
[5] **Executive Order on Improving the Nation's Cybersecurity.** May 2021. Executive order 14028

Secure Computing with Heterogeneous Scientific Computing Workflows

Matthew Anderson
Idaho National Laboratory
Matthew.anderson2@inl.gov

The Department of Energy's (DOE's) key scientific workflows are increasingly being driven by users with requirements for not only storage encryption but also confidential computing via trusted execution environments (TEE) with remote attestation. The scientific computing workflows of industry collaborators often include the use of export restricted or controlled access codes that might be only accessible to users when running on DOE high performance computing (HPC) systems. These codes include heterogeneous applications designed for deployment on general purpose graphics processing units (GPGPUs) that figure prominently in the DOE roadmap for planned Exascale computing architectures. This position paper highlights a use case representative of multiple nuclear energy partners from industry. These partners use DOE licensed codes on DOE HPC systems via the Nuclear Computational Resource Center (NCRC) but cannot execute confidential computing with heterogeneous workflows *since no such TEE exists at present.* This is an urgent topic for future research. We expect this TEE requirement only to grow as users increase the safeguards over their data, intellectual property, and research. We also anticipate the need for heterogeneous confidential computing to extend to other industries outside nuclear energy and even to more general DOE research computing environments.

TEEs are currently available via software guard extensions (SGX) [1] for Intel CPUs, TrustZone [2] for ARM CPUs, Komodo [3] for ARM CPUs, Keystone [4] for RISC-V CPUs, Graviton [5] for GPGPUs, and Telekine [6] for GPGPUs. However, attempts at TEEs appropriate for CPU-GPGPU applications have slowed down application performance by as much as 46 times [7], thereby negating any benefit for the heterogeneous architecture. GPGPU memory is especially difficult to secure; data that move between the CPU and GPGPU are visible and vulnerable to both administrators and malicious attackers—*yet this is the exact type of data movement that is crucial to future HPC architectures to reach Exascale computing capability* and overcome the performance barriers resulting from the end of Dennard scaling and Moore's law.

The performance penalty for using a TEE can vary widely based on the application. When using a GPGPU TEE on just one or two GPUs, Telekine reports a performance penalty for graph algorithms from 18–41% and for machine learning from 0–8% [6]. DOE heterogeneous scientific computing workflows are generally dynamic and adaptive with hard-to-predict memory access patterns that look more similar to graph algorithms than to the static, predictable memory access patterns associated with machine learning frameworks. It is highly likely that DOE scientific computing workflows will be among the most complex applications for securing with a TEE and will also involve the largest performance penalty. Reducing this penalty in the context of heterogeneous scientific computing workflows will be paramount for advancing future end-user applications like those from nuclear energy industry users.

Typical heterogeneous scientific computing workflows can involve many thousands of accelerators and operate at a scale where most TEEs have never been tested. Up to now, accelerators have largely been GPGPUs provided by a single vendor, but this is changing. Multiple vendors will be providing GPGPUs for DOE HPC systems in the near future, each with different memory hierarchies, data movement schedulers, and varying modalities for extracting concurrency. These variations will directly impact the design and deployment of the hardware assisted TEEs necessary to support the accelerators and the systems that host them. Emerging accelerator technologies will only increase the level of heterogeneity as processing-in-memory, field programmable gate arrays, application specific integrated circuits, and

non von Neumann architectures arrive as core components in HPC workflows. Each will add complexity to designing, implementing, and deploying hardware assisted TEEs to support confidential computing as required by users.

HPC users seeking to protect themselves against unauthorized data breaches and theft of intellectual property must either set up their own internal datacenter and trust that neither privileged account holders nor malicious attackers will steal data, or, alternatively, use external cloud or DOE HPC resources and trust that the external HPC system policies and procedures will prevent the theft of data. This position paper highlights a use case where nuclear industry users *are required to use DOE HPC resources* to access DOE licensed codes. Because of the lack of existing hardware assisted TEEs for heterogeneous scientific computing workloads, this limits or even precludes the full realization of their work. While our use case has centered on nuclear energy industry users, confidential computing requirements readily extend to other industries, including renewable energy, healthcare, pharmaceutical, finance, and material design. As DOE develops solutions for accelerator supported hardware assisted TEEs, these will certainly be adopted and leveraged in other industries outside DOE HPC.

References

[1] Victor Costan and Srinivas Devadas. Intel SGX Explained. 2016.

[2] Arm Limited. Introducing Arm TrustZone. https://developer.arm. com/technologies/trustzone.

[3] Andrew Ferraiuolo, Andrew Baumann, Chris Hawblitzel, and Bryan Parno. Komodo: Using Verification to Disentangle Secure-enclave Hardware from Software. In Proceedings of the 26th Symposium on Operating Systems Principles, SOSP '17, pages 287–305, New York, NY, USA, 2017. ACM.

[4] Dayeol Lee, David Kohlbrenner, Kevin Cheang, Cameron Rasmussen, Kevin Laeufer, Ian Fang, Akash Khosla, Chia-Che Tsai, Sanjit Seshia, Dawn Song, and Krste Asanovic. Keystone Enclave: An Open-Source Secure Enclave for RISC-V. https://keystone-enclave.org/files/ keystone-risc-v-summit.pdf, 2018.

[5] Stavros Volos, Kapil Vaswani, and Rodrigo Bruno. Graviton: Trusted Execution Environments on GPUs. In USENIX Symposium on Operating Systems Design and Implementation (OSDI), 2018.

[6] Tyler Hunt, Zhipeng Jia, Vance Miller, Ariel Szekely, Yige Hu, Christopher J. Rossbach, and Emmett Witchel. Telekine: Secure Computing with Cloud GPUs. 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20), 2020.

[7] Wenting Zheng, Ankur Dave, Jethro G. Beekman, Raluca Ada Popa, Joseph E. Gonzalez, and Ion Stoica. Opaque: An Oblivious and Encrypted Distributed Analytics Platform. In USENIX Symposium on Networked Systems Design and Implementation, NSDI, 2017.

ASCR Workshop on Cybersecurity and Privacy of Scientific Computing Ecosystems

Position Paper: Large-Scale Resilient Collaborative Machine Learning

Corresponding author: Olugbenga Moses Anubi, Florida State University, oanubi@fsu.edu,

The past few decades have seen a tremendous increase in the volume and complexity of data generated in scientific discovery processes. Moreover, due to the rapid growth in internet and networking technology, it is now common for these experiments to be composed of geographically dispersed components. Each of the components generates and stores a huge dataset which captures only a portion of the global phenomenon in question. This poses a tremendous challenge for data analysis, even with the most advanced Machine Learning/ AI methods. The state-of-the-art approaches to this problem involve either routing data to a trusted central location where the learning task takes place or iteratively performing the learning task over the dispersed data sources. However, in addition to low efficiency issues and high cost, there is often a single point of failure, resulting in low resiliency to faults and adversarial targeting.

Use Case: Large-scale resilient collaborative learning of proprietary heterogenous models over proprietary non collocated datasets while preserving privacy.

This use case involves collaborative learning among a large number of stakeholders. Each stakeholder has private models and datasets that should not be shared with the other stakeholders. However, their datasets only contain partial information. Training on only such datasets will result in systems that don't generalize well. It is therefore essential to leverage the dissimilarity of all datasets to the mutual benefit of all stakeholders involved. **So, the challenge is to develop learning systems and frameworks that will simultaneously train all models using all datasets without violating privacy requirements and provide resiliency against bad actors.** State-of-the-art Federated AI/ML is very good at training one model over non collocated datasets. Privacy preservation can also be guaranteed with some newer variants [1] [2]. However, the fundamental concept and available frameworks do not generalize to this use case. Newer concepts and frameworks need to be developed to address this open, but very important, challenge.

Distributed Learning (DL), Federated Learning (FL) and Collaborative Learning (CL)

Distributed machine learning [3] refers to the techniques and algorithms for training a single model or architecture over a large-scale and/or sparse data sources using distributed systems that enable parallel computation, data distribution, and resilience to failures. Unlike distributed machine learning, which is fairly matured with countless variants and implementations in commercial and open-source frameworks, **federated machine learning** [4] is a relatively new research topic with a lot of open questions. Indeed, the term *federated learning* was introduced only 5 years ago [5]. This research area exists to solve a longstanding goal of large-scale learning left unaddressed by distributed learning: to analyze and learn from data distributed among many owners while respecting proprietary information and data privacy. The term ***collaborative learning*** has been used, on a few occasions, interchangeably with federated learning. However, this term is used here to emphasize the concurrent learning of multiple heterogenous models, which plays a key role in several significant improvements.

Consequently, the above use case stresses the need to develop collaborative learning concepts and frameworks that are universal, resilient, adaptive, robust, fault-tolerant, scalable, trust-worthy and privacy-preserving. Specifically, the following open challenges need to be addressed:

1. Develop, analyze, and validate a plug-and-play universal framework for very large-scale collaborative learning
2. Develop learning methods and new convergence analysis that use connectivity information to speed up convergence robustly and safely
3. Develop methods, analysis, and tools to minimize the data movement bottlenecks inherent with learning over large networks.
4. Develop privacy-preserving knowledge similarity measures to achieve knowledge consensus among diverse heterogenous models in the framework
5. Combine graph theory and operator splitting theory to develop efficient methods to distribute learning tasks over arbitrary networks, resiliently and robustly.
6. Develop graph-theoretic-based metrics to assess the resiliency of distributed learning algorithms
7. Develop methods to assess and quantify the level of privacy and confidentiality protection over a collaborative/federated learning system.

References

[1] Y. Zhang, G. Bai, X. Li, C. Curtis, C. Chen and R. K. Ko, "PrivColl: Practical Privacy-Preserving Collaborative Machine Learning," in *European Symposium on Research in Computer Security*, 2020.

[2] J. So, B. Guler and A. S. Avestimer, "A Scalable Approach for Privacy-Preserving Collaborative Machine Learning," *arXiv preprint arXiv:2011.01963,* 2020.

[3] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen and J. S. Rellermeyer, "A survey on distributed machine learning," *ACM Computing Surveys (CSUR),* vol. 53, no. 2, pp. 1-33, 2020.

[4] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konecný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu and S. Zhao, "Advances and Open Problems in Federated Learning," *Foundations and Trends® in Machine Learning,* vol. 14, no. 1, 2021.

5] B. H. McMahan, E. Moore, D. Ramage, S. Hampson and B. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *20th International Conference on Artificial Intelligence and Statistics (original version on arXiv Feb. 2016)*, 2017.

Federated Neural Architecture Search for Privacy-Preserving AI/ML

Prasanna Balaprakash[1*] (corresponding author), Krishnan Raghavan[1], and Roozbeh Yousefzadeh[2]
[1]Argonne National Laboratory and [2]Yale University
pbalapra@anl.gov

Topic: Federated AI/ML, differential privacy, and randomized algorithms

The field of automated machine learning (AutoML) offers new avenues for privacy-preserving AI/ML model development for various DOE applications. AutoML seeks to develop AI/ML models in an autonomic way without humans experts in the loop. Federated neural architecture search (NAS) [1] is a class of AutoML that tries to automate the design and development of neural network models from decentralized training data. However, the key algorithmic techniques for NAS, such as reinforcement learning (RL) and differentiable search (DS) strategies, lack privacy preservation capabilities required for new and emerging AI/ML-driven DOE applications. We envision the advancement of foundations and applicability of the federated NAS methods for privacy preservation through differential privacy, data encryption schemes, and game theory.

1. Privacy-preserving neural architecture search: We focus on the setting with multiple sites, where each site has its own data that cannot be shared. The federated RL-NAS approach follows a manager-worker distributed learning paradigm, wherein neither the dataset nor the weights/gradients of the neural network models leave the site. The manager has a parameter server, and each site has an RL agent with a policy network. In the beginning, each agent starts with the same policy network, generates a set of neural architectures, evaluates them (training and validation), and computes the gradient for RL policy network update. The parameter server receives the policy network gradient updates from the agents, combines them, and sends the result to the agents. In the DS-NAS approach, the parameter server maintains the state of architecture variables and weights of an overparameterized network. Each site has its own copy of the overparameterized network and calculates the gradient updates of architecture variables and weights based on its local data. The parameter server aggregates the gradients from different sites and sends the updated gradients back to each site. For both RL-NAS and DS-NAS, we can achieve differential privacy by leveraging the newly introduced, theoretically sound, mathematically tractable f-differential privacy [2], wherein Gaussian noise can be added to the gradients to achieve privacy without losing accuracy. For DS-NAS, we envision homomorphic encryption methods to allow untrusted sites to train or use the overparameterized model.

2. Shearlet system with cryptography: Federated learning methods that keep the data at the local sites have to rely on the computing power available at each of the sites. Doing so is not always feasible at all sites because expensive computations have to be performed at local sites that host the data, likely with less capable computing resources [3]. Arguably, the data can be moved from compute-limited sites to compute-enhanced sites, a process that can be enabled through ESnet and is expected beyond 5G wireless networks. However, this setting requires privacy preservation. To address these issues, we propose to develop a novel encryption framework that relies on shearlet systems [4]. Here, data is encrypted at local sites by using shearlet systems. The encrypted datasets are then transferred from the compute-limited sites to compute-enhanced sites. After assembling the data, federated NAS can be utilized to learn from all the data. The encryption process, performed at the local site level, requires a relatively inexpensive computation and therefore does not require large amounts of computing power. Shearlet systems are computationally complicated, and the resulting encrypted data can be just a set of complex-valued numbers. Such complex-valued

numbers would be hard to decrypt without the keys, yet easy to transfer from one site to another. The decryption module works with private and public keys, hidden inside a neural architecture. The output of the decryption module remains hidden inside the model while being passed through the neural network layers. Because shearlet transform is a reversible computational operation, we can reconstruct the original data from the encrypted version. Therefore, the trained model will be trained on the original data and not a modified version of it. This approach ensures that encryption does not adversely affect the generalization of model, while preserving the privacy of the data. To further ensure and quantify the privacy of the system, Shamir's Secret Sharing scheme [5] can be leveraged. In this scheme, if encryption keys at a certain number of sites are compromised, the data from other sites will still remain secure. Furthermore, the mathematical basis and theory must be developed to guide the design of encryption keys and to quantify the possibility of breaking the encryption keys.

3. Consensus-driven multiagent continual game: The key issue in federated learning with NAS is that one cannot verify whether data at each site is identical or not. This provides a unique challenge where gradients averaged across different sites (as is done typically [6]) cannot be directly used to update the parameter server. To that end, a theoretically sound gradient-consensus algorithm that provides privacy guarantees for NAS is necessary. We envision a dynamic gradient combination scheme where the gradients from different sites are combined to form a consensus (a combination of gradients). In order to guarantee that the consensus is optimal, a multiagent game should be designed such that each site acts as an agent with the goal of obtaining the contribution of each site to the consensus. The consensus should be capable of updating the network parameters and must allow the agents (different sites) to form a Nash equilibrium (the guarantee of equilibrium would allow for convergence of the learning). However, the design of a gradient combination scheme raises a new quandary. As each new gradient provides new information about the sites and the corresponding data, the problem is nonstationary, wherein the distribution of the gradients is time varying. Therefore, one must continually aggregate gradient information from multiple agents and repeatedly obtain consensus (to solve the multiagent game and find the Nash equilibrium). In order to facilitate this process, a continual learning method [7] must be developed to perform this aggregation while maintaining privacy guarantees and learning optimality. Moreover, these approaches need to model the distribution shift due to different data samples and the discrepancy due to information collected between different sites.

References

[1] Hangyu Zhu, Haoyu Zhang, and Yaochu Jin. From federated learning to federated neural architecture search: a survey. *Complex & Intelligent Systems*, 7(2):639–657, 2021.

[2] Qinqing Zheng, Shuxiao Chen, Qi Long, and Weijie Su. Federated f-differential privacy. In *International Conference on Artificial Intelligence and Statistics*, pages 2251–2259. PMLR, 2021.

[3] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2):1–19, 2019.

[4] Héctor Andrade-Loarca, Gitta Kutyniok, and Ozan Öktem. Shearlets as feature extractor for semantic edge detection: the model-based and data-driven realm. *Proceedings of the Royal Society A*, 476(2243):20190841, 2020.

[5] Mustafa Ulutas, Güzin Ulutas, and Vasif V Nabiyev. Medical image security and EPR hiding using Shamir's secret sharing scheme. *Journal of Systems and Software*, 84(3):341–353, 2011.

[6] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[7] R Krishnan and Prasanna Balaprakash. Meta continual learning via dynamic programming. *arXiv preprint arXiv:2008.02219*, 2020.

Title: *Can we triple dip? Research to provide database representations that simultaneously preserve meaning, lower dimensionality, and facilitate differential privacy guarantees.*
Authors: *Robert A. Bridges (bridgesra@ornl.gov), Mingyan Li (lim3@ornl.gov)*

We propose investigations of a seemingly fortuitous intersection of three ideas: (1) semantic preserving embeddings of databases that permit both (2) leveraging sparsity for an invertible & dimension-reduced representation and (3) employing randomness to provide differential privacy (DP) guarantees. If successful, the proposed result would be a generalizable method for low dimensional analytics in a metrizable topological vector space with a metric (distance function) that respects relationships in the data and that provides quantifiable risk with respect to privacy as defined by DP bounds. Research to evaluate the tradeoff in accuracy of the analytics, computational performance, and privacy (ε,δ-DP) is proposed. Two potential examples—both centered on sharing of analytics designed/trained on databases with privacy concerns, namely enterprise network data and healthcare data—are discussed.

Part 1 Metrizable Embeddings of Databases: Recent work is emerging with the goal of identifying embeddings of databases that preserve some meaning, that is, finding a mapping of records in the database to vectors in a metrizable (has a distance function) vector space so that mappings with pairwise small/large distance are images conceptually similar/different records E.g., authors of [1] use natural language processing (NLP) based algorithms to find representations of a vector space, while in [2] a metric space embedding is manually defined for any database whose fields are categorical, numerical, or textual. By design, the distances preserve semantics. The frameworks in these works have permitted worthwhile analytics in proof-of-concept applications, yet privacy concerns for sharing analytics or models trained on the data persist. Privacy concerns notwithstanding, we hypothesize that like the effect Word2Vec [10] and BERT [11] embeddings had on the NLP community—providing a portable mechanism to assist a wide variety of text-based analytics—these analogues for databases hold the promise of providing a generalizable method to catalyze research in a wide variety of applications.

Parts 2 & 3 RIP+DP - Invertible Lower Dimensional Representations that Admit DP: Emerging research is identifying mechanisms for randomly sampling or deterministically producing matrices that are simultaneously a dimension reduction mapping that satisfies the restrictive isometry property (RIP)—this means when restricted to sparse vectors, the mapping smashes vectors into a lower dimensional space but is still invertible—and provides DP proofs [5, 12]. Tradeoffs are being explored in these works between the computational, accuracy, and privacy implications of using random projections to enable RIP and DP (analytics happening in the low dimension space), or providing noise generation in a low dimension representation, then inverting (to the high dimension, original space) before analytics.

Regardless, there are prerequisites that cause two outstanding hurdles for marrying a database embedding with RIP+DP embeddings. Problem (1): invertibility of the dimension reduction relies on sparseness of the original vectors. As for sparseness of the database representation, we note that previous research has found it an irritating consequence of real-world data [2]. Sparsity, while necessary mathematically for RIP may not be a problematic limitation in many real-world applications. Problem (2) is that the "database" is assumed to live in $(\boldsymbol{R}^n, ||\bullet||)$, i.e., with the usual norm, when RIP is considered. As RIP has been shown valid for the case of a topological vector space [6], we hypothesize that semantic embeddings of databases (to $\boldsymbol{R}^n$ endowed with a metric dependent on the database) admit RIP+DP. Research to prove when these conditions hold is proposed.

Overall, we hypothesize that metrizable vector space embeddings of databases, if sufficiently sparse, can be leveraged to enable respect semantics of the original data and will permit lower dimensional mappings while enabling DP mechanisms. The consequence of this result would be far reaching if true—

effectively for databases in which each row is sufficiently sparse the result would allow computation in a much lower dimensional space and subsequent recovery to the original high dimensional version while provably preserving privacy. Research is required to prove the DP guarantees, and then to quantify the tradeoff between the analytical accuracy, computational expense, and privacy guarantees. Once basic research can be established, prototypical implementations can verify the benefits on real world data.

Healthcare: Many previous research efforts have shown that data mining of massive healthcare databases can yield novel insights aiding diagnosis of disease, effective treatment, or needed interventions [7, 8]. However, for healthcare data, individual privacy concern of their health records is very important. In order to glean the benefits of big healthcare data in the advancement of medical science and the improvement of citizen's wellbeing, the data privacy concerns must be addressed [9]. Ideally, whether hand-crafted or data-driven, an embedding of records (or better a patient's records over time) can be found that preserves similarity in the health-domain (e.g., similar but different treatments or conditions will land close in the embedding). Such a representation promises to enable any analytics on the data, and further, assuming positive results of the proposed research above, lower dimensionality and preserve privacy. Notably, if time-varying data is considered, more privacy concerns are entailed [4] as correlation over time can leak information.

Network Security: For cybersecurity, currently, automated attack detection and prevention often begins by crafting indicators of compromise (IOCs) from attacks discovered through forensic investigations. These IOCs are brittle rules crafted to identify specific, known attacks, but are shared with no privacy concerns (as they contain no information of user or the enterprise's network) and are designed from real attacks. Modern commercial vendors claim to emulate advanced attacks in a simulated environments in order to generate training data for machine-learning-based detectors, with the promise of identifying never-before-seen attacks via less-brittle, behavior-based features. The metric space approach of [2] married with DP provides a potentially new avenue for creating detectors that realizes the benefits of both methods: the metric provides distances that can indicate similar but only slightly different patterns (softening the brittle rules), the DP may permit *shareability* of detectors allowing more general IOCs from *real network attack data.*

[1] Rajesh Bordawekar and Oded Shmueli. Using Word Embedding DEEM'17.ACM. 2017

[2] Miki Verma and Robert A. Bridges. Defining a metric space … (Big Data), 2018

[3] Pauwels & Gerchinovitz. An Introduction to differential privacy, 2016

[4] Cao, Yang, et al. Quantifying differential privacy under temporal correlations. ICDE, 2017

[5] Upadhyay J. Randomness efficient …, arXiv:1410.2470. 2014

[6] Hertrich, Johannes, et al. The basins of attraction ..., 2020

[7] Erik Roeloft et al. Benefits of a clinical data…, Radiotherapy and Oncology 2013

[8] Sarwar Kamal et al. Large scale medical data …, 2017

[9] Karim Abouelmehdi et al. Big Healthcare data: preserving security and privacy. J. of Big Data, 2018

[10] Mikolov, Tomas, et al. Efficient estimation …, arXiv:1301.3781, 2013

[11] J. Devlin, M.-W. et al. BERT: Pre-training ACL, 2019, DOI: 10.18653/v1/N19-1423

[12] Li, Yang D., et al. Compressive mechanism: Utilizing …, ACM 2011

Securing Next-Generation Interfaces to DOE Scientific Computing Resources

Shane Canon[1], Shreyas Cholia[2], Damian Hazen[1], Rollin Thomas[1]

1. National Energy Research Scientific Computing Center, Lawrence Berkeley National Laboratory.

2. Scientific Data Division, Lawrence Berkeley National Laboratory.

Scientific computing is becoming increasingly collaborative, and is often distributed across large, geographically diverse teams. The COVID-19 pandemic has accelerated the need to enable remote modes of collaboration among teams. For instance, scientists across the globe may want to work together on a common light-source data analysis run, using interfaces and services where they can collectively run and manage their workflows. Platforms like the DOE KBase Project provide a web-based platform that enables users to perform advanced analysis of biological data, share and publish those results.

To achieve this, scientists are relying on rich interfaces that enable sophisticated analysis tools and expose high-performance computing (HPC) and data resources to users over the web using science gateways or services like Jupyter [1] or Open OnDemand [2]. These have enormous potential to make using HPC easier, more accessible, and more productive for more users. Automated, agent-driven, service-oriented models of interacting with HPC [3] leveraging REST APIs and other web protocols are also becoming increasingly important, especially as experimental and observational science facilities turn to HPC to handle their data management and analytics problems. The days when users only used Secure Shell (ssh) to interact with a remote HPC system are coming to a close; these new tools hold new promise but pose new questions and challenges that can only be addressed through basic cybersecurity research and innovation.

Over several decades, HPC facilities have developed security infrastructure to help monitor and protect HPC assets [4]. These approaches have included high-performance real-time intrusion detection, the DMZ model for high-performance data transfers, instrumented services, and other best practices [5]. These systems have been further hardened through multi-factor authentication and carefully engineered firewalls that protect but still deliver high performance. These tools were designed and optimized for an access model dominated by Secure Shell access where the primary threat was stolen credentials and, occasionally, vulnerable services and detection focused on detecting well-defined signatures or anomalous behavior. Many of these tools are layered on top of the UNIX/POSIX model for managing resources which is limiting and may not be able to capture the rich levels of roles, access, and permission needed to orchestrate interactions across different users and teams.

Distributed, team-based science is increasingly dependent on an ever-more complex and sophisticated set of services. For example, scientists are using web-based platforms like Jupyter to conduct analysis and share results. A network of web services and other science portals have also become critical to conducting science. The security methods of the past are not aligned with the needs of applications that are built in the web-API mode of shared interactions. To best protect these services while still allowing legitimate users to unlock their full potential, an improved security infrastructure is required. These expanded capabilities include improved monitoring and anomaly detection methods for services running over encrypted channels (e.g. SSL/https) and potentially using new protocols such as websockets. In addition to network based security models, improved monitoring tools running within large systems that can monitor at scale with low overhead are required. Given the pace of change, more

sophisticated anomaly detection methods that move well beyond signature-based or tripwire based models are required. Ideally these new methods could leverage emerging AI-based methods to recognize potential attacks or misuse.

In addition, the security models need to adapt to address workflows that often involve multiple parties each requiring access to specific resources and a framework that helps define and enforce policies. Consider a workflow that uses a shared JupyterLab instance, with a subset of users able to execute commands, another subset able to annotate the workflow, and a separate set of users that require read-only access to generated data products. Addressing this type of use case requires a common understanding of the needs of web-based science workflows that manage access to a diverse set of resources, and how we might facilitate fine-grained access and collaboration across classes of users. There has been significant prior work in the cloud computing space with respect to Identity and Access Management (IAM). In this model, access to resources is managed through policy engines, which can provide a very high degree of flexibility and fine-grained control. This also decouples resources on the backend from having to implement these access policies - they are brokered through the web API and the policy engine. Federated Authentication[6] can provide a common layer for managing user identities, but these need to be then integrated with access management. Seeing how these approaches might apply to DOE computing resources will be critical in ensuring that this infrastructure is useful and aligned with the priorities of modern team science, which is highly distributed and collaborative.

DOE scientists are adopting more contextual, interactive, and collaborative tools in order to meet the mission needs of DOE-SC. These new tools enable sophisticated analysis and tightly integrate with HPC resources to harness the computational capabilities they provide. These tools, based on popular and pervasive web technologies, carry with them new challenges and risks from the security perspective. We suggest that DOE has a role to play in funding basic cybersecurity research into new models that take these new modes of access into account, adapt and extend existing security tools to monitor them, promote a common framework that can integrate into existing web tooling and provides a reference model, and engage with developers on security issues of mutual interest to build a community that is jointly addressing these challenges.

[1] R. Thomas and S. Cholia, “Interactive Supercomputing With Jupyter,” *Computing in Science Engineering*, vol. 23, no. 2, pp. 93–98, Mar. 2021.

[2] D. Hudak *et al.*, “Open OnDemand: A web-based client portal for HPC centers,” *J. Open Source Softw.*, vol. 3, no. 25, p. 622, May 2018.

[3] S. Cholia, D. Skinner, and J. Boverhof, “NEWT: A RESTful service for building High Performance Computing web applications,” in *2010 Gateway Computing Environments Workshop (GCE)*, Nov. 2010, pp. 1–11.

[4] V. Paxson, S. Campbell, J. Lee, and Others, “Bro intrusion detection system,” Lawrence Berkeley National Laboratory, 2006. [Online]. Available: https://www.osti.gov/biblio/1245188

[5] E. Dart, L. Rotman, B. Tierney, M. Hester, and J. Zurawski, “The Science DMZ,” presented at the SC13: International Conference for High Performance Computing, Networking, Storage and Analysis, Denver Colorado, Nov. 2013. doi: 10.1145/2503210.2503245.

[6] J. Peila, A. Ghilarducci, and LAWRENCE LIVERMORE NATIONAL LAB CA, “OneID accurate Identity: The next security perimeter,” LAWRENCE LIVERMORE NATIONAL LAB CA, Mar. 2021. Accessed: Oct. 15, 2021. [Online]. Available: https://apps.dtic.mil/sti/citations/AD1126603

Toward Trustworthy Accelerated Federated Learning

Franck Cappello (ANL), Sheng Di (ANL), Kibaek Kim (ANL), Ravi Madduri (ANL),
Jon Calhoun (Clemson), Dingwen Tao (WSU).
Contact: cappello@anl.gov

Goal: Our goal is to study and develop techniques to improve trust and accelerate federated learning. In this white paper, trust is bidirectional: protect the privacy of information and provide a guarantee of correctness and protection against attacks.

Context: Federated learning is becoming an important support for data science in the context of the convergence between AI, HPC, and edge computing [1]. It is considered as one of the major learning methods to mitigate systemic privacy risks and costs resulting from traditional, centralized machine learning and data science approaches [2]. Federated learning has DOE relevant applications in all situations where data in distributed training cannot be shared: in a cross-silo context for a handful of large data clients (typically facilities) and in a cross-device context for a large number of small devices (typically sensors). Examples of application domains are: Smart Grid [3,4,8], engineering and manufacturing [1], federated instrumentation [1], Bridge2AI [5].

Current techniques for federated learning: In centralized federated learning, edge devices connect to a trusted central server to contribute to the learning process. In decentralized federated learning, the edge devices communicate directly between them in a peer-to-peer mode. Each edge device runs a local machine learning algorithm to update the global model. However, because of its distributed aspect and its connectivity (5G, WIFI), federated learning is vulnerable to attacks. The current approaches to counter or avoid these malicious adversaries include secure multi-party computation (e.g., homomorphic encryption), secure enclaves, and differential privacy. The data communicated among federated training agents are the results of the local training. When training involves gradient descent, the data are a vast amount of gradients in floating-point format. Because of the limited communication capacity of edge devices or to avoid congestion on the centralized server, the gradients are also compressed. Several simple lossy compression methods have been explored to compress gradients: e.g. quantization, sampling, sparsification. In some situations, it is also necessary to compress the communications from server to clients.

Gap analysis: *Compression:* The scientific community has developed a profound understanding of compression methods for floating-point numbers and has developed advanced lossy compressors for this type of data. Compared to all other lossy compression techniques, these sophisticated compressors have demonstrated superior data reduction and better information preservation. They often exceed the compression ratios of simple techniques by 1 order-of-magnitude or more for comparable data preservation. Because higher compression ratios directly translate into lower communication times[1], these sophisticated compressors have the potential to accelerate federated learning.
Privacy/security: full encryption of the gradients that could account for terabytes of data for large scale training applications is computationally and energetically expensive. Moreover, porting the secure encryption protocols to the cross-device setting (e.g., large number of edge devices) may not be viable. Differential privacy has been recently proposed as a faster and less energy-consuming data protection method in federated learning. This technique adds noise (e.g., Gaussian and Laplacian noises) to either model outputs (e.g., gradients), inputs, or loss function. However, achieving

[1] depending on message size and communication medium bandwidth

differential privacy without losing the training/learning performance can be difficult.
Combination of Compression with Privacy/Security: Compressors can also provide support for privacy/security through the use of a codebook and quantization. For example, Huffman tree mutation and chaotic Huffman tree have been proposed in the context of joint compression and encryption (JCAE) schemes to obfuscate the Huffman codebook [6]. Vector quantization, which is a common technique used in lossy compression, has recently been proposed to anonymize the data [7]. However, state-of-the-art differential privacy algorithms are not designed to work with compressed or quantized communications. The combination of encryption, differential privacy, lossy compression and potentially homomorphic schemes (encryption, compression) opens new opportunities to investigate faster privacy/security methods.

Research directions: The application, adaptation, and optimization of these advanced lossy compression algorithms to federated learning for scientific applications has not been explored. For example, it is unknown how their error characteristics (distribution, autocorrelation, bias, etc.) are tolerated by different machine learning models. Moreover, the efficient combination of security/privacy techniques (homomorphic encryption, differential privacy) with advanced lossy compression for floating-point data is an open question. More specifically, the investigation of the following research opportunities will significantly improve our understanding and enable the development of trustworthy accelerated federated learning for DOE applications:

- Sensitivity analysis of the global model training convergence to the nature of the noise created by advanced lossy compression,
- Novel advanced lossy compression algorithms improving global model training convergence,
- Combination of compression and encryption and the trade-off between scalability and performance (communication, compression), resource usage (power, memory footprint, computation overhead), and security/privacy,
- Convergence analysis for training models when integrating lossy compression and differential privacy,
- Combination of homomorphic encryption and lossy compression,
- Innovative combinations of compression, partial encryption, and differential privacy to accelerate model training convergence and improve trust in federated learning.

References

[1] R. Stevens, V. Taylor, J. Nichols, A.B. Maccabe, K. Yelick, D. Brown, AI for Science. ANL technical report: ANL-20/17 158802, 2020. Web: https://doi.org/10.2172/1604756
[2] National Security Commission on Artificial Intelligence Final Report, https://www.nscai.gov/wp-content/uploads/2021/03/Full-Report-Digital-1.pdf
[3] Y. L. Tun, K. Thar, C. M. Thwal and C. S. Hong, "Federated Learning based Energy Demand Prediction with Clustered Aggregation," 2021 IEEE International Conference on Big Data and Smart Computing (BigComp), 2021, pp. 164-167, doi: 10.1109/BigComp51126.2021.00039.
[4] N. Gholizadeh, P. Musilek, Distributed Learning Applications in Power Systems: A Review of Methods, Gaps, and Challenges. MDPI Energies 2021, 14, 3654.
[5] U.S. Department of Energy, “Bridge2AI and Privacy-Preserving Artificial Intelligence Research, DOE National Laboratory Program Announcement Number: LAB 21-2502, 2021
[6] H. Hermassi, R. Rhouma, S. Belghith, Joint compression and encryption using chaotically mutated Huffman trees, Communications in Nonlinear Science and Numerical Simulation, Volume 15, Issue 10, 2010, Pages 2987-2999.
[7] Y. Miche et al.,Data Anonymization as a Vector Quantization Problem: Control Over Privacy for Health Data. Lecture Notes in Computer Science, vol 9817. Springer, 2016
[8] Ryu, Minseok, and Kibaek Kim. "A privacy-preserving distributed control of optimal power flow." IEEE Transactions on Power Systems (accepted), 2021.

Productionizing Data Science Oriented Intrusion Detection

Abstract

The basic premise of system call based anomaly detection has been well researched over the past 25 years[1], and advances in open source software, the Linux kernel[2] and machine learning[3] have resulted in open source tools that provide an Off the Shelf ML based anomaly detection using real time system call activity[4]. With the underlying science and tooling already in place, the time is ripe to deploy these tools within DOE computing facilities and provide them with ML training sets tuned specifically for project accounts. Cluster computing activity for project accounts usually follow standard, project specific workflows; these established workflows make it much easier to establish a targeted profile of normal behavior, and flag anomalies. By creating, documenting and packaging an anomaly detection stack that integrates into DOE computing environments and creating a process for generating and sharing training datasets that match system call patterns of DOE science programs, we can accelerate the adoption of these new technologies across the DOE laboratory complex. A major benefit of this stack would be that it can also be trained as both signature and anomaly based intrusion detection. By focusing on real time syscall analysis, detection occurs at the kernel level and is dramatically harder to evade. ML based detectors are also easier to train and more accurate[12]. A longer term benefit is that system call based intrusion detection is an ongoing research topic that is benefiting from the burgeoning data science movement. A common platform in DOE production environments would promote new approaches, provide a constant stream of raw data for analysis and accelerate techniques from laboratory to production. This would accelerate the pace of development in host based anomaly detection in the constant race to stay ahead of cybersecurity adversaries.

Discussion

Anomaly detection systems look for system behavior that departs from established patterns - as such they do not look for a particular attack, but look for anything that departs from a baseline profile. For example, an account that typically runs code related to coronavirus research[5] would trigger an alert if it started to run programs with system call patterns matching cryptocurrency mining[6][7]. An account that normally annotated genome data would trigger an alert if it began connecting to other hosts on the network and trying different login credentials. Accounts that normally only read/write to project directories that change their behavior to read and write to directories containing secure credentials or access control lists would trigger alarms. Rather than look for specific attacks, anomaly detection looks for abnormal behavior - enabling such a system to generate alerts for zero day attacks.

Researchers have shown that ML based approaches are capable of very high levels of accuracy for detecting anomalous behavior when properly tuned and given the proper training

Stephen Chan
sychan@lbl.gov

data[8]. The current tools for observing system call behavior can operate with tiny impacts on system performance, and return “live” information of what system calls are being made, as they are being made. This allows real time anomaly detection for batch jobs that are specific to each collaboration. In addition, categorizers for the syscall signatures of known exploits can be created and applied to real time syscall analysis of interactive nodes, batch nodes and Jupyter Notebook services to activate IP address blocking and process termination defenses. These defenses offer the possibility of stopping a system compromise before it is able to complete, or before the compromise can be exploited to harvest credentials or compromise other machines within the cluster.

In the past, these kinds of activities could only be discovered during the forensics phase of a security incident - which may occur hours, days or in some cases months after a system compromise. In some instances, compromises were only detected many months after the attacker had already gained supervisor privileges on a system and installed multiple backdoors, sometimes resulting in expensive and embarrassing shutdowns of high profile supercomputer clusters such as at SDSC in 2004 and NERSC’s Seaborg in 2006. After the deployment of Instrumented SSH[9] at NERSC, the success of attacks of the type that took Seaborg offline in 2006 dramatically declined due to the new signature based intrusion detection capabilities. Arguably a tool such as Instrumented SSH or similar terminal loggers may have prevented or at least mitigated the recent European Supercomputer compromises: Instrumented SSH demonstrated that identifying attackers in real time blocked many attacks before they could progress to more expensive and damaging stages. However, Instrumented SSH requires deep and brittle modifications to source code, making it hard to maintain. Furthermore, the entire class of terminal monitoring tools is blinkered to only traffic displayed or typed in a terminal session; activity that is embedded in a program is invisible to terminal loggers, rendering them ineffective. As contemporary interactive sessions move away from shell terminals, the usefulness of tools like Instrumented SSH also rapidly declines. An approach based on syscall analysis removes these blinkers and enables analysis of all activity on the system, whether it is interactive, batch, or buried deep in otherwise benign code because syscall analysis operates at the kernel level.

Due to this deep inspection, syscall tracing approaches can also secure interactive Jupyter and R-Studio notebooks used for ad hoc analysis: rather than look for departures from normal, baseline activity, a signature based approach using training sets identifying risky and/or malicious system call patterns, can be applied. The same infrastructure can be used to look for attack signatures rather than anomalies. As interactive tools like Jupyter become more popular within DOE, they become a vast new attack surface which is relatively undefended - addressing this problem is vital to maintaining the security of DOE computing environments.

Data science based approaches have become popular in recent host based intrusion detection research at roughly the same time that fast, easy syscall monitoring has become widely available in mainstream Linux kernels. Cybersecurity teams typically have far more data than they are capable of analyzing using previous techniques - the addition of voluminous syscall logs from production compute clusters would only worsen the data overload. Data science approaches are ideal for rapidly processing large amounts of data and have been demonstrated to be more effective than previous syscall analysis approaches, with much higher detection rates (99% vs 79%)[10] and simpler detectors. This intersection has enabled research

proof of concepts such as Hidden Markov Model based anomaly detectors operating in-kernel to detect syscall anomalies[11]. This intersection also presents an opportunity for data science approaches to make an impact on *production* intrusion detection methods at a time when existing cybersecurity approaches have faced unprecedented challenges. What is needed are not narrow, brittle, bespoke solutions, but collaborative and extensible platforms based on community supported open source tools, that make syscall data available to data scientists who can then collaborate with systems professionals to implement and test new approaches for intrusion detection. The goal is not to promote a single approach, but to create a substrate that enables different approaches to be tested, tuned and disseminated rapidly. So that methods can move from conference paper or journal article to active, operational defense on a timescale measured in months, not years or decades.

[1] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, “A sense of self for Unix processes,” in Proceedings 1996 IEEE Symposium on Security and Privacy. IEEE, 1996, pp. 120–128. url: http://wenke.gtisc.gatech.edu/ids-readings/unix_process_self.pdf
[2] Brendan Gregg. 2019. BPF Performance Tools: Linux System and Application Observability (1st. ed.). Addison-Wesley Professional. url: https://www.brendangregg.com/ebpf.html
[3] Y. Xin et al., "Machine Learning and Deep Learning Methods for Cybersecurity," in IEEE Access, vol. 6, pp. 35365-35381, 2018, doi: 10.1109/ACCESS.2018.2836950.
[4] Sysdig. “Anomaly Detection Alerts.” Sysdig Documentation, url: https://docs.sysdig.com/en/docs/sysdig-monitor/alerts/anomaly-detection-alerts/
[5] Corfield, Gareth. “Danger Zone! Brit Research Supercomputer Archer's Login Nodes Exploited in Cyber-Attack, Admins Reset Passwords and SSH Keys.” The Register® - Biting the Hand That Feeds IT, The Register, 14 May 2020, url: https://www.theregister.com/2020/05/13/uk_archer_supercomputer_cyberattack/
[6] Moss, Sebastian. “European Supercomputers Hacked, Apparently to Mine Cryptocurrency.” *Data Center Dynamics*, Data Centre Dynamics Ltd, 18 May 2020, url: https://www.datacenterdynamics.com/en/news/european-supercomputers-hacked-mine-cryptocurrency/
[7] R. Karn, P. Kudva, H. Huang, S. Suneja and I. Elfadel, "Cryptomining Detection in Container Clouds Using System Calls and Explainable Machine Learning" in IEEE Transactions on Parallel & Distributed Systems, vol. 32, no. 03, pp. 674-691, 2021.
doi: 10.1109/TPDS.2020.3029088
keywords: {containers;cloud computing;malware;machine learning;cryptocurrency;data mining}
url: https://doi.ieeecomputersociety.org/10.1109/TPDS.2020.3029088
[8] O. Tunde-Onadele, J. He, T. Dai and X. Gu, "A Study on Container Vulnerability Exploit Detection," 2019 IEEE International Conference on Cloud Engineering (IC2E), 2019, pp. 121-127, doi: 10.1109/IC2E.2019.00026. url: http://dance.csc.ncsu.edu/papers/IC2E19.pdf
https://www.computer.org/csdl/proceedings-article/wiw/2016/6039a104/12OmNBIXs30
https://www.sciencedirect.com/science/article/pii/S1742287618300392?via%3Dihub
https://ieeexplore.ieee.org/document/932213
[9] Campbell, Scott, & Campbell, Scott. “Instrumented SSH,” 27 May 2009, Lawrence Berkeley National Lab white paper. url: https://doi.org/10.2172/960441

Stephen Chan
sychan@lbl.gov

[10]Lee, B.A., Amaresh, S., Green, C., & Engels, D.W. (2018). Comparative Study of Deep Learning Models for Network Intrusion Detection. Url: https://scholar.smu.edu/cgi/viewcontent.cgi?article=1009&context=datasciencereview
[11] J. Byrnes, T. Hoang, N. N. Mehta and Y. Cheng, "A Modern Implementation of System Call Sequence Based Host-based Intrusion Detection Systems," 2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), 2020, pp. 218-225, doi: 10.1109/TPS-ISA50397.2020.00037. Url: https://ieeexplore.ieee.org/document/9325401
[12] Pan, Y., Sun, F., Teng, Z. et al. Detecting web attacks with end-to-end deep learning. J Internet Serv Appl 10, 16 (2019). https://doi.org/10.1186/s13174-019-0115-x url: https://doi.org/10.1186/s13174-019-0115-x
[13] “SDSC Statement on Recent Widespread Cyber Attacks,” 9 April 2004, San Diego Super Computer Center website sdsc.edu, url https://www.sdsc.edu/News%20Items/PR040904.html

Stephen Chan
sychan@lbl.gov

Benchmarking Federated Learning in a Principled Fashion

Mosharaf Chowdhury (mosharaf@umich.edu)
SymbioticLab, University of Michigan

Abstract The biggest challenge federated learning (FL) faces is not a lack of algorithms but having too many without any means to separate chaff from wheat. This is a call to action to curate realistic large-scale FL benchmarking datasets and to build a practical FL system to efficiently perform apples to apples comparison.

1 Introduction

Federated learning (FL) is a machine learning (ML) paradigm where a logically centralized coordinator orchestrates many distributed clients to collaboratively train a model [4, 8]. In the presence of client heterogeneity, existing efforts have focused on optimizing the FL: (1) *System efficiency*: reducing computation load (e.g., using smaller models [10]) or communication traffic (e.g., local SGD [9]) for faster execution; (2) *Statistical efficiency*: designing data heterogeneity-aware algorithms (e.g., client clustering [6]) to obtain better training accuracy with fewer training rounds; (3) *Privacy and security*: developing reliable strategies (e.g., differentially private training [7]) to make FL more privacy-preserving and robust to potential attacks.

The performance of an FL solution greatly depends on the characteristics of data, device capabilities, and participation of clients. Existing benchmarks for FL fall short in multiple aspects: (1) they are limited in the versatility of data for various real-world FL applications. Instead, their datasets often contain synthetically generated partitions derived from conventional datasets and do not represent realistic characteristics (e.g., LEAF [5]); (2) they often overlook different aspects of practical FL. For example, system speed and availability of the client are largely missing (e.g., FedML [3]), which discourages efforts from considering FL system efficiency and resilience, and leads to overly optimistic statistical performance; (3) their experimental environments are unable to reproduce the practical scale of FL deployments, which again can under-report the realistic FL performance.

For the last 18 months, we have been working on FedScale to enable comprehensive FL benchmarking.[1] FedScale currently has 18 realistic FL datasets spanning across different scales for a wide variety of FL tasks (Table 1). In addition, we are building an automated evaluation platform, FedScale Automated Runtime (FAR), to simplify and standardize a more realistic FL evaluation. FAR integrates real-world traces to simulate the realistic behaviors of an FL deployment and can identify practical FL concerns. It can perform the training of thousands of clients in each round using only a few GPUs. We are also creating software backends for FL training on Android devices, browsers etc.

Category	Name	Data Type	#Clients	#Instances
CV	`OpenImage`	Image	13,771	1.3M
	`Charades`	Video	266	10K
	`VLOG`	Video	4,900	9.6K
	`Waymo Motion`	Video	496,358	32.5M
NLP	`Europarl`	Text	27,835	1.2M
	`Reddit`	Text	1,660,820	351M
	`LibriTTS`	Text	2,456	37K
	`Common Voice`	Audio	12,976	1.1M
Misc ML	`Taobao`	Text	182,806	20.9M
	`Fox Go`	Text	150,333	4.9M

Table 1: *Statistics of* partial *FedScale datasets. FedScale has 18 real-world federated datasets.*

2 FedScale: DataSet and Evaluation Platform

FL performance relies on at least three aspects: (1) *Client statistical data*: it determines the statistical efficiency of FL tasks (e.g., convergence and model accuracy); (2) *Client system behavior*: the compute/communication speed of client devices and temporal availability determine the efficiency of FL tasks (e.g., drift of data distribution); and (3) *Task categories*: model and application combinations can exhibit different reliance on client data and execute at different system speeds. Hence, an ideal FL benchmark should cover all three aspects.

2.1 Realistic Workloads for Federated Learning

Client Statistical Dataset FedScale currently has 18 realistic FL datasets (Table 1) for a wide variety of task categories, such as image classification, object detection, language modeling, speech recognition, machine translation, and reinforcement learning. Meanwhile, these datasets cover different scales, from hundreds

[1] FedScale is available at https://github.com/SymbioticLab/FedScale.

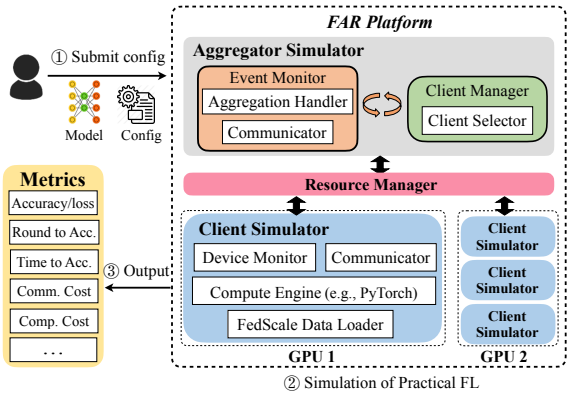


Figure 1: ***FAR enables the developer to benchmark various FL efforts with practical FL data and metrics.***

to millions of clients, to accommodate diverse FL scenarios. The raw data of these datasets are collected from different sources in various formats. We clean up the raw data, partition them into new FL datasets using their real client-data mapping, and streamline new datasets into consistent formats. e.g., we use the `AuthorProfileUrl` attribute of the `OpenImage` data to map data instances to each client. We notice these realistic datasets are indeed have client heterogeneity.

Client System Behavior Trace We formulate the system trace of different clients using *AI Benchmark* [1] and *MobiPerf Measurements* [2] on mobiles. *AI Benchmark* provides the training and inference speed of diverse models (e.g., MobileNet) across a wide range of device models (e.g., Samsung Galaxy S20), while *MobiPerf* has collected the available cloud-to-edge network throughput of over 100k world-wide mobile clients. As specified in real FL deployments [4, 12], we focus on mobile devices that have larger than 2GB RAM and connect with WiFi. To account for the dynamics of client availability, we clean up a large-scale user behavior dataset spanning 136k users [11] to emulate the behaviors of clients, which includes 180 million trace items of client devices (e.g., battery charge or screen lock) over a week. So we can evaluate the resilience of FL optimizations under client dynamics.

2.2 FAR: FL Evaluation Platform

Existing FL evaluation platforms can hardly reproduce the scale of practical FL deployments and fall short in providing user-friendly APIs, which requires great developer efforts to deploy new plugins. As such, we introduce FedScale Automated Runtime (FAR), an automated and easily-deployable evaluation platform, to simplify and standardize the FL evaluation under a practical setting. As shown in Figure 1, the resource manager orchestrates the available physical resource for evaluation to maximize the resource efficiency (e.g., queuing and balancing client events across machines), and FAR components will simulate real FL runtime using realistic client trace. For example, the communicator will record the simulated client communication time ($\frac{network\_traffic\_size}{client\_bandwidth\_trace}$); the device monitor will simulate the client dynamics (e.g., clients rejoin or fail); and participants are running on real heterogeneous federated dataset. So it can provide various practical FL metrics, such as computation/communication cost, latency and wall clock time.

References

[1] AI Benchmark: All About Deep Learning on Smartphones. http://ai-benchmark.com/ranking_deeplearning_detailed.html.

[2] MobiPerf. https://www.measurementlab.net/tests/mobiperf/.

[3] PySyft. https://github.com/OpenMined/PySyft.

[4] Keith Bonawitz, Hubert Eichner, and et al. Towards federated learning at scale: System design. In *MLSys*, 2019.

[5] Sebastian Caldas, Sai Meher, Karthik Duddu, and et al. Leaf: A benchmark for federated settings. *NeurIPS' Workshop*, 2019.

[6] Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

[7] Peter Kairouz, Brendan McMahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. In *arxiv.org/abs/2103.00039*, 2021.

[8] Peter Kairouz, H. Brendan McMahan, and et al. Advances and open problems in federated learning. In *Foundations and Trends® in Machine Learning*, 2021.

[9] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017.

[10] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

[11] Chengxu Yang, Qipeng Wang, and et al. Characterizing impacts of heterogeneity in federated learning upon large-scale smartphone data. In *WWW*, 2021.

[12] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. Applied federated learning: Improving Google keyboard query suggestions. In *arxiv.org/abs/1812.02903*, 2018.

Towards Secure Heterogeneous Computing Systems for Distributed Science

Joaquin Chung*, Raj Kettimuthu, Nate Evans, Pete Beckman, Ian Foster

Argonne National Laboratory, USA

Abstract — Scientific computing ecosystems are constantly improving. The most recent trends propose self-driving labs, where artificial intelligence (AI) and robots will handle many different tasks. Furthermore, advanced wireless technologies (e.g., 5G and WiFi-6) will enable new modalities of distributed sensors that take advantage of a digital continuum of computing and storage resources, all interconnected by high-speed networks. This transformation will allow scientists to tackle currently intractable problems, but will involve new ecosystems with heterogeneous computing technologies that may open new avenues for cyber attacks. Moreover, these heterogeneous systems will likely reside in independent security domains, from the Department of Energy computing complex to the public Cloud to Edge computing sitting next to cell towers. To get ahead of the growing cybersecurity threat, we need to begin studying the challenges that will emerge when these heterogeneous and complex systems are fully integrated into distributed science ecosystems and used to support autonomous science.

1 Heterogeneous Computing Systems for Distributed Science

As improvements in central processing units (CPUs) slow, we are witnessing a transition to specialized processors (e.g., graphics processing units: GPUs, and AI accelerators). For instance, new exascale computers are designed to provide users with access to powerful GPUs. Furthermore, a digital continuum of computing and storage resources is becoming available from device/sensor to edge and from edge to HPC. This continuum is further expanded when advanced wireless technologies such as 5G are used to connect a multitude of sensors to remote HPC clusters [1]. For instance, the Wildebeest project [2] will support advanced AI that can "migrate" across the digital continuum to allow scientists to optimize sensors and instruments automatically in the field to report the most scientifically valuable data in real-time. Moreover, in-network computing techniques enable the offloading of tasks that are traditionally run on compute nodes to networking devices typically used for traffic forwarding (for example, we have demonstrated how to leverage in-network computing for performing scientific operations not supported in network devices using approximations in a streaming fashion [3]). As the diversity of heterogeneous resources increases, so too does system complexity. For this reason, researchers look for ways to abstract this complexity, with function as a service (FaaS) being the most recent instantiation. For example, funcX [4] is a platform that enables flexible, scalable, and high performance remote function execution over existing clouds, clusters, and supercomputers.

2 Science Drivers

To illustrate how heterogeneous computing systems can be used to conduct distributed science, we present two network-based application use cases that could be deployed in future infrastructure.

Sea Level Measurement: Climate change has made flooding events in coastal areas more frequent and damaging. Deploying water level sensors in flood prone areas will help emergency-response teams to plan and respond during these events by providing access to real-time sea level measurements. Typically, an IoT-based sea level measurement system is composed of wireless water level sensors that send data to an IoT gateway, which aggregates data from sensors and forwards (through either wireless or wired links) the aggregated data to an analysis (and storage) service in the Edge or Cloud. Because most critical functions in this architecture reside on the wireless part of the network, the system can suffer from network interruptions during heavy rain events that cause flooding. Some systems can rely on local storage to store measurements and release them when the network connectivity is available. They can also rely on historical data to predict sea levels on a short time span. However, these solutions sacrifice the real-time nature of the system. Moreover, an adversary could hijack the wireless last-mile with powerful radio signal or poison the data with malicious sensors. We require cybersecurity mechanisms that maintain the real-time nature of critical functions during disruptions and protect data and the measurement infrastructure from adversaries.

Self-Driving Laboratories: Advances in machine learning and artificial intelligence lead scientists to envision self-driving laboratories that autonomously design, perform, and interpret experiments. Such autonomous laboratories may integrate many different devices, including modular robotics [5], with a need

*chungmiranda@anl.gov

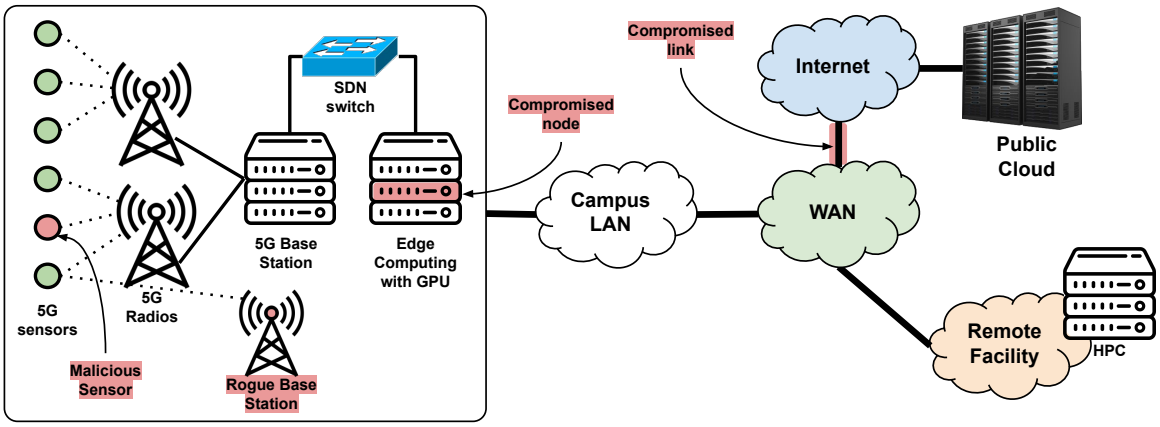


Figure 1: Attack vectors of a heterogeneous computing system for distributed science

for rapid dispatch of computations for data analysis such as feature extraction and machine learning-based classification and regression of the results. 5G technology can play a significant role in this integration by enabling high-rate, low-latency data exchange between instruments while allowing access to edge computing resources. Non-fully-autonomous instruments could be remotely controlled/monitored through wireless networks. However, cyber-physical systems open new avenues for exploitation. For instance, it has been demonstrated that autonomous cars can be brought to a halt by sending spurious signals to their LIDAR, and computer vision systems can be tricked into thinking a stop sign is a speed limit sign by poisoning the learning dataset and adding a sticker to actual signs. Although science ecosystems are more secure, new technologies may enable new attacks. For instance, a Nation State adversary could submit a specially crafted input to gain unauthorized access to a bioengineering facility to cause synthesis of a dangerous microorganism. As autonomous laboratories, synthetic biology, and bioengineering grow, we must minimize the risk of them being used for malicious purposes.

3 Challenges in Cybersecurity and Privacy

Figure 1 shows examples of attack vectors that could compromise a heterogeneous computing system for distributed science. An adversary could plant malicious sensors that poison the data or deploy a rogue base station to either jam the signal or steal data. A sophisticated adversary could compromise servers deployed at the edge by a supply chain attack—particularly problematic because edge computing servers reside on the network. Furthermore, if the adversary can compromise a network on the continuum between sensor and edge or sensor and HPC, they could launch a man-in-the-middle attack that can lead, for example, to data exfiltration. The cybersecurity community has developed robust countermeasures for these attacks, such as encryption, firewalls, and intrusion detection systems, and trusted hardware (e.g., Intel SGX) can provides security guarantees to solutions such as FaaS. Nevertheless, the increasing heterogeneity of distributed science ecosystems makes the deployment of such solutions challenging. For instance, using encryption in resource-constrained IoT devices may compromise accuracy of measurements [6]. Moreover, we cannot always ensure that we will have trusted enclaves for deploying edge computing or FaaS solutions. We need to investigate and develop novel cybersecurity approaches for distributed science ecosystems that take into account the heterogeneity of the future digital continuum.

References

[1] P. Beckman *et al.*, "5G enabled energy innovation: Advanced wireless networks for science, workshop report," March 2020.

[2] J. Napolitano, "DOE awards $1.2M to Argonne and Northwestern to maximize scientific data sent over 5G network." `https://www.anl.gov/article/doe-awards-12m-to-argonne-and-northwestern-to-maximize-scientific-data-sent-over-5g-network`.

[3] G. C. Sankaran, J. Chung, and R. Kettimuthu, "Leveraging in-network computing and programmable switches for streaming analysis of scientific data," in *7th IEEE International Conference on Network Softwarization*, pp. 1 – 5, 2021.

[4] R. Chard, Y. Babuji, Z. Li, T. Skluzacek, A. Woodard, B. Blaiszik, I. Foster, and K. Chard, "Funcx: A federated function serving fabric for science," in *29th Intl Symp. on High-Performance Parallel and Distributed Computing*, p. 65–76, 2020.

[5] A. Aspuru-Guzik and K. Persson, "Materials acceleration platform," *Mission Innovation*, 2018.

[6] B. Bordel, R. Alcarria, J. Chung, R. Kettimuthu, and T. Robles, "Evaluation and modeling of microprocessors' numerical precision impact on 5G enhanced mobile broadband communications," in *Information Tech. & Systems*, pp. 267–279, 2021.

Unified privacy, generalization, and convergence for large-scale deep learning

Guojing Cong, Steven Young, Vandy Tombs, and Don March
{congg, youngsr, tombsvj, marchdd}@ornl.gov

Topics: Differential privacy, Federated AI/ML, Randomized Algorithm, Stochastic Optimization

Challenge: With increased awareness from the general public and growing interest from the federal agencies, privacy is becoming a necessity in many domains for machine learning. In order for a model trained with sensitive data to be shared with the public for maximum benefit, the privacy of the data used to train the model must be preserved. Recent studies have shown that sensitive information in training data can be reconstructed by manipulating the trained models, even when all training data are protected by a mechanism like secure enclave. Researchers approach this problem from the perspective of differential privacy [**DR14**]. In deep learning, methods such as differentially private stochastic gradient descent (DPSGD) [**AC16]** prevent privacy leaking by adding random noise to the computed gradients and by clipping the "large" gradients.

While these methods improve privacy, they have serious negative impact on training efficiency and model accuracy. Training time grows substantially and accuracy degrades significantly, making learning on large inputs even more challenging. Additionally, with current privacy preserving deep learning approaches such as DPSGD, efficient methodologies to validate and evaluate privacy claims are lacking. Currently, a trained model is produced with accompanying privacy accounting signatures such as ε and δ. In contrast to convergence and generalization, no quick validation of privacy claim is available. The user either has to trust the learning process or will have to inspect the source code to carry out the privacy accounting by herself. Even though a massive amount of data have been collected such as health-care records, these issues pose a huge challenge for maximizing the value of the data for the benefits of the general public and national interests.

Opportunity: Opportunities abound in providing fast, scalable machine learning approaches that produce models with good generalization performance without leaking private information. Two aspects, randomness and augmentation, are key to achieving such objectives.

Role of randomness: First we note that the privacy requirement demands new stochastic optimization approaches that address privacy, generalization, and convergence in a unified manner, as all three aspects are intimately linked to randomness. Understanding the impact of randomness on these three aspects and carefully orchestrating randomness in training approaches is critical to good privacy, generalization and performance behavior. Depending on the form of randomness, it can be beneficial to some aspects but harmful to others. For example, random sampling in SGD has been shown to result in better generalization than full-batch gradient descent [**AK21**]. In methods like DPSGD, random noise added to the gradients prevents privacy leakage but hurts convergence. Yet research has also shown that random noise added to the model during training with SGD can help avoid getting stuck at saddle points thus accelerate convergence [**JG17**]. It is currently very difficult to design and implement scalable solvers for a specific target of convergence, generalization, and privacy goals.

Introducing the right amount of randomness in the appropriate form at the right algorithmic steps is key. Many open questions need to be addressed, and prior conclusions may need to be re-evaluated. How is a model's generalization performance related to its privacy guarantee? If (differential) privacy is introduced, is SGD necessarily better in terms of convergence and generalization than whole-batch or large-batch gradient descent as previously established? Can distribution robust optimization [**YH19**] be better adapted to privacy preserving optimization than vanilla SGD? Given a privacy budget, what is the

best optimizer and what kind of generalization and convergence behavior can be expected? For scalable solvers on HPC systems, what are the best ways to provide privacy guarantees? And ultimately, how can an algorithm help the practitioner balance privacy, convergence, generalization and resource consumption for her task?

Role of augmentation: In most current solvers including DPSGD, gradient or model augmentation is routinely employed for achieving the privacy, generalization or convergence goals. For example, DPSGD clips large gradients; momentum methods establish a momentum on a sequence of updates to the model. Privacy signatures for momentum methods, heavy-ball methods, adaptive gradient methods such as Adam [**KB15**] and AdaGrad [**DH11**], and regularization methods need to established. Additionally, distributed training such as model averaging [**ZC18**] (including federated learning) can be viewed as model or gradient augmentation as the models are averaged and the gradients are summed; these methods can obfuscate gradients from a privacy perspective. Thus if we aim to maintain convergence and generalization performance while introducing differential privacy, the best approach for augmentation needs to be found.

Timeliness: Progress in this area addresses the urgent need to share models trained on sensitive data (and train them in a federated manner) in order to provide the most public benefit. This need is pervasive throughout healthcare, scientific research, and other problems of national interest. The development of algorithms and implementations will lay the foundation for building systems around future security enclaves. Having a verifiable machine learning mechanism for privacy will enable the public, institutions, and industries to share data and form new collaborations while maintaining confidence in their privacy. For the Department of Energy, this will provide its scientists to have a provable and flexible way to disseminate results while protecting sensitive information.

In the near term, augmenting current deep learning practices for privacy and accelerating differential privacy training methods will address some of the immediate needs. Ultimately, however, "optimal" and "verifiable" solutions will need to be developed within a solid algorithmic and practical framework. We expect this work will build on the existing foundations of stochastic optimization and deep learning. Taking a unified view towards privacy, convergence, and generalization is not only possible but also imperative as private models that are not useful have very limited value.

References

[DH11] *John Duchi, Elad Hazan, and Yoram Singer,* "*Adaptive suggradient methods for online learning and stochastic optimization*", *Journal of Machine Learning Research 12 (2011)*
[**DR14**] *Dwork, Cynthia and Roth, Aaron*, "*The Algorithmic Foundations of Differential Privacy", Foundations and Trends in Theoretical Computer Science, 2014*
[KB15] *Diederik P. Kingma and Jimmy Ba, "Adam: A Method for Stochastic Optimization", ICLR 2015*
[AC16] *Martin Abadi, Andy Chu, Ian Goodfellow, et al., 2016. "Deep Learning with Differential Privacy", 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*
[JG17] Chi Jin, Rong Ge, et al., "*How to escape saddle points efficiently", Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017*
[ZC18] *Fan Zhou and Guojing Cong, "On the Convergence Properties of a K-step Averaging Stochastic Gradient Descent Algorithm for Nonconvex Optimization", 2018 International Joint Conference on Artificial Intelligence, IJCAI 2018*
[YH19] *Kai Yang, Jianwei Huang, et al., "Distribution Robust Optimization", Springer Optimization and Engineering manuscript, 2019*
[AK21] *Zaghloul Amir, Tomer Koren and Roi Livni, "SGD Generalizes Better Than GD (And Regularization Doesn't Help)", COLT 2021*

Point of Contact: Dave Cowley david.cowley@pnnl.gov , Pacific Northwest National Lab, EMSL

Background:
The Environmental Molecular Sciences Lab (EMSL) at PNNL is a DOE BER scientific user facility with an annual budget of over $43M. In FY20, EMSL had 216 active user projects that involved 766 users. As a user facility, it accepts proposals from across the global scientific community, in response to calls for proposals that reflect current and emerging priority areas for BER. The proposals that are most highly ranked by external peer reviews are awarded EMSL user projects, which provide access to EMSL's instruments and staff expertise. EMSL hosts over 150 scientific instruments, which include a broad range of mass spectrometry capabilities, cellular and molecular imaging, microscopy, NMR, and unique technologies.

EMSL also hosts a variety of computing resources to support modeling work as well as data analytics on experimental data captured at EMSL and elsewhere. Computing resources include: the heterogeneous supercomputer Tahoma, a mid-range resource for diverse BER needs; the archival file storage system Aurora; Kubernetes-based web applications for data analysis and integration; and high-speed file transfer via ESnet and GLOBUS. EMSL has invested in a new system for data and metadata, NEXUS, to ensure compliance with FAIR principles. EMSL has partnerships with multiple other scientific user facilities, including the Joint Genome Institute (JGI) and Atmospheric Radiation Measurement (ARM). Some of these are activities at the user program level (FICUS) and others are at the operational level (providing additional compute capacity). EMSL computing is presently engaged with ASCR's Distributed Computing and Data Ecosystem (DCDE) [4] via the Future of Laboratory Computing Working Group (FLC/WG) of the National Laboratory Research Computing Group (NLRCG).

Activities relevant to workshop priority areas:
EMSL's activities cover numerous areas relevant to the workshop focus, particularly in automation, integration, and complex scientific workflow pipelines. As a result, EMSL would be a model partner for identifying use cases, embedding prototypes, and validating deliverables. EMSL recently published a new decadal roadmap and strategic plan [1], which focuses on building capabilities to address the grand challenge problems identified by DOE BER [2, 3]. Activities and plans central to workshop areas are summarized below.

1. Integration: EMSL is piloting the capability to provide a form of elastic compute to another user facility (JGI) via JGI's workflow system JAWS: data is transferred through GLOBUS, alongside critical job data/metadata. An additional challenge for EMSL in security and privacy, especially in offering integrated capabilities, is the fact that instrument procurement and operation are often shared by multiple PNNL organizations or sponsors, e.g. proteomics and metabolomics instrumentation shared with PNNL's Biological Sciences Division; or the EMSL/BER Krios CryoEM. A third, long-standing challenge is that vendors often provide specialized workstations for instrument's primary data collection/quality control software, which is specific to particular operating systems that don't/cannot be updated, and likewise for drivers.
2. Complex scientific workflows: EMSL is developing multiple automated systems in which uploading data triggers computational workflows. In these and other areas, as we engage more with public cloud providers (e.g. collaborators), there are multiple risks, including triggering of unnecessary or malicious workloads, lack of spend limits, etc. As an example, the BER-funded CD-MII effort (PI: Tim Scheibe) couples data/metadata and simulation capabilities – complex, computational workflow.

3. Automation: EMSL already has several significantly automated capabilities, particularly in high-throughput 'omics. With BER support, EMSL is leading development of a plan for an automated facility for microbial phenotyping. EMSL is also developing automated platforms for soil analysis and organic matter characterization. One question of great interest is, "How can we manage risks associated with deploying research software involving AI/ML for automation?" Separately, given diversity of user science, privacy technologies may be applicable in automation, as a means to avoid revealing not-yet-released data/biasing analyses/instrument guidance/etc.
4. Instrument development and deployment: EMSL is often at the cutting edge of a range of capability development and deployment. Both spaces challenge common security models and accordingly COTS solutions, owing to both time and cost pressures as well as EMSL's open and operational environment. Examples in capability development include Bio Atom Probe Tomography (APT), NanoPOTS (Nanodroplet Processing in One pot for Trace Samples), Dynamic Transmission Electron Microscopy (DTEM), and the CoreMS framework that enables standardized analysis workflows of MS experiments. Examples in deployment include the 21 Tesla Fourier transform ion cyclotron resonance mass spectrometer (FTICR MS) (one of two in the world when deployed) and CAMECA's Nanoscale secondary ion mass spectrometer (NanoSIMS: EMSL is the first user facility in the US to offer this platform). As a result of these activities, EMSL has strong relationships with vendors, leading to partnerships in capability development and EMSL technologies being licensed for external use. One example of the latter would be the licensing of PspecteR for analysis of proteomics data.
5. Remote access to instruments: It is of interest to provide easier remote access to instruments, . This is of course particularly spurred by COVID, but was of interest before. There are additional considerations around remote access driven by early analysis of preliminary results. Many of the same concerns for remote access are the same as for automation.

Other considerations about EMSL:
As an open user facility, EMSL has special considerations compared to traditional PI-driven labs or industry labs. These include the requirement for interfaces for users (the NEXUS web portal as well as HPC resources and remote access to instruments), citizenship and background check requirements, and a large number of on-site visitors who bring their own laptops, USB storage, etc. It should be noted that EMSL's considerations around data privacy are distinct from many in the biomedical facility space, particularly because EMSL computing and data storage resources are not for patient data or other data having regulatory requirements on it. This is an explicit and permanent hard requirement that users must attest to, as part of proposal submission.

Many use cases are difficult or impossible to identify *a priori* given that (as a user facility) the diversity of scientific workflows is essentially infinite. Further, EMSL establishes general policies to protect against projects that would violate security practice, leading both users and EMSL staff to develop and refine proposals that stay within the boundaries. Clearly, however, the proliferation of automation and cyber-physical integrations for cutting-edge scientific platforms will challenge current practices and necessitate new technologies and frameworks for managing risk.

References:
[1] EMSL Strategic Plan https://content-qa.emsl.pnl.gov/sites/default/files/2021-07/EMSLStrategicPlanFY2021_0.pdf
[2] BSSD Strategic Plan https://genomicscience.energy.gov/2021bssdstrategicplan/
[3] CESD [now EESSD] Strategic Plan https://science.osti.gov/-/media/ber/pdf/workshop-reports/2018_CESD_Strategic_Plan.pdf

Using Commutative Filters to prevent Adversarial ML Attacks

Dipankar Dasgupta
Computer Science Department
The University of Memphis
dasgupta@memphis.edu

Abstract:
We developed a trustworthy AI/ML framework that employs an adaptive strategy to inspect both inputs and decisions. In particular, input streams are examined by a series of diverse filters before sending to the learning system and then crossed checked its output through anomaly (outlier) detectors before making the final decision. Experimental results (using benchmark datasets) demonstrated that our commutative-filtering strategy could mitigate adaptive or advanced adversarial manipulations for wide-range of ML attacks with higher accuracy.

Introduction:
Machine Learning (ML)-based decision support techniques, e.g. deep neural networks (DNNs), are widely being used, which are trained offline (supervised learning) using large datasets of different types. In spite of their major breakthroughs in solving complex tasks, it has been lately discovered that ML techniques (especially artificial neural networks and data-driven artificial intelligence) are highly vulnerable to deliberately crafted samples (i.e., adversarial examples) either at training or at test time. All the adversarial attacks (either by input data manipulation or ML system using TrojAI) can be broadly of two types: misclassification within a class, out of scope data to map to a member class which results in different False Positive (FP) and False Negative (FN) performances [1]. There are three basic types of adversarial attacks: (1) Poisoning attack: In this attack, the attacker can corrupt training data and create adversarial examples later to work on the model. It happens in training time. (2) Evasion attack: In this attack, testing inputs change in a way that they miss-classify to another random or targeted class. (3) Trojan AI attack: In this attack, the AI model's architecture changes in a way so that it misclassifies the input. To safeguard ML techniques against malicious adversarial attacks, several countermeasure schemes have been proposed. These countermeasures generally fall within two categories: adversarial defense and adversarial detection.

Commutative Filtering Framework:

Towards robustly handling such adversarial attacks, a preemptive filtering scheme is developed to provide robust ML platform for decision support systems. It employs different filtering mechanisms (at the input and at the output/decision end of learning systems) to thwart adversarial attacks. In particular, input filter's main aim is to filter misleading and out of distribution inputs (e.g., image of animal but not human face in a face recognition system). The output filter's goal is handling larger variations and restricting misclassification rates to improve overall accuracy of the system. A machine learning framework usually consists of four main modules: feature extraction, feature selection (optional), classification/clustering, and decision. As depicted in Fig. 1, the input and output filters are placed before pre-processed feature are fed to the input filter to determine if the received feature/sample is either clean or noisy/adversarial and accept or reject accordingly and then pass to main ML model. The outcome of ML module after classification/clustering/raw decision is given to the output filtering (negative filtering mechanism [2]) for further scrutiny. The output filters use context-information and/or communicates with the input filter bank to make the correct final decision. The CF framework employs a filtering scheme which forms a closed loop via signaling and message-passing mechanisms. An ensemble of different noise removal or Adversarial Attack detection filters was successfully applied in a recent work [3-5].

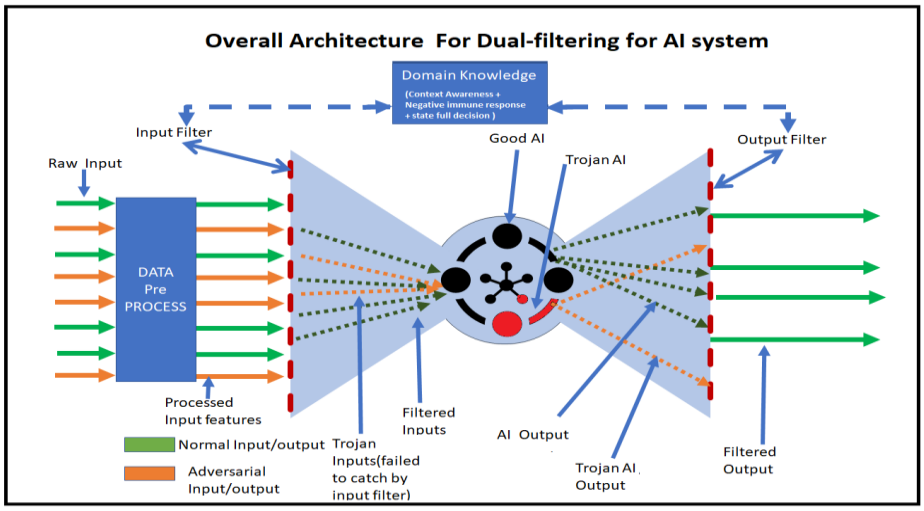


Figure 1: Schematic of the proposed commutative-filtering framework.

Conclusions:

The developed CF software can be used as a wrapper to any existing ML-based decision support system in order to prevent a wide variety of adversarial attacks such as manipulated input and contaminated learning systems in which existing heavy-weight trained ML-based decision models likely to fail. Our CF framework will be a game changer in developing trustworthy ML systems. Moreover, it will benefit society and national security as developed intelligent framework will also be applicable in related fields, e.g., medical image forgery detections, etc., thereby creating more secured and trustworthy identification of faces and other images online. Our methods will have an impact on the analysis of all types of AI employed in different applications and information security and will work as an additional layer of defense shield against attacks on learning systems.

References:

[1]. S. Tian, G. Yang, and Y. Cai. Detecting adversarial examples through image transformation. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[2]. Kishor Dattagupta and Dipankar Dasgupta. “Using Negative Detectors for Identifying Adversarial Data Manipulation in Machine Learning” Conference: “2021 International Joint Conference on Neural Network (IJCNN), 2021.

[3]. Kishor Dattagupta and Dipankar Dasgupta. "Who is responsible for Adversarial Defense”, Workshop on Challenges in Deploying and monitoring Machine Learning Systems, ICML 2021

[4]. K. D. Gupta, D. Dasgupta, Z. Akhtar. Adversarial Input Detection Using Image Processing Techniques (IPT). In 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), pp. 309-315, October 28, 2020.

[5]. K. D. Gupta, D. Dasgupta and Z. Akhtar, "Applicability issues of Evasion-Based Adversarial Attacks and Mitigation Techniques," *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, Canberra, ACT, pp. 1506-1515, Australia, December 5, 2020.

Towards Transparent Data Management Workflows for Analyzing Privacy-Utility Trade-Offs in Human-Building Interactions

Aritra Dasgupta, Assistant Professor
Department of Data Science, New Jersey Institute of Technology
aritra.dasgupta@njit.edu aedeegee.github.io
Co-authors: Vikas Chandan and Soumya Kundu; Pacific Northwest National Laboratory

In building energy modeling, a persistent challenge is to understand how human behavior drives energy consumption. Besides extrinsic factors such as weather, building layout, etc., consumption is largely driven by the number and activities of occupants and their preferences for indoor conditions. Granular data about human-building interaction is like a double-edged sword. On the one hand, there is immense value in collecting and sharing occupant data [Hu19]. They can be used for developing more accurate energy consumption models using machine learning techniques. These models can be trained by curating benchmark buildings data, which when shared with researchers, can lead to community-led improvement of the state-of-the-art modeling strategies. However, on the flip side, occupant data can not only have sensitive attributes, but even their anonymized version can be susceptible to inference attacks. Attack strategies can be used by malicious actors to gain information that can pose both privacy and security risks. For example, they might be able to infer the habits of individuals and even know precise times when buildings might have low occupancy. As we can observe, there is a fundamental trade-off between preserving the privacy of human behavior and activities in buildings data and their value and utility for accurate calibration of energy consumption patterns. Addressing this trade-off will be both a challenge and opportunity to develop futuristic solutions towards developing smarter grid-interactive buildings and addressing sustainability goals of our built environment by decoding human footprint and behavior.

We posit that transparent data management workflows will be needed to calibrate both privacy risks and the potential value of occupant information in buildings. We imagine a privacy-as-a-service solution for data custodians (e.g., organizations and third-party utility companies) that can help quantify, simulate, and visualize the risk-utility trade-off in real-world scenarios. We believe that the two important components for privacy-as-a-service will be: i) end-to-end provenance where various post-hoc transformations of the data can be tracked and assessed after applying privacy-preserving algorithms, and ii) privacy-preserving data visualization where one can interactively tune parameters for developing a mental model about the risks and loss of data utility and make decisions about risk tolerance for accessing and sharing data.

Motivating use cases that are impeded by privacy concerns:

Facilitating Demand-Response for Grid-Interactive Buildings: Grid Interactive Efficient Buildings (GEB) have been identified by DOE [Neu19] as a key enabler to achieve energy efficiency and decarbonization goals. It allows buildings to transition to a flexible resource, powered by smart technologies and communications, leveraging underlying flexibility in the operation of sub-systems, such as heating, ventilation, and air conditioning (HVAC). However, the participation of occupants, whose activities and behaviours are the drivers of consumption, is key to realizing the full potential of the GEB concept. Hence. data relating to occupancy, occupant preferences and their

interactions with the building is needed but this also poses privacy risks. These concerns have been highlighted in literature [Rot19, Kur11].

Pre-empting Adversarial attacks: Building automation systems (BAS) are susceptible to adversarial attacks, and cyber-security has been recognized as an important challenge going forward as buildings become more automated and building operations become more data-driven [Fu21, Myl17]. Including privacy sensitive data, such as occupancy and occupant interaction as part of the BAS, makes them an attractive target for malicious agents. However, such data also enables the BAS to be more effective in achieving underlying goals of building management. For example, information about zone level occupancy across the buildings can help the operator decide which zones to target for minimizing energy waste.
Curating Buildings Benchmark Data: There is need to develop benchmark datasets that can act as a standard resource to compare Machine Learning and other approaches being developed for serving use cases. DOE is actively supporting efforts to develop such datasets. However, several building use cases need access to occupancy and occupant activity data. The practical experience of the DOE sponsored Benchmark Datasets project team [DOE21] has been that it is very difficult to convince building managers to release such data, not just publicly but even privately.

Future research directions:

1) Developing large scale open data infrastructure with privacy guarantees and utility benchmarks will be necessary to scale up the effort towards curating/sharing buildings data.
2) Privacy-As-A-Service for data custodians will be needed to augment data management workflows so that trade-offs between privacy risk and utility can be transparently assessed.
3) Advanced Human-Machine Interfaces for Trustworthy Decision-making with the help of visualization techniques can help implement privacy policies and simulate what-if scenarios for preventing adversarial attacks.
4) **Privacy-preserving machine learning for building energy modelling** will help train large scale models for more accurate energy consumption prediction while at the same time guarding against potential model inversion attacks.

References:

[Hu19] Huebner, G. M., & Mahdavi, A. (2019). A structured open data collection on occupant behaviour in buildings. *Scientific data*, *6*(1), 1-4.
[Le21] Lee, D., & Hess, D. J. (2021). Data privacy and residential smart meters: Comparative analysis and harmonization potential. *Utilities Policy*, *70*, 101188.
[La19] Langevin, J., 2019. Longitudinal dataset of human-building interactions in US offices. *Scientific data*, *6*(1), pp.1-10.
[Bha20] Bhattacharjee, K., Chen, M., & **Dasgupta, A**. (2020, June). Privacy-preserving data visualization: reflections on the state of the art and research opportunities. In *Computer Graphics Forum* (Vol. 39, No. 3, pp. 675-692).
Dasgupta, A., Kosara, R., & Chen, M. (2019, October). Guess me if you can: A visual uncertainty model for transparent evaluation of disclosure risks in privacy-preserving data visualization. In *2019 IEEE Symposium on Visualization for Cyber Security (VizSec)* (pp. 1-10). IEEE.
[Neu19] Neukomm, M., Nubbe, V., & Fares, R. (2019). *Grid-interactive efficient buildings* (No. DOE/EE-1968). US Dept. of Energy (USDOE), Washington DC (United States); Navigant Consulting, Inc., Chicago, IL (United States).
[Rot19] Roth, A. and Reyna, J., 2019. *Grid-interactive efficient buildings technical report series: whole-building controls, sensors, modeling, and analytics* (No. NREL/TP-5500-75478; DOE/GO-102019-5230). National Renewable Energy Lab.(NREL), Golden, CO (United States).
[Kur11] Kursawe, K., Danezis, G. and Kohlweiss, M., 2011, July. Privacy-friendly aggregation for the smart-grid. In *International symposium on privacy enhancing technologies symposium* (pp. 175-191). Springer, Berlin, Heidelberg.
[Fu21] Fu, Y., O'Neill, Z., Yang, Z., Adetola, V., Wen, J., Ren, L., Wagner, T., Zhu, Q. and Wu, T., 2021. Modeling and evaluation of cyber-attacks on grid-interactive efficient buildings. *Applied Energy*, *303*, p.117639.
[Myl17] Mylrea, M., Gourisetti, S.N.G. and Nicholls, A., 2017. An introduction to buildings cybersecurity framework. In *2017 IEEE symposium series on computational intelligence (SSCI)* (pp. 1-7). IEEE.
[DOE21] US Department of Energy, Building Technologies Office, *Benchmark Datasets,* https://bbd.labworks.org/ (Accessed October 14, 2021)

Adversarial modeling, simulation, and learning for trustworthy scientific computing ecosystems

Ewa Deelman*(corresponding author), Prasanna Balaprakash,
Mariam Kiran, and Anirban Mandal
University of Southern California, Argonne National Laboratoy,
Lawrence Berkeley National Laboratory, and RENCI - UNC Chapel Hill
deelman@isi.edu

Topic: Adversarial modeling & simulation, graph algorithms

DOE science workflows are becoming increasingly complex, executed on geographically distributed computing ecosystems that are managed by different labs and domain scientists. Consequently, the operators and the researchers that utilize these resources have inadequate understanding of the behavior of the entire set of resources that science workflows span. This difficulty is further exacerbated by the emerging data-driven AI workflows that connect experimental facilities and leadership computing systems and beyond 5G networks that can provide high bandwidth and low latency data transfers [1]. A single weak security link in the entire complex workflow can allow adversaries to corrupt the scientific results, which can directly affect the integrity of the DOE science and the trustworthiness of scientific computing ecosystems. There are several challenges that the DOE/ASCR community has to overcome. Here, we focus on an integrated adversarial modeling, simulation, and learning approach to enhance the trustworthiness of scientific computing ecosystems.

1. Adversarial modeling, simulation, and testbeds:

Adversarial modeling and simulation are critical to establish and to enhance trustworthiness in DOE scientific workflows. To that end, we need effective and scalable tools that allow us to inject various types of adversarial behavior into scientific computing workflows and simulate their signatures. Major challenges in developing workflow execution simulators are accuracy, i.e., the ability to capture the behavior of a real-world system, and scalability, i.e., the ability to simulate large and/or long-running execution scenarios efficiently, obtaining results in a reasonable amount of time, and being able to simulate systems of sufficient size and complexity to make the simulations realistic [2]. We envision a highly configurable multi- fidelity/resolution simulator that provides various services for adversarial modeling and simulation,. These will include *compute services* that provide access to compute resources to execute workflow tasks; *storage services* that provide access to storage resources for storing workflow data; *network monitoring services* that can be queried to determine network distances; and *adversarial modeling and injection services* that allows for the development and injection of various types of anomalies into the workflow and simulate their effect.

Highly configurable experimental testbeds representative of scientific computing ecosystems will enable the execution of what-if scenarios, running unconventional software and intentionally adversarial workloads, allowing injections of several kinds of adversarial perturbations and insertion of artificial loads on compute and storage resources or changing network latency, protocols and other parameters to emulate different scenarios. These testbeds can be used to validate the adversarial modeling and simulation efforts and collect more realistic data. To that end, we envision cybersecurity-centric experimental testbeds that enable simulator calibration, validation, and tuning.

2. Simulator surrogate with graph neural networks: A major challenge in the conventional adversarial modeling and simulation approach is the time required for simulation. The number of possible adversarial scenarios that one needs to simulate grows exponentially with respect to the

number of perturbation parameters. We cannot simulate and study all those adversarial scenarios even on yotta- and zetta- scale machines. To that end, a promising approach is to develop machine-learning-based surrogate models that can provide fast approximation to the simulator. In a typical supervised learning formulation that requires a number of training examples of input-output pairs $(x,\ y)$ to develop surrogate models, we can partition the input x into three components: 1) a workflow graph $x^g = (V, E)$, where each vertex $v \in V$ describes a scientific workflow and each edge $e \in E$ represents the dependency; 2) platform parameter vector x^p that describes the target platform on which the workflow will be run; 3) adversarial perturbation vector x^e that describes types of abnormal perturbations that will be injected into the platform. Similarly, the output y can have different components: is there an adversarial behavior and if so, when and where. To develop surrogate models for the simulator, message-passing neural networks (MPNNs), a class of graph neural networks (GNNs) [3], are promising because they provide a generic framework to incorporate node and edge features of workflows. The advantage of MPNNs over other GNNs is their ability to integrate edge features into the message passing of node features. Specifically, edge features are processed by the MPNN to generate weights to guide the message passing between nodes. This is particularly critical in our setting for learning complex interdependencies in the workflow and how anomalies affect the behaviour. Moreover, it is critical to develop explainable and interpretable MPNNs with uncertainty quantification capabilities for increasing trust in the MPNN model and for understanding of the impact of adversarial behavior and the corresponding workflow characteristics.

3. Real-time adversary detection and attribution: Ability to detect adversarial behaviors and attribute them to a particular class in real time are among the key properties of a trustworthy scientific computing ecosystem [4]. We envision developing new physics-/domain- informed machine learning based approaches that can detect adversarial behavior during workflow execution. These models will be developed based on normal and adversarial data coming from the simulator, and using data collected on testbed infrastructure being perturbed in a systematic fashion. A workflow running on a platform will generate a multivariate time series of monitoring metrics, both from the workflow and from the infrastructure, such as time series data for (1) compute, memory, and I/O metrics for CPUs and hardware accelerators as observed by workflow jobs, (2) network performance metrics observed by workflow data transfers, (3) CPU, disk, memory utilization, network interface statistics, packet error rates, *tstat* metrics and other network telemetry from the infrastructure. To that end, a promising research direction is spatial temporal machine learning methods that operate on graphs. The scientific computing ecosystem can be modeled as a graph, wherein the multivariate time series of monitoring metrics will evolve over time. We envision leveraging both unsupervised and supervised graph-based spatial temporal deep learning methods. The former can be used to learn latent low-dimensional representations of the multivariate time series data, which can be used to separate normal from adversarial data. The latter can be used to identify adversarial anomalies.

References

[1] P. Beckman et. al. 5G enabled energy innovation: Advanced wireless networks for science, workshop report. 2020. doi: 10.2172/1606538.

[2] H. Casanova et. al. Developing Accurate and Scalable Simulators of Production Workflow Management Systems with WRENCH. *Future Generation Computer Systems*, 112:162–175, 2020. doi: 10.1016/j.future.2020.05.030.

[3] Zonghan et. al. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[4] S. Piesert. ASCR cybersecurity for scientific computing integrity. 2015.

Cybersecurity as a case study for evaluating critical questions in Federated ML

Dr. Benjamin Blakely, CISSP, CISM; Dr. William Horsthemke, CISSP; Dan Harkness, CISSP; Dr. Nate Evans, CISSP
Argonne National Laboratory, Strategic Security Sciences; October 2021

Many problem domains of scientific computing rely on information that is too sensitive to share, for reasons of security and privacy (e.g., medical [1], financial [2], cybersecurity, military). This is particularly problematic in the problem domain of machine learning, as many of these algorithms rely on ground truth datasets and they generate more robust and generalizable models with inputs from a diversity of sources. Federated machine learning has the potential to open new avenues by moving computation to edge devices. This not only might address the security and privacy concerns, but the inherently distributed nature also decreases the reliance on centralized computing resources.

There are challenges in a federated machine learning model, however. It might be difficult to determine an optimal algorithm and its hyperparameters *a priori* for a system of federated machine learning nodes, and finding the appropriate interval for retraining and synchronization could impact accuracy and convergence. Finding an optimal model requires iteration and thus coordination between autonomous agents in a swarm configuration, or with a centralized coordination system. Swarm configurations might have advantages in terms of flexibility and resiliency, but might struggle more to converge than a centralized coordination system. Edge devices also might lack sufficient computational power to train models, versus to evaluate new inputs on existing models due to a lack of in-built GPU/TPUs or the nature of the device (e.g., low-power mobile or IoT systems). The trustworthiness of learned parameters is paramount – unless there is sufficient diversification in edge devices so as to dilute the potential of any single device to poison the shared model, security considerations of participants must be considered [3].

Cybersecurity-related machine learning presents an ideal area to study these challenges and opportunities due to the use of standardized protocols, and a degree of homogeneity in systems, attack vectors, and risks. By their nature these application areas may consist of large numbers of systems with computational capabilities; and the lack of available datasets (due to restrictions on information sharing) has been a critical limitation to cyber/ML research to-date. Current Literature on Cybersecurity-related federated ML exists and includes malicious URL detection [4], intrusion detection [5], anomaly detection on lower-power IOT devices [6], IOT ensemble learning [7], handling gradient delay for convergence [8], and we expect to see several new examples in the next few months [9]. However, literature on comparative performance of a federated machine learning-based intrusion detection system (IDS) against currently available signatured-based engines is light compared to the broader IDS/ML literature.

We propose there are a number of vital questions to be answered to advance the state of the art in federated ML that can be answered with the use of cybersecurity-based experiments. These experiments represent a unique opportunity to contribute to two related fields simultaneously, while also establishing the foundation for an operational program of deploying systems built on these principles to protect HPC resources. The specific research problems we propose would comprise such a research agenda include:

Optimize Model Design Strategies

1) Determining how to "seed" a federated machine learning model for cybersecurity purposes with initial algorithmic choices and hyperparameter values, and how to adapt these over time as necessary.
2) Determine how model convergence and accuracy depend upon the number of nodes, algorithm used, and types of cyber scenarios of interest.

3) Determine whether it is necessary to use generative models to balance training inputs.
4) Determine optimal information sharing and re-training intervals as a function of node population size, input size and change rate, or convergence/accuracy impacts.

Compare Federation Architectures

1) Measure the performance and energy impacts (or limitations imposed on learning by resource constraints) on edge devices.
2) Evaluate hybrid strategies where a GPU/TPU-enabled local master node receives data from edge devices, maintains an online model, and shares learned parameters with peer or superior nodes.
3) Determine which information sharing topologies perform best – fully decentralized (i.e., swarm), hierarchical, or centralized.

Enable Privacy Strategies

1) Evaluate relative utility of feature inputs to determine which are more readily masked to further address privacy or sharing concerns.

Assess Risk of Adversary Attacks

1) Evaluate potential for an adversary in possession of the trained parameters to gain information about the training inputs and networks
2) Evaluate the potential for an adversary to poison the overall network by compromising various ratios of nodes.

The work presented in this paper was partially supported by the U.S. Department of Energy, Office of Science under DOE contract number DE-AC02-06CH11357

[1] N. Rieke *et al.*, "The future of digital health with federated learning," *Digital Medicine*, p. 7, 2020, doi: 10.1038/s41746-020-00323-1.
[2] W. Zheng, L. Yan, C. Gou, and F.-Y. Wang, "Federated Meta-Learning for Fraudulent Credit Card Detection," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, Yokohama, Japan, Jul. 2020, pp. 4654–4660. doi: 10.24963/ijcai.2020/642.
[3] R. A. Mallah, G. Badu-Marfo, and B. Farooq, "Cybersecurity Threats in Connected and Automated Vehicles based Federated Learning Systems," *arXiv:2102.13256 [cs]*, Jun. 2021, Accessed: Oct. 02, 2021. [Online]. Available: http://arxiv.org/abs/2102.13256
[4] E. Khramtsova, C. Hammerschmidt, S. Lagraa, and R. State, "Federated Learning For Cyber Security: SOC Collaboration For Malicious URL Detection," in *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, Singapore, Singapore, Nov. 2020, pp. 1316–1321. doi: 10.1109/ICDCS47774.2020.00171.
[5] Y. Sun, H. Ochiai, and H. Esaki, "Intrusion Detection with Segmented Federated Learning for Large-Scale Multiple LANs," in *2020 International Joint Conference on Neural Networks (IJCNN)*, Glasgow, United Kingdom, Jul. 2020, pp. 1–8. doi: 10.1109/IJCNN48605.2020.9207094.
[6] T. Zhang, C. He, T. Ma, L. Gao, M. Ma, and S. Avestimehr, "Federated Learning for Internet of Things: A Federated Learning Framework for On-device Anomaly Data Detection," *arXiv:2106.07976 [cs, eess]*, Sep. 2021, Accessed: Oct. 02, 2021. [Online]. Available: http://arxiv.org/abs/2106.07976
[7] D. C. Attota, V. Mothukuri, R. M. Parizi, and S. Pouriyeh, "An Ensemble Multi-View Federated Learning Intrusion Detection for IoT," *IEEE Access*, vol. 9, pp. 117734–117745, 2021, doi: 10.1109/ACCESS.2021.3107337.
[8] P. Tian, Z. Chen, W. Yu, and W. Liao, "Towards asynchronous federated learning based threat detection: A DC-Adam approach," *Computers & Security*, vol. 108, p. 102344, Sep. 2021, doi: 10.1016/j.cose.2021.102344.[1]
[9] A. Dehghantanha, R. Parizi, Q. Zhang (Eds.), "Special Issue on Federated Learning for Decentralized Cybersecurity," [Special Issue] Computers & Security, vol. 111. Dec. 2021. [Online]. Available: https://www.journals.elsevier.com/computers-and-security/call-for-papers/federated-learning-for-decentralized-cybersecurity

AI based Formal Specification for Scientific Security

Noah Evans
Sandia National Laboratories

Sam Pollard
Sandia National Laboratories

Robert Armstrong
Sandia National Laboratories

Jacob Hobbs
Sandia National Laboratories

October 14, 2021

1 Introduction

The correctness and security of science codes and the national cyber infrastructure is increasingly in doubt. The considerable investment already made in verification and validation (V&V) of scientific simulations can be extended in new directions to help address these needs. Building on traditional V&V to formally "correct by construction" [1] techniques in code development and coupling new AI techniques to discover security and correctness issues in existing codes is an opportunity to address these problems.

To make scientific computing secure we must first be able to understand its behavior. **We envision a future where – given a problem, a set of constraints, and a target – we leverage Machine Learning (e.g., adversarial ML, model building, reinforcement learning, genetic programming) and formal methods to infer the models, heuristics, tests, and/or policies that govern and optimize the computational science workflows needed for DOE and the NNSA's mission needs. This understanding will provide the foundation for the rigorous cyber-security guarantees on mission codes.** This need for automated workflows is increasingly urgent as V&V becomes the largest expenditure in computational science –Software must be secure as well as correct. To make rigorous levels of trust possible and V&V expenditures more manageable, our proposed workflow would analyze the behavior of existing scientific applications and use learned models to derive formal specifications of their behavior. At the compiler level, for example, the workflow could be used to learn heuristics for better code portability, while at the system level, the workflow could suggest directed cyber-security tests or more resilient resource management policies. Formal methods-based derivations based on learned system behavior models would enable a full-stack "Model Based System Engineering" (MBSE) workflow that provides greater assurance of scientific correctness and cyber-security in current and future systems and scientific applications.

2 Necessary Research

This combination of Machine Learning and Formal Methods techniques make it possible to understand and verify cyber-security guarantees on scientific applications.

2.1 New types of AI research for scientific cybersecurity

Effective application of "black box" Model Learning, used when analysis of source code is impractical, will require tens of thousands of input/output traces of the application under analysis. In cases where production source code is available, "white box" Model Learning may be used and the source code itself (and analyses derived thereof) comprises another data set. These artifacts will provide the foundation for the Model Learning data set. This data set will form the "System Under Learning" upon which we will apply active automata learning techniques such as Angluin's Algorithm to derive formal specifications for existing artifacts.

This research would lead to AI architecture for Model Learning of scientific applications and hardware behaviors from application source code and hardware artifacts.

2.2 New types of formal methods

Using these generated models it would be possible to manage the complexity of providing cybersecurity and privacy guarantees for scientific software using advances in automated theorem proving and proof assistants.

HPC computing applications have grown in complexity to both utilize the exponential increases in computing power as well as achieve the most efficient use of available hardware. Both of these lead to code which is nearly impossible for one person to understand its full behavior, which in turn makes verification and validation out of reach for most scientific computing teams.

Proof assistants such as Coq and automated theorem proving tools such as Satisfiability Modulo Theories (SMT) potentially make this complexity tractable by providing powerful automation and proof engineering tools to verify software security properties and to create programs which are provably correct to their specification.

However, the greatest strength and most difficult challenge of advanced theorem proving is that it makes no assumptions about a given system, scientific or otherwise, and so all behavior must be completely specified before a theorem prover will guarantee a proof is correct.

The work to provide the underlying foundation to prove scientific computing systems secure and correct is ongoing. Two relevant examples are those for real analysis [2] and floating-point arithmetic [3], though there are many such libraries in areas such as cryptography or other high-level mathematics.

References

[1] Andrew W Appel, Lennart Beringer, Adam Chlipala, Benjamin C Pierce, Zhong Shao, Stephanie Weirich, and Steve Zdancewic. Position paper: the science of deep specification. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 375(2104):20160331, 2017.

[2] Sylvie Boldo, Catherine Lelay, and Guillaume Melquiond. Coquelicot: A User-Friendly Library of Real Analysis for Coq, March 2015.

[3] Sylvie Boldo and Guillaume Melquiond. Flocq: A unified library for proving floating-point algorithms in coq. In Elisardo Antelo, David Hough, and Paolo Ienne, editors, *Proceedings of the 20th IEEE Symposium on Computer Arithmetic*, ARITH '11, pages 243–252, Tübingen, Germany, July 2011. IEEE Computer Society.

Algorithmic Development for Unreliable Computing Environments

Alyson Fox (fox33@llnl.gov), Colin Ponce, Chris Vogl, Jean-Paul Watson, Steve Chapin - Lawrence Livermore National Laboratory, Agnieszka Międlar – University of Kansas

Edge computing, also known as collaborative autonomy [1,2], is an emerging technology where networked, autonomous devices work collaboratively in applications such as human-machine search-and-rescue operations [2] or fully-decentralized smart grids [3] to achieve a common goal through a synergy of humans and machines. Wearable health devices, content delivery systems, and smart home devices are pushing applications, data, and services away from centralized environments and onto networks, closer to the requests, requiring on device data processing. By operating on data in-situ, edge computing avoids the cybersecurity concerns of first aggregating the data in high performance (HPC) or cloud computing (CC) facilities, e.g., theft and/or encryption of large databases. Additionally, the in-situ approach with either encryption or privacy-preserving techniques between devices [4,5] avoids the privacy concerns of centralized data aggregation, including restrictions on large databases of proprietary consumer and/or protected health data.

The nature of hardware in edge computing devices and the links between them make for an inherently unreliable environment, where frequent device and communication link failures are expected. Of course, the same device and communication link failures have been addressed in the HPC and CC communities [6,7], where strategies for fault detection, mitigation, and recovery typically rely on the devices maintaining redundant copies of data and/or executing redundant application algorithms. Such approaches are typically too costly to implement in edge computing environments where, as an example, the memory for a specific device may only be large enough to store a single copy of that device's data. Thus, the widespread use of edge computing by scientific computing applications is currently blocked until new fault detection, mitigation, and recovery strategies are devised that do not depend so heavily on redundant data storage and execution.

A different approach that seeks to minimize the amount of replication is an algorithm-based fault tolerance method (ABFT) or "algebraic and data-based detection" [6]. For edge computing, however, significant gaps in algorithm-level reliability remain. To our knowledge, there has been relatively little work done on developing algorithm-based fault tolerant techniques addressing the unreliable nature of current edge computing hardware and environments. Most of the existing algorithmic solutions designed/proposed for these unreliable environments have been based on solving decentralized consensus optimization problems. These methods have been very successful as they remove the bottleneck of synchronization points while also providing rejection criteria for data in the case the updates deviate too far from the expected value [8]. On the downside, such optimization methods can lack scalability, and not all applications can be easily reformulated in a suitable format. Additionally, if enough devices have been corrupted then obtaining a solution may no longer be feasible without incorporating additional resiliency measures. Current scalable algorithms in HPC applications tend to assume that the hardware is reliable and that redundancy and checkpointing prevent permanent corruption or loss of data. Thus, many compelling algorithms that have been successful in the HPC and CC environments cannot be directly adopted for edge computing. Although we believe that many practical and

LLNL-MI-827865

scalable algorithms developed primarily for HPC can be reformulated using resiliency techniques introduced for consensus optimization problems, there is a crucial need for new real-time computing concepts, e.g., resiliency methods that incorporate data privacy, data approximation, and randomization techniques. For algorithm-based fault tolerant techniques to be used in practice, we must consider several additional questions or implementation concerns for each resilient algorithmic building block:

(1) What levels of corruption and type of corruption can we tolerate and still guarantee a suitable solution?
(2) Can algorithms incorporate a sensitivity to data privacy, i.e., can algorithms adapt if certain portions of the data are not available, delayed, or if the data are encrypted?
(3) Is there a way to incorporate tunable parameters for the end users to trade-off resiliency and computational overhead?
(4) Can we theoretically quantify the trustworthiness of a delivered solution?
(5) What is the robustness of the system, i.e., the ability to withstand unexpected internal or external threats?
(6) Can multiple algorithms work in parallel to increase the robustness of the solution? For example, assuming a data imputation algorithm is the main workhorse for an application, could a detection method work in parallel, identifying corrupted data and providing information on how to incorporate this data without causing instabilities?
(7) How can we further incorporate research on nature-inspired (bio-inspired) cybersecurity and resiliency solutions?
(8) What mechanisms need to be incorporated within the algorithms such that they will be able to predict, detect, withstand, recover, or adapt to occurring disruptions?

This approach radically departs from HPC's current approach to reliable computing components and will require a sustained long-term investment in algorithmic research and software development.

References

[1] A. Chen. 2018. Building a Network of Collaborative Autonomous Machines. In Science & Technology, Research Highlights. Lawrence Livermore National Laboratory.
[2] V. Tzoumas. 2020. Algorithmics Foundations of Resilient Collaborative Autonomy. Chair's Distinguished Lecture, Department of Aerospace Engineering, The University of Michigan.
[3] V. Kekatos and G. B. Giannakis. 2012. Distributed Robust Power System State Estimation. IEEE T. on Power Syst.
[4] R. Hall, S.E. Fienberg, and Y. Nardi. 2011. Secure multiple linear regression based on homomorphic encryption. Journal of Official Statistics.
[5] A.F. Karr, X. Lin, A.P. Sanil, and J.P. Reiter. 2009. Privacy-Preserving Analysis of Vertically Partitioned Data Using Secure Matrix Products. Journal of Official Statistics.
[6] R. Canal, C. Hernandez, R. Tornero, A. Cilardo, G. Massari, F. Reghenzani, W. Fornaciari, M. Zapater, D. Atienza, A. Oleksiak, W. Piatek, and J. Abella. 2020. Predictive Reliability and Fault Management in Exascale Systems: State of the Art and Perspectives. ACM Computing Surveys.
[7] M.A. Mukwevho and T. Celik. 2021. Toward a Smart Cloud: A Review of Fault-Tolerance Methods in Cloud Systems. IEEE Transactions on Services Computing.

[8] Q. Li, B. Kailkhura, R. Goldhahn, P. Ray and P. K. Varshne. 2017. Robust Decentralized Learning Using ADMM with Unreliable Agents. ArXiv.

Understanding and Improving Collaborative Protected Data Workflows

Dan Gunter (dkgunter@lbl.gov), Keith Beattie (ksbeattie@lbl.gov), Ludovico Bianchi (lbianchi@lbl.gov)
Lawrence Berkeley National Laboratory

The connection between DOE funded work in basic science and commercial domestic entities that build physical infrastructure or products is important and broadly encouraged. One of the primary ways this connection is made is through direct and funded research collaboration between DOE grantees in national laboratories or universities and commercial companies. With respect to the computing aspects of this collaboration, often the most important contribution from the company is data and the understanding of how to interpret it. This data, in part or whole, is often considered proprietary and cannot be released into the public domain. Safeguarding the data is a key element in building and maintaining the commercial partners' trust, and thus an essential part of a productive and mutually beneficial relationship.

However, research teams that are funded by DOE science and applied offices can be ill-prepared to take on the responsibility for protecting datasets. We have written earlier [1] about the challenges of managing the workflow around protected assets in a distributed collaborative team. Researchers are accustomed by academic training and on-the-job experience to working with public data that can be freely emailed, added to GitHub repositories, shown on posters, etc. Both the design and business model of GitHub and other hosted code repositories encourage "open", as opposed to "closed", permissions; their fundamental ability to recover the full history of any object makes it difficult or even impossible to retract anything that has been accidentally made public. *The co-existence of open-source and "open science" practices for data and methods, with data and methods from external partners that are protected under legal agreements, is the defining challenge that we are addressing in this position paper.*

From our experience on several projects that involve commercial collaboration and protected data, we have come to believe that the key to success is a combination of social procedures (essentially project management) and technical infrastructure. The exact nature of this combination will depend on the team, the dataset, and the collaboration: for example, in the Materials Project[1] we constructed a shadow database that was accessible only to the team and the collaborator and managed it within the core infrastructure team; whereas on the Institute for the Design of Advanced Energy Systems (IDAES)[2] project, a larger and more distributed team with multiple independent industry collaborations, we stored all protected data in secure file-based storage and put each member through an onboarding process that described specific procedures for avoiding leaks. We are leveraging lessons learned from the latter approach for other projects in this space, modifying the approach in each case for the social dynamics and technical needs of the team and project.

While these efforts have been successful, there were also a number of remaining challenges. We feel that the tools and procedures we have developed have been too custom and experimental, given that similar challenges are found in many projects. The social aspects of the work are time-consuming and require specialized expertise, but neither we nor our funders fully appreciated this at the outset; nor do we have reliable methods to predict the scope of effort for new projects. And tool support for managing protected data is minimal — particularly when one considers the restriction that it must be accessible and user-friendly to commercial collaborators. Furthermore, tool support for integrating protected data securely

[1] Website: https://www.materialsproject.org/

[2] Website: https://www.idaes.org/

into the "normal" open-source coding workflow is virtually non-existent, placing undue reliance on training and individual behavior to avoid leaks. Given the potential of data leaks to break trust relationships and therefore have serious repercussions in both the short and long term, addressing these challenges will continue to be essential to successful collaborations in this space.

To address these needs more broadly and deeply, we believe that DOE/ASCR should consider research directions in two related areas. One research area would be socio-technical and geared towards understanding the roles, relationships, and effects of different kinds of training and incentives: which practices are the most likely to lead to information leaks, and how can we avoid these; how can the "onboarding" process be more standardized and streamlined across projects; and to lead these efforts, what kind of expertise do we as a community need to recruit or develop? This research area would study collaborative software engineering workflows involving protected data, evaluating their effectiveness in mitigating risks together with their overhead on the development process. A broad view of the landscape can help understand and predict future challenges, e.g., for larger and more complex datasets and AI/ML workflows. Another area of research would address the "tool gap" for automating detection, secure use, storage, and verification of protected data; particularly in coordination with the open-source development workflows that are ubiquitous in DOE science. Of course, these two research directions are related: understanding the workflows drives tool development, and better tools can fundamentally alter the workflows. It is our position that an investment in these research directions, properly combined with outreach, education, and training of the community, could significantly de-risk and streamline public/private collaborations across the DOE complex.

References:

[1] Keith Beattie and Dan Gunter. "Lessons learned working with protected assets in an open-source collaborative scientific software project." (2021). https://doi.org/10.1109/BoKSS52540.2021.00014

Contact author: Dan Gunter dkgunter@lbl.gov

Transformer-based Frameworks: Bridging the Gap from Machine Readable Data to Actionable Cyberdefense Insights

Mahantesh Halappanavar* Edoardo Serra† Alex Pothen‡

We posit that a significant amount of information currently exists that can be transformed into actionable insights to quantify the risk and exposure of cybersystems (such as High Performance Computing systems), and to enable near real-time decision making. Both the volume and streaming nature of cyber data make analysis challenging, and thus requiring innovation. We propose the construction of a multi-modal (foundational) framework to process data to enhance security and availability of computational resources for science. We demonstrate the utility of such a framework using the case study of textual information for software vulnerabilities, which when mapped to weakness enumerations and attack path enumerations lead to actionable information for rapid mitigation of existing and novel vulnerabilities.

Background and Challenges: A *weakness* is an architecture, design, or implementation error that occurs in cyberproducts (such as software, operating system, or hardware) that allows an unintentional and exploitable behavior of the product. A *vulnerability* is a set of one or more weaknesses in a specific cyberproduct that can be potentially exploited by an attacker for malicious operations. Unpatched vulnerabilities are a leading cause of cybersecurity incidents that result in significant economic damages to organizations. A large number of product-specific vulnerabilities are known, and new vulnerabilities are discovered each day by both attackers and defenders. For example, just for the year 2020, the United States Computer Emergency Readiness Team (US-CERT) reported about 17K newly discovered vulnerabilities (of which 4K were classified as high-severity, 10K were medium-severity, and 3K were low-severity).

In recent years, we have been witnessed an increasing number of attacks targeting High Performance Computing (HPC) systems, generally using newly discovered vulnerabilities. HPC technology is evolving rapidly, and the deployment is ubiquitous. Thus, the emergence of new vulnerabilities will both be frequent and in larger numbers than before. It becomes really important to recognize and mitigate such HPC vulnerabilities in a timely manner independent of the source of their discoveries and reporting. The NIST National Vulnerability Database (NVD) and Common Weakness Enumerations (CWE) provide a blueprint for understanding product flaws and their impact through a hierarchically designed dictionary of product weaknesses. NVD, in particular provides and collects structured data about vulnerabilities and connects them with common vulnerability scores (base, impact, and exploitability scores), weaknesses associate, attack patterns for various software and hardware products. NVD represents a landmark for security analysts to learn about existing vulnerabilities without having the burden of discovering them on their own.

However, before vulnerabilities are analyzed, scored, classified and published by NIST, about three-quarters of them are publicly reported online on news sites, blogs and social media pages, including hard to access parts of the web such as dark web, paste sites, and criminal forums. In the best cases, the lag between the time at which a new vulnerability is reported online and the time at which it is published in NVD is about seven days, generally much longer than that. In 2020, the estimated gap was at least 21 days. This delay occurs because humans perform each update manually to assure the quality of the information stored (i.e., they identify the new vulnerabilities, decide which unique code should be assigned to them, classify and score the vulnerability, etc.). A common attacker strategy rests on exploiting vulnerabilities not yet classified in existing repositories: attackers identify new vulnerabilities (zero-day vulnerability) and use that information to penetrate the first network containing that vulnerability. This situation places a greater onus on cyberdefense teams to overcome the lag and identify new vulnerabilities independently in a timely manner. In addition, repositories such as NVD do not indicate if the vulnerability can impact HPC systems. This makes the prioritization of these vulnerabilities and their mitigation even more onerous.

To further improve the timely diffusion of new threats/vulnerabilities, programs facilitating the sharing of such threats among many organizations are created. One of them is the program created by CISA for Automated Indicator Sharing (AIS). Such programs use industry standards such as Structured Threat Information Expression (STIX) (for cyberthreat indicators and defensive

*Pacific Northwest National Laboratory, Richland, WA (hala@pnnl.gov)
†Boise State University and Pacific Northwest National Laboratory (edoardoserra@boisestate.edu)
‡Purdue University, West Lafeyette, IN (apothen@purdue.edu)

measures information) and Trusted Automated Exchange of Indicator Information (TAXII™) (for machine-to-machine communications) facilitate the diffusion of such collaborative infrastructure. Even if these standards are well structured similarly to NVD, how organizations use these standards does not produce the same structural quality provided by NVD. Concerns of privacy and disclosures of institutional vulnerability are key reasons that hinder full disclose of critical vulebrability information. A first step towards overcoming some of these issues is exemplified by the V2W-BERT framework [1]

V2W-BERT Problem Statement: Common Vulnerabilities and Exposures (CVE) report has metadata and a short textual description containing information about the vulnerability. Common Weakness Enumerations (CWE) classes have sentences describing the nature of the weaknesses. Our goal is to map a given vulnerability description to the most relevant weakness descriptions organized in a hierarchy. The hierarchical relations start with a broader definition at the root level and proceed to specific definitions at the leaf level. Formally, we can express the problem as, $p = F_\omega(t_{cve}, t_{cwe})$, where the goal is to learn the trainable parameters ω of the function F, which takes CVE and CWE text descriptions (t_{cve}, t_{cwe}) as input, and returns the confidence probability of their association [1]. Since a link to the specific CWE at lower levels of hierarchy inherit the properties of their ancestors, we prepare positive training links for a CVE considering their ancestors, and the rest of the CWE classes are attributed to negative links and are used for learning ω

V2W-BERT Framework: The V2W-BERT architecture uses Siamese networks of transformer models to formulate links between CVEs and CWEs [1]. Figure 1 illustrates the general structure of the entire framework. V2W-BERT has two primary components: (i) Link Prediction (LP) to establish the links between a pair of CVE and CWE descriptions, and (ii) Reconstruction Decoder (RD), a masked language head on top of Transformer model to preserve the context. The entire model is trained in two steps. First, the Transformer model is pre-trained considering Masked Language Model (LM) without using any CWE class label information (highlighted in gray color in Figure 1). Second, we use the entire architecture to associate CVEs and CWEs while preserving the context learned in pre-training steps.

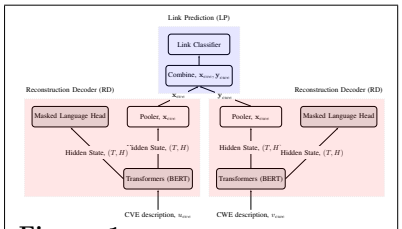


Figure 1: Architectural overview of the V2W-BERT framework. The Transformer component including the Reconstruction Decoder (RD, in red), and the Link Prediction (LP, in blue). Pre-training is done by Masking (in gray within the red boxes), but the entire framework use link prediction.

Generalizability: V2W-BERT, winner of the best application paper award at DSAA2021, represents a big step towards the creation of an autonomous system to handle vulnerabilities and provides actionable cyberdefense insights. However, there is a need to generalize and broaden the approach, especially for HPC resources. The framework can be extended to incorporate multimodal data in the form of text, images, tables, etc., from a variety of sources to perform a wide range of classification and prediction tasks (e.g., recognition of new vulnerabilities, estimation of impact scores, prediction of possible attack patterns, etc). Further, to enhance reliability and trustworthiness, an automated system should be interpretable and explainable. The system should focus on the needs of cyber analysts (e.g. creation of trustable confidence scores and intuitive explanation procedures for each specific automatic decision). Moreover, once this vulnerability knowledge base is created in a timely manner and updated by this automatic system, large scale tools helping HPC cyberdefense teams prioritize their mitigation strategies should be created (a preliminary example, which is neither large-scale nor focused on HPC, is [2]). Additional mechanisms to determine the safe amount of information that an organization can share about threats from current vulnerability knowledge should be provided to mitigate privacy concerns while sharing sufficient information for mitigative and preventative actions. We envision these future research directions are poised to enhance operational security and improved availability of mission-critical computational resources and scientific instruments of national significance.

References

[1] S. S. Das, E. Serra, M. Halappanavar, A. Pothen, and E. Al-Shaer, "V2W-BERT: A framework for effective hierarchical multiclass classification of software vulnerabilities," in *DSAA 2021, Porto, Portugal, October 6-9, 2021.* IEEE, 2021.

[2] E. Serra, S. Jajodia, A. Pugliese, A. Rullo, and V. Subrahmanian, "Pareto-optimal adversarial defense of enterprise systems," *ACM Transactions on Information and System Security (TISSEC)*, vol. 17, no. 3, pp. 1–39, 2015.

Trusting 5G and edge components for controlling instrumentation

Grant Johnson (grantj1@ameslab.gov), Ames Laboratory

Leveraging the 5G opportunity

With the increasing installation of 5G networks containing edge computing, the ASCR ecosystem will be asked to extend trust in the execution of complex workflows on service provider and participant wireless infrastructure adjacent to scientific instrumentation. To leverage this ecosystem for the ASCR vision of accelerating scientific discovery with complex workflows [1], dynamic control of scientific instrumentation will be coupled to decentralized digital twins and distributed scientific experts will remotely collaborate on data within these semi-trust wireless networks. Therefore, to trust these last mile environments, it is imperative to protect against attacks that impact the safety of operators and damage equipment as well as prevent the manipulation of scientific data inputs or results. This requires research to be pursued that secures communications within edge components as well as the intersection to ES NET, and to design for detection, monitoring, and forensics to be integrated into security operations from the beginning. Where the workflows are executed and who collaborated on data within wireless edge components will become a dynamic request that is self-optimized by newly designed intelligent schedulers between each requested execution [2]. To choose trustworthy infrastructure and reconstruct complex workflows, these newly coupled computational networks must trust in the authenticity of a data source, integrity of the data and commands, and enable traceability of workflows within the edge platforms for security operations.

Extending the ASCR capabilities to wireless scientific instrumentation enables coupling simulation, modeling, digital twins, and large-scale computing to drive controls of instruments in near real time. The communications amongst these decision-making nodes and the instrument must be secured for safe operations and uncompromised data integrity. This requires implementations for securing a diverse number of protocols for the purposes of ensuring privacy, integrity, authentication, and enforcing authorizations. Distributed ledger and blockchain technologies such as immutability and distributed consensus are an opportunity to secure federated communications by increasing the trust between organizations and decentralizing logic in a low-trust wireless environment. As research evolves in computational node security such as virtualization, trusted execution environments [3], and attack detection, linkage can be made of vulnerabilities and attack targets to the hardware and software context of node configurations. Therefore, platforms can provide the calculated risk of edge nodes and wireless network paths as an input for intelligent scheduler decisions. A platform scheduler can then optimize a complex workflow to execute only on trustworthy infrastructure. Decentralizing this decision making on the edge nodes manages risk as close to the data source as possible.

Finally, it is important at the platform design stage to include security operations as a critical stakeholder and incorporate their needs from the beginning. This enables integration of the detection, response, and recovery phases into security operations of government platform operators and research participants. This includes designing data collection and analysis to assist both human analysts and response automation. Platforms are required to collect data for forensics purposes, standardize on data formats and common information models, and agree on governance rules for detecting and responding to manipulated data or maliciously compromised computing nodes or networks. It will be important for platforms to automatically include collecting forensic records for workflow reconstruction and the context of the workflow execution such as what edge node provided computation, the versions of node hardware/software, and what networks the workflow traversed. Designing for detection and response will enable the community to reconfigure the computation and network based on trust, more rapidly respond to compromised experimental results, and to share attacker tactics, techniques, and procedures (TTPs) against the common environments.

Future research opportunities

Leveraging 5G for secure control of scientific instrumentation: Ames Lab is building on top of a fully configurable, advanced wireless testbed with edge computing to guide experiments on transmission electron microscopes. The enhanced wireless coverage in a configurable testbed presents the opportunity for basic security research at the intersection of a 5G edge ecosystem with local user facilities connected to ES NET and DOE HPC. Scientific instrumentation and edge computing comprises a diverse assortment of communication protocols and internal SDKs for edge execution. Platforms would benefit from research into a decentralized sub-system that assesses and tracks trustworthiness of data sets, nodes, and networks. Through the BLOSEM program with NETL [4], Ames Lab has deployed a blockchain network with HyperLedger Fabric. This project is exploring the strengths of blockchain technologies for managing device subcomponent configuration, traceability to vulnerabilities, and continuously confirming device integrity. A Platform research thrust can explore leveraging immutability, smart contracts, and distributed consensus: 1) maintain secure node identities and associated claims, 2) maintain hash-based integrity of sub-sets of experimental data inputs used in workflows, 3) record node configuration of subcomponents for traceability to vulnerabilities, 4) continuously confirm of device integrity (e.g., hardware based signatures), 5) assess risk and immutably maintain repositories of trusted nodes and network paths, and 6) decentralize automated response to a detected attack (e.g., reconfiguration, network isolation, revocation, etc.).

Designing platforms for analysis within Security Operations: To complement advancements of research in internal node security (e.g., virtualization and TEEs), research should focus on defining what should be recorded to reconstruct distributed execution of complex workflows within edge networks. This would include recording features that assist the human analyst or automated detection to more rapidly disposition an anomaly as malicious and have traceability for more rapid understanding of recovery steps. As basic security research advances for detecting malicious software or manipulated data, it is important for research to explore in parallel how to filter false positives and act on alerts. This includes having the right context in the required time to act for both human analysts and automation. Ames Lab has collaborated on relevant projects such as a big data platform framework for cyber analysis of operational technology data with PNNL and machine to machine cyber threat information sharing operated by ANL [5]. Research should be pursued for integrating into formats, information models, and tools used by security operations that express TTP(s) and attack information for the unique ASCR ecosystem. This includes research into: 1) integration into common monitoring, alerting, and visualization tools (e.g., SIEM), 2) extending common cybersecurity data formats and protocols for the unique ASCR threat environment (e.g., STIX, TAXII, and MITRE ATT&CK), and 3) creating communities and tools for sharing machine to machine, actionable data (e.g., MISP and CFM operated by ANL).

References

[1] B. Brown, A Vision for the ASCR Facilities Enterprise, Advanced Scientific Computing Advisory Committee, 2021, p. 2021.

[2] R. D. Friese, N. R. Tallent, M. Schram, M. Halappanavar and K. J. Barker, "Optimizing Distributed Data-Intensive Workflows," 2018 IEEE International Conference on Cluster Computing (CLUSTER), 2018, pp. 279-289, doi: 10.1109/CLUSTER.2018.00045.

[3] A. Akram, A. Giannakou, V. Akella, J. Lowe-Power and S. Peisert, "Performance Analysis of Scientific Computing Workloads on General Purpose TEEs," 2021 IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2021, pp. 1066-1076, doi: 10.1109/IPDPS49936.2021.00115.

[4] National Energy Technology Laboratory, "Blockchain for Optimized Security and Energy Management (BLOSEM)," [Online]. Available: https://netl.doe.gov/BLOSEM.

[5] Argonne National Laboratory, "Cyber Fed Model," [Online]. Available: https://cfm.gss.anl.gov/.

Federated Scientific Machine Learning

Patrick Johnstone (pjohnston@bnl.gov), Yuewei Lin (BNL), Shinjae Yoo (BNL)

Motivation and Use Cases

Federated Learning (FL) refers to an arrangement where a number of clients, each possessing their own data, engage a server with the goal of learning a single model trained on the collection of all client's data. Crucially, the clients are unable or unwilling to share their data directly with the server, usually for the following reasons: 1) to preserve privacy, 2) because the data are too big to move, or 3) to perform continual learning on local data. Thus centralized ML procedures are not available and the clients must collaborate with the server in order to train the model. Various scientific learning use cases require FL. As an example of case 1), nuclear safeguarding and biomedical data are two prime examples where organizations cannot share the data directly due to privacy concerns, but still want to learn from the pooled data from multiple sites/sources. An instance of case 2) occurs with the Earth System Grid Federation (ESGF) is an example where each grid participant has data but such a huge international collaboration of climate data would be too big to communicate and host on a single site. Finally, the SmartGrid is an example of case 3) where edge/fog computation requires updating the model with locally/regionally available data, so that the AI model is adapted to local environmental changes. Further, electricity usage on the distribution side of SmartGrid applications is sensitive data, necessitating FL solutions.

Challenges and Barriers

When we work on federated scientific learning use cases, there are several challenges we encounter. Here we list several FL challenges:

Non-IID Data A major challenge in FL is that the data are often not independent and identically distributed (IID) among different nodes/clients [1]. The most obvious violation of "identicalness" is that different nodes typically have different distributions of data due to different research focus, geographic locations, device specifications, operating conditions, etc. Also when different nodes have quite differing amounts of data, this can lead to biases in the final model towards nodes with more abundant data. "Independence" can also be violated, since participating nodes must satisfy a range of availability requirements (downtime, network availability etc.), hence data are not drawn independently.

The main challenge posed by non-IID data for optimization algorithms is that samples drawn at one node might not aid with learning a model suitable for a node with a completely different data distribution. This can lead to some of the client's models diverging from each other, inhibiting convergence of the overall FL algorithm.

Participating Node Availability At any given time, not all nodes or devices are typically available to participate in the learning process. For instance, some nodes in ESGF are not available all the time but when they are available, they can participate learning process. As another example, consider a smartGrid consisting of 10^7 consumers, smart appliances, and renewable suppliers. It would be highly impractical to enforce all of these clients to participate in every FL process, given communication and system outages, and the demands that would place on the client system. Thus, in FL, learning algorithms that require all nodes/clients to be available and participate at every round are rarely feasible.

Heterogeneity A primary challenge in FL is caused by differences in computational capabilities among the nodes and clients. In the case of the SmartGrid application, there is a wide discrepancy

among the device types such as μPMU (micro Phasor Measurement Unit) and SmartMeter, and their system capabilities are variably effected by the ambient conditions and background tasks, which could differ from micro-grid to micro-grid and from transmission to distribution. In the cases of biomedical data and ESGF, different participating organizations cannot be constrained to have identical computational facilities. In fact, they often have entirely different software stacks (eg: tensorflow vs PyTorch) if not differing software versions. From a learning point of view, this can lead to the so-called "straggler problem" where the server has to wait for the slowest clients in a round of computation. In some schemes, it is common to ignore the slowest workers updates, however this adds a bias in update selection towards the faster clients.

Communication Costs ESGF participants span the globe and are connected via cross-continent internet connections. In the case of nuclear safeguard, there are also a mixture of connection speeds to the safeguard control agency due to the nature of the international community. In SmartGrids, the upload speeds at the devices are typically far slower than download speeds, placing the bottleneck on the client-to-server update.

Future Research Directions

To address the above research challenges, we propose three major future research directions:

Investigate Operator Splitting Methods These methods may provide a principled way to handle the non-IID data case and have only recently been introduced as potential FL solutions [2]. In these methods, the updates performed by each client are regularized in such a way that global consensus can be reached asymptotically despite client distribution differences. Further, clients are allowed to perform different amounts of work depending on system heterogeneity. However, there are still many open problems around such methods including how many local steps to perform, adequate stopping criteria for local iterations, and relaxing the requirement that all clients participate in every round.

Asynchronous Optimization Asynchrony allows client-server communications to happen at any time and removes the requirement of waiting for all the client updates before continuing to the next round. However, asynchrony has been shown to degrade model accuracy when used in many algorithms. Thus there is a need to develop new asynchronous FL methods that are robust to the computational delays incurred by asynchrony.

Reduced Communication Learning By this, we mean methods that do not require rounds of communication as often as benchmark methods such as synchronous SGD. This is typically achieved by increasing the number of local updates performed by the clients. While this gives an immediate savings in communication, it may effect convergence, especially in the non-IID data setting. Thus care is needed to develop new methods that can allow for many local updates without adversely effecting convergence and final model performance. There could be alternative strategies to reduce communication cost, such as gradient and model compression, and the community could develop various complementary strategies.

References

[1] Peter Kairouz et al. "Advances and open problems in federated learning". In: *arXiv preprint arXiv:1912.04977* (2019).

[2] Reese Pathak and Martin J Wainwright. "FedSplit: An algorithmic framework for fast federated optimization". In: *arXiv preprint arXiv:2005.05238* (2020).

Secure and Private End-to-End Provenance Capture and Verification for Multi-party Data Workflow

Taeho Jung, Boris Glavic, Alexander Rasin

Introduction and background Given humanity's continuously increasing rate of data production and the increasing reliance of decision making processes on data, it is no longer possible for a single party to just work with the data it generates itself. Thus, parties have to resort to data sharing / trading to fulfill their data needs using, e.g., payed data platforms (e.g., Zoominfo with 20,000+ customers including Google, Zoom, Amazon, and SAP Concur; Bombora with 6,000+ customers including LinkedIn, Salesforce, and Adobe; and many others), mutual data exchange, or free sharing with attribution. However, sharing of sensitive information (e.g., business secrets or personally identifiable data) across parties results in increased risks. According to a study by IBM [10], a staggering 29.6% of companies have experienced data breaches in 2019, where the breaches involving the *3rd-party data sharing* was the highest cost amplifier that contributed to the average cost of $3.92M per breach. Unfortunately, there are currently no mechanisms for monitoring the workflow of data shared with multiple parties in a decentralized manner (i.e., without a trusted central authority).

Limitations of State-of-the-art Research Provenance capture throughout multi-party data workflow must be performed without accessing raw datasets/logs of parties (for privacy of each party), be robust against malicious parties, guarantees integrity of collected provenance, and be efficient enough. These requirements result in several critical research challenges that cannot be addressed with existing technologies in the literature. Accurately tracking provenance across multiple parties including adversaries is an extremely challenging open problem that requires novel research. For the data privacy of Since data is secured via all-time-encryption, even the cloud cannot perform arbitrary logging due to the high cost associated with executing computations directly on encrypted data. Integrity of provenance storage can be achieved through consensus mechanisms such as blockchains, smart contracts, or TEE [5, 11, 13, 15], but these approaches do not consider malicious parties that may tamper with the provenance capture process. Past work on protecting provenance collection against adversaries [1, 7, 8, 14] (i) assumes a high level of control over each networking device (i.e., centralized) and often (ii) incurs prohibitively large overhead, which is impractical in our scenario. Finally, existing works for secure and private data sharing either focus on the access control [6, 9, 12] or focus on the fair exchange of digital token (e.g., cryptocurrency) and the data [2–4]. Efficient and privacy-preserving techniques for capturing the provenance of data shared with third parties do not exist.

Proposed Research Topics There is an imminent need for a secure and private decentralized environment for monitoring data workflow among multiple parties with the following properties even when some parties are malicious: data owners are able to (1) track their shared data across parties without accessing others' raw logs of activities and data and without disclosing their own logs/data to others; (2) spend their limited resources on verifying the records collected from other

potentially malicious parties; and (3) monitor the provenance records of data to determine whether their data usage/access policies are respected by other parties during the workflow.

It is extremely challenging to have such guarantees because we do not allow anyone (even the administrators who maintain the infrastructure/servers hosting the data) to access parties' raw datasets / logs and there exist malicious parties. Because we consider the data is shared in a decentralized environment without a central broker, the monitoring and provenance capture need to rely on the records generated and provided by the parties who are potentially malicious. Malicious parties may want to hide part of the provenance records or provide incorrect provenance records, e.g., to be able to hide an illegal operations on a shared dataset that violates the policies the owner has set for this dataset. Furthermore, we cannot require parties to provide a full account of all of their internal operations due to the overhead issues and, thus, global provenance graphs will be incomplete. The presence of incomplete and possibly incorrect provenance necessitates the development of techniques for detecting missing and spoofed provenance, i.e., *provenance verification*. This must be done in an efficient, privacy-preserving, and secure manner without placing unrealistic constraints on the operations of parties in the multi-party data workflow.

References

[1] Peter Buneman, Sanjeev Khanna, and Wang-Chiew Tan. Data provenance: Some basic issues. In *FSTTCS*, pages 87–93. Springer, 2000.

[2] Sergi Delgado-Segura, Cristina Pérez-Solà, Jordi Herrera-Joancomartí, and Guillermo Navarro-Arribas. Bitcoin private key locked transactions. *Information Processing Letters*, 140:37–41, 2018.

[3] Sergi Delgado-Segura, Cristina Pérez-Solà, Guillermo Navarro-Arribas, and Jordi Herrera-Joancomartí. A fair protocol for data trading based on bitcoin transactions. *Future Generation Computer Systems*, 107:832–840, 2020.

[4] Stefan Dziembowski, Lisa Eckey, and Sebastian Faust. Fairswap: How to fairly exchange digital goods. In *CCS*, pages 967–984, 2018.

[5] Kai Fan, Shangyang Wang, Yanhui Ren, Hui Li, and Yintang Yang. Medblock: Efficient and secure medical data sharing via blockchain. *Journal of medical systems*, 42(8):136, 2018.

[6] Chaosheng Feng, Keping Yu, Ali Kashif Bashir, Yasser D Al-Otaibi, Yang Lu, Shengbo Chen, and Di Zhang. Efficient and secure data sharing for 5g flying drones: a blockchain-enabled approach. *IEEE Network*, 35(1):130–137, 2021.

[7] Ragib Hasan, Radu Sion, and Marianne Winslett. The case of the fake picasso: Preventing history forgery with secure provenance. In *FAST*, volume 9, pages 1–14, 2009.

[8] Ragib Hasan, Radu Sion, and Marianne Winslett. Preventing history forgery with secure provenance. *TOS*, 5(4):12, 2009.

[9] Yuncong Hu, Sam Kumar, and Raluca Ada Popa. Ghostor: Toward a secure data-sharing system from decentralized trust. In *NSDI*, pages 851–877, 2020.

[10] IBM. How much would a data breach cost your business?

[11] Jiawen Kang, Rong Yu, Xumin Huang, Maoqiang Wu, Sabita Maharjan, Shengli Xie, and Yan Zhang. Blockchain for secure and efficient data sharing in vehicular edge computing and networks. *IoT-J*, 2018.

[12] Imran Makhdoom, Ian Zhou, Mehran Abolhasan, Justin Lipman, and Wei Ni. Privysharing: A blockchain-based framework for privacy-preserving and secure data sharing in smart cities. *Computers & Security*, 88:101653, 2020.

[13] Riccardo Paccagnella, Pubali Datta, Wajih Ul Hassan, Adam Bates, Christopher W Fletcher, Andrew Miller, and Dave Tian. Custos: Practical tamper-evident auditing of operating systems using trusted execution. In *Proc. of the Symposium on Network and Distributed System Security (NDSS)*, 2020.

[14] Yogesh L Simmhan, Beth Plale, and Dennis Gannon. A survey of data provenance in e-science. *SIGMOD Record*, 34(3):31–36, 2005.

[15] Qi Xia, Emmanuel Boateng Sifah, Kwame Omono Asamoah, Jianbin Gao, Xiaojiang Du, and Mohsen Guizani. Medshare: Trust-less medical data sharing among cloud service providers via blockchain. *IEEE Access*, 5:14757–14767, 2017.

Secure, Robust, and Accountable Federated Learning using Trusted Execution Environments

Murat Kantarcioglu, University of Texas at Dallas [1]

1. Overview: Federated Learning (FL) has been introduced as a distributed machine learning protocol. Through FL, a set of agents can collaboratively train a model without sharing their data with each other, or any other third party, and only sharing updates to the model at each round. This makes FL suitable for settings where data privacy is desired, especially when the model update aggregation is done securely. Such secure aggregation could be achieved in practice using trusted execution environments (TEEs). Using hardware supported TEEs, each agent can send their encrypted updates to a TEE running in the cloud, and the TEE can decrypt the updates, aggregate the updates, and send the encrypted model update to each agent. Compared to existing cryptographic approaches, TEEs can provide efficient aggregation assuming that the hardware is trusted. Furthermore, with recent developments like the Microsoft Confidential computing cloud, TEE-based solutions can be deployed in commercial cloud computing infrastructure easily.

Unfortunately, secure aggregation using TEEs is not enough to protect against all attacks. FL has been shown to be susceptible to *backdoor attacks* (e.g., [1]). In a backdoor attack, an adversary modifies the training process to make the model learn a *targeted misclassification functionality*. In FL, since the data is decentralized, the backdoor attacks are typically carried through *model poisoning*. That is, the adversary tries to construct a malicious update that encodes the backdoor in such a way that, when it is aggregated with other agents' updates, the aggregated model exhibits the backdoor. Therefore, straightforward execution of FL could be vulnerable to attacks.

Several works try to prevent such backdoor attacks by designing robust aggregation functions (e.g., our prior work [1]). Still, there is no guarantee that existing defenses will succeed in defending against any type of adversary. Therefore, we also need to make FL model training *accountable* to discourage attackers and determine later how the attack happened. That is, *even if an attack is not prevented during training, we may want to detect and penalize the adversarial agents later at a time when the backdoor is found in the trained model*. This could be important to understand how the attack happened in the first place.

The above observations suggest that an FL framework needs to securely aggregate the updates sent by the agents, be robust against various attacks such as backdoor attacks, and provide accountability. Below, we discuss our proposed framework that addresses all these important FL framework requirements.

2. The proposed FL architecture: To make FL secure, robust, and accountable, we leverage *private blockchains* since they are compatible with the decentralized nature of FL, provide practical immutability and Turing-complete computation on the logged data via smart contracts. In this framework, the private blockchain that supports TEEs (i.e., Hyperledger Fabric with TEE support) performs the aggregation inside a smart contract running on a TEE node, stores appropriate encrypted logs about the updates on the chain, and sends the global updates back to the agents.

2.1. Robust aggregation for FL: The aggregation function running inside the TEE on the blockchain is carefully chosen to be robust against poisoning attacks. For example, our prior work [1] on robust aggregation against poisoning attacks could be used as the aggregation function.

[1]Contact: muratk@utdallas.edu

In different use cases, we may need to leverage different aggregation functions. For example, the signSGD aggregation function that requires sending of one bit per gradient could be used for communication efficiency and privacy protection. Also, for efficiency purposes, aggregation can be distributed to multiple nodes on the blockchain depending on the number of available TEEs.

2.2. Log storage and analysis on blockchains: During each aggregation round, TEEs doing the aggregation can log the updates sent by different agents. These logs later on, when an attack is detected, could be analyzed to detect which updates caused the backdoor. In our previous work [2], we show that such logs stored on the Hyperledger Fabric-based blockchain could be analyzed for detecting the agent that launched the attack.

3. Research Challenges : The above proposed architecture leverages private blockchains and TEEs to provide basic security, robustness, and accountability for FL.

Still, there are other challenges that need to be addressed. In the above framework, each agent still sees the aggregated model update after each round. These updates could leak some sensitive information. Some prior work proposed adding noise to the updates. Another approach we will explore is to run at least some part of the local update inside a TEE as well. In this extension, each agent will have a TEE running on their local machines. Since running the entire local learning process inside TEE will be too slow, we hypothesize that running at least a few iterations combined with local data and noise would be enough to prevent many privacy attacks based on the model updates sent at each round. Therefore, more research is needed to understand the privacy impact of the model updates sent in each round.

As we discussed above, for different use cases, different FL aggregation functions may be needed. Robust aggregation functions (e.g., our work [1]) require more sophisticated update aggregation functions. We cannot just implement these different aggregation functions inside TEE and hope that they will be secure. It has been shown before that TEEs can leak important information due to their memory access patterns (see our prior work on this leakage and how to prevent it using oblivious execution [3]). Therefore, more research is needed to implement robust FL aggregation functions securely inside TEE without leaking memory access patterns.

References

[1] M. S. Özdayi, M. Kantarcioglu, and Y. R. Gel, "Defending against backdoors in federated learning with robust learning rate," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*. AAAI Press, 2021, pp. 9268–9276. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17118

[2] H. B. Desai, M. S. Özdayi, and M. Kantarcioglu, "Blockfla: Accountable federated learning via hybrid blockchain architecture," in *CODASPY '21: Eleventh ACM Conference on Data and Application Security and Privacy*, A. Joshi, B. Carminati, and R. M. Verma, Eds. ACM, 2021, pp. 101–112. [Online]. Available: https://doi.org/10.1145/3422337.3447837

[3] F. Shaon, M. Kantarcioglu, Z. Lin, and L. Khan, "Sgx-bigmatrix: A practical encrypted data analytic framework with trusted processors," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, 2017, pp. 1211–1228. [Online]. Available: https://doi.org/10.1145/3133956.3134095

Formal Methods-based Certification Frameworks for Scientific Computing Applications

Ariel Kellison [1,2], Geoff Hulette [2], John Bender [2], Samuel D. Pollard [2], and Heidi K. Thornquist [2]

[1]Cornell University, Ithaca, New York
[2]Sandia National Laboratories, Livermore, California

October 14, 2021

A significant barrier to certifiable, trustworthy computational science at the Department of Energy is the prevalent belief that proofs of accurate correspondence between an implementation and a desired conceptual model along with its solution are intractable. The current verification paradigm for computational science at the DOE therefore relies on the *absence of proof of incorrectness* [7]; this paradigm is woefully behind common verification practices employed in industry and academia, and ultimately means that national security and policy decisions based on computational models lack the highest assurances possible.

To address the absence of formal guarantees of correctness of critical computational models and simulations across DOE labs, we suggest the development of end-to-end formal methods-based certification frameworks for scientific computing applications. These frameworks can bridge the gap between conceptual models and their corresponding high-performance implementations.

Numerous examples from academia and industry provide strong evidence suggesting that the development of such frameworks is possible, and that these frameworks can be used to provide correctness guarantees for large and complex systems. Below, we provide some details on formal methods-based certification frameworks relevant to each of the three areas of verification [7] fundamental to trustworthy computational science.

Numerical algorithm verification

The current method employed for numerical algorithm verification across Department of Energy labs relies on the accumulation of evidence from test cases. The validity of this evidence rests on the assumption of a representative set of test cases and aims to prove that the implementation behaves according to an informal specification of correct behavior. In contrast, formal-methods based approaches that are extensively used in industry [6] enable the development of mathematically precise, verifiable program specifications. Furthermore, machine-checkable proof certificates of the correctness and numerical accuracy of imperative implementations of numerical algorithms have been demonstrated [1, 8].

Software Quality Assurance

The complexity of computational science models and simulations increases the likelihood of human errors and leaves the software development cycle vulnerable to adversarial agents. Before trusting

the results of critical models and simulations, we need to establish extremely high confidence that each component of the system on which the simulation relies, from hardware to software, is correct. To that end, formal methods used in industry enable the precise definition of program behavior, and ensure that records of expected behavior are consistent throughout the software development stack [5, 6]. Fully verified software stacks that account for the behavior of underlying hardware have been demonstrated [4]. Furthermore, recent work has demonstrated that machine-checkable proof certificates of correctness for small system components can be composed to ensure the correctness of larger complex systems [3].

Solution Verification

Given a set of continuous ODEs or PDEs, solution verification entails a quantitative study of how accurately a discrete numerical implementation represents the original set of continuous equations and its qualitative behavior. Popular opinion in the computational science community is that formal mathematical proofs of compatibility between a numerical implementation and a continuous model are intractable in practice [7]. However, demonstrations of such proofs have appeared in the literature [2]. Notably these proofs come equipped with machine-checkable proof certificates that can be integrated with the methods proposed for numerical algorithm verification and software quality assurance.

References

[1] Andrew W. Appel and Yves Bertot. C floating-point proofs layered with VST and Flocq. *Journal of Formalized Reasoning*, 2020.

[2] Sylvie Boldo, François Clément, Jean-Christophe Filliítre, Micaela Mayero, Guillaume Melquiond, and Pierre Weis. Trusting Computations: A Mechanized Proof from Partial Differential Equations to Actual Program. *Computers and Mathematics with Applications*, 2014.

[3] Ronghui Gu, Zhong Shao, Hao Chen, Xiongnan Wu, Jieung Kim, Vilhelm Sjöberg, and David Costanzo. CertiKOS: An Extensible Architecture for Building Certified Concurrent OS Kernels. In *USENIX Symposium on Operating Systems Design and Implementation*, OSDI'16, 2016.

[4] Chris Hawblitzel, Jon Howell, Jacob R. Lorch, Arjun Narayan, Bryan Parno, Danfeng Zhang, and Brian Zill. Ironclad Apps: End-to-End Security via Automated Full-System Verification. In *USENIX Symposium on Operating Systems Design and Implementation*, OSDI'14, 2014.

[5] K. Rustan M. Leino. Dafny: An Automatic Program Verifier for Functional Correctness. In *Logic for Programming, Artificial Intelligence, and Reasoning*, 2010.

[6] Chris Newcombe, Tim Rath, Fan Zhang, Bogdan Munteanu, Marc Brooker, and Michael Deardeuff. How Amazon Web Services Uses Formal Methods. *Commun. ACM*, 2015.

[7] William L. Oberkampf and Christopher J. Roy. *Verification and Validation in Scientific Computing*. Cambridge University Press, 2010.

[8] Tahina Ramananandro, Paul Mountcastle, Benoît Meister, and Richard Lethin. A Unified Coq Framework for Verifying C Programs with Floating-Point Computations. CPP 2016, 2016.

Reliable Privacy-Preserving Peer-2-Peer Distributed Reinforcement Learning for Science (RP4RL)

Sami Khairy (corresponding author, skhairy@anl.gov), Argonne National Laboratory
Prasanna Balaprakash, Argonne National Laboratory

Research Motivation. Reinforcement learning (RL) is a machine-learning paradigm for data-driven sequential decision making under uncertainty. By automating the design, control, and optimization of complex processes, it has the promise to accelerate scientific discovery in several DOE science domains such as smart energy, material science, drug discovery, and quantum computing. RL algorithms however require data samples on the order of hundreds of millions to solve a control task, which is infeasible in scientific applications where data generation is computationally expensive and time-consuming. Developing scalable distributed data-parallel RL algorithms capable of generating high-throughput data is necessary to accelerate training.

Research Challenge. Modern distributed RL architectures, such as IMPALA [2] and SEED-RL [1] seek to massively scale RL by decoupling the learning logic from data generation. The general architecture is that of a controller-agent in which thousands of agents, possibly in remote geographical locations, interact in parallel with independent instances of the same environment to speed up data generation. Once an agent is done collecting experience, raw trajectories in the form of a sequence of (states, actions, rewards) tuples are sent to the central controller where policy parameters are updated using a gradient-based learning algorithm. It has to be emphasized that sending raw experience enables stable off-policy learning and is necessary to support asynchronous model updates which is critical for compute resource utilization [2]. This is in contrast to traditional privacy-preserving federated learning architectures in which processed noisy gradient updates are sent to the central controller [3]. From a ***privacy standpoint***, highly-scalable modern RL architectures introduce serious privacy concerns as local data is transported over communication links through the cloud to a central controller. If a malicious eavesdropper taps into a communication link and gains access to the raw trajectories, the internal logic of the RL environment, which can be a digital twin of a sensitive system such as a nuclear power plant or nation's energy grid, will be exposed. This is because raw trajectories can be used to 1) deduce the control policy, 2) build a surrogate model which mimics the behavior of the simulator, and 3) reverse engineer the optimization objective through inverse RL. From a ***cybersecurity standpoint***, the central controller is a single point of failure that demeans the system's reliability. If the controller fails due to a security threat, the whole system will breakdown and remote agents will not be able to adapt to the ever-changing dynamics of the environment. Therefore, it is necessary to design a new, reliable and privacy-preserving, distributed RL architecture that can meet the requirements of scalable data-driven control of scientific applications without introducing privacy or security risks.

Research Directions. We envision a reliable, privacy-preserving, peer-2-peer (P2P) distributed RL architecture (RP4RL) as shown in Figure 1. In this architecture, there are N learner-actor agents connected using a P2P topology through 5G's ultra reliable low latency communication network (URLLB). Learning and data generation are coupled in each of the agents to achieve reliability and preserve privacy. In addition, there is a tracker software in the cloud that can be thought of as a virtual critical section with a semaphore to control concurrency. The function of the tracker is to track the identity of the agents in possession of the L most recent policy models by maintaining a list $\mathcal{L}$, $(L < N)$. Each agent in the P2P network collects experience asyncrhonously based on its own policy model. For ease of exposition, suppose agent A is done collecting environment experience. Agent A will query the tracker to retrieve the identity of an agent in $\mathcal{L}$. At

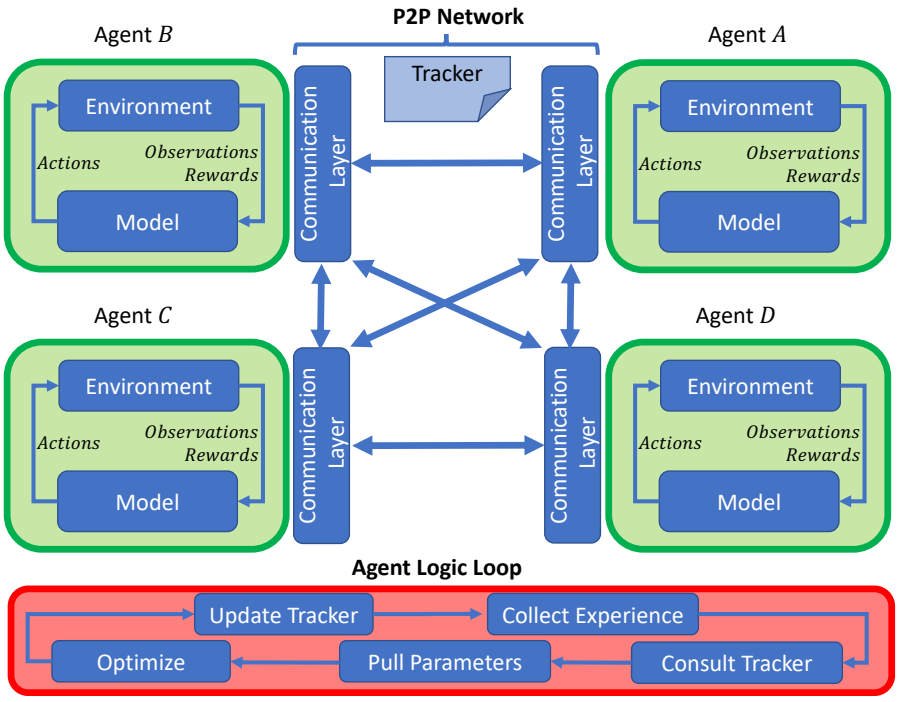


Figure 1: Architecture of the proposed RP4RL for science

this point, the tracker's list, $\mathcal{L}$, may be empty because there are already L agents in the process of model update. In this case, agent A joins a waiting queue until one of the L agents is done with model update and render itself accessible for other agents by being admitted to $\mathcal{L}$. Otherwise, the tracker will provide the identity of one agent from $\mathcal{L}$ to agent A, say agent B, and agent B will be removed from $\mathcal{L}$. Next agent A will request model parameters from agent B and optimize model parameters in an off-policy approach using the V-trace algorithm [2], which yields a new improved policy compared with that of agent B's. Agent A will then notify the tracker to be added to $\mathcal{L}$. Consequently, model parameters are communicated between agents only when necessary. Model parameters can be protected using the f-differential privacy framework in which gaussian noise is added to the gradients to achieve privacy without sacrificing performance [4]. By design, this architecture is ***privacy-preserving*** since local data remains completely private at the agent's site. This architecture is also ***reliable*** in the sense that the system will remain functional even if one of the agents is taken down due to a cyber threat. By joining the federation, all the agents in the P2P network cooperate to learn the optimal policy faster than any one can do on its own, and modern scalability can be achieved while privacy is preserved.

References

[1] L. Espeholt, R. Marinier, P. Stanczyk, K. Wang, and M. Michalski. Seed rl: Scalable and efficient deep-rl with accelerated central inference. *arXiv preprint arXiv:1910.06591*, 2019.

[2] L. Espeholt, H. Soyer, R. Munos, K. Simonyan, V. Mnih, T. Ward, Y. Doron, V. Firoiu, T. Harley, I. Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International Conference on Machine Learning*, pages 1407–1416. PMLR, 2018.

[3] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[4] Q. Zheng, S. Chen, Q. Long, and W. Su. Federated f-differential privacy. In *International Conference on Artificial Intelligence and Statistics*, pages 2251–2259. PMLR, 2021.

ASCR Workshop on Cybersecurity and Privacy for Scientific Computing Ecosystems

Position Paper: Challenges in Algorithm Design for Privacy-Preserving Federated Learning

Kibaek Kim[1] (kimk@anl.gov), Minseok Ryu[1], Youngdae Kim[1], Zichao Wendy Di[1], Sven Leyffer[1], and Ravi Madduri[2]

[1]Mathematics and Computer Science, [2]Data Science and Learning, Argonne National Laboratory

Federated learning (FL) is a machine learning (ML) setting where multiple entities (clients) collaborate in training a ML model, under the coordination of a central server or service provider. An important DOE application are cybersecurity federated models (e.g., Argonne's CFM [1]) that are community-based security models to promote a global defense against common threats within large heterogeneous and distributed organizations such as the DOE science complex (e.g., national labs, computing facilities, universities, data centers). Such models can be enhanced by recent advances in FL approaches by adding a common threat-detection capability without jeopardizing the privacy of the individual members. In this position paper, we discuss challenges and opportunities arising from the interdisciplinary nature of FL, when developing advanced algorithms for privacy-preserving FL.

Key Challenges. The problems and applications considered in FL are inherently *interdisciplinary*, requiring techniques from distributed optimization, statistics, differential privacy, cryptography, security, as well as ML. An important challenge is the *composability* of FL algorithms with other techniques. For example, an FL algorithm implementation typically consists of a distributed optimization algorithm (e.g., FedAvg), data characteristics/modalities, communication strategy (e.g., network topology and protocol, compression), a privacy technique (e.g., differential privacy, secure computation). Each combination of different techniques is likely to result in significantly different algorithmic behaviors such as convergence and learning accuracy.

Some of the key challenges also arise from each technique necessary for successful adoption of FL algorithms. The first challenge is *preserving the privacy of model and data*. While providing some level of privacy to the learning agents by decentralizing the training data, FL itself does not provide any privacy guarantees against malicious adversaries, which necessitates privacy techniques for cybersecurity models, as well as other applications (e.g., [2]). The second challenge is that data can be *unbalanced and non-IID* (identically and independently distributed) across the clients. The non-IIDness can make the model training challenging particularly in cross-device setting (e.g., millions of sensors and edge devices), where a subset of clients may be sampled at every iteration of a global update, implying that the underlying distributions change at every step of the algorithm. The other challenge is *communications* required for training a model across the clients. FL with data distributed in a few large silos (i.e., cross-silo FL) may experience computational load imbalance across the clients, due to which some local model trains may take more time than the others. On the other hand, in the case of cross-device FL, a large number of devices may suffer from communication bottleneck with limited network capacities. Moreover, in both cross-silo and cross-device settings, some clients may be temporarily unavailable or fail during the training.

State of the Art. Secure computations (e.g., secure multi-party computation, secure enclaves) do not interfere with FL model accuracy as the functions are evaluated on distributed inputs from the clients securely with strong privacy guarantee. However, this technique assumes an honest secure server and can be computationally expensive or infeasible in cross-device settings. Differential privacy is the state-of-the-art privacy technique for quantifying and limiting information disclosure by random perturbation that is supported by theoretical justifications [3]. Depending on how the random perturbation is applied, the FL algorithm needs to be designed carefully to balance the trade-off between the privacy and model accuracy (i.e., utility), which are typically unknown

before training. The convergence analysis techniques of FL algorithms for the IID case have been simply extended and applied to the non-IID case under various assumptions (e.g., uniform client sampling). It is still largely unknown whether the convergence/performance can be further improved by exploiting the characteristics of non-IID data (e.g., [4]). Moreover, most algorithms are limited to some variants of the federated averaging algorithm, and it is not clear how other algorithms such as alternating direction method of multipliers [3] can be applied with the client sampling due to the stateless nature. A small number of FL frameworks are actively being developed, but most of them are in their early stages with limited capabilities [5]. In the cybersecurity domain, the application of FL approaches has been recently applied to malware classification [6], where none of the challenges discussed above was addressed.

New Research Directions. We propose new research directions in privacy-preserving FL with the application of federated cyberthreat detection systems. An important research question to answer is: how do we design and develop effective and efficient FL frameworks with the capability of easily composing multiple techniques from interdisciplinary fields? Key requirements include identifying essential algorithmic components and characterizing their mathematical and computational properties. In addition to the key algorithmic components described above, other important components include identification of bias in training data, verifiable computation, heterogeneous computing resources, and (inference) attack models. We conclude this paper with a list of critical research directions for FL framework and algorithms:

- a composable FL framework design, where each technical component can be plug-and-play, allowing a heterogeneous set of clients such as national labs and universities to participate,
- enable optimal design of centralized and decentralized network topology, communication frequency, and compressed communications,
- data reduction technique and latent space representation for information integration to avoid data/model sharing and to enable customization to heterogeneous clients,
- privacy-preserving techniques (e.g., differential privacy, secure computation, and their hybrid) to prevent malicious actors from gaming the cybersecurity model,
- numerical verification of the privacy of client models and data against malicious attacks,
- development of federated threat detection and sharing within individual organizations and across organizational boundaries.

References

[1] "Argonne National Laboratory's Cyber Fed Model: Improving Energy Cybersecurity," Accessed in 2021. https://www.anl.gov/sss/cyber-fed-model-improving-energy-cybersecurity.

[2] U.S. Department of Energy, "Bridge2AI and Privacy-Preserving Artificial Intelligence Research." DOE National Laboratory Program Announcement Number: LAB 21-2502, 2021.

[3] M. Ryu and K. Kim, "Differentially Private Federated Learning via Inexact ADMM," *arXiv preprint arXiv:2106.06127*, 2021.

[4] X. Zhang, M. Hong, S. Dhople, W. Yin, and Y. Liu, "FedPD: A Federated Learning Framework with Adaptivity to Non-IID Data," *IEEE Transactions on Signal Processing*, pp. 1–1, 2021.

[5] T. R. Gadekallu, Q.-V. Pham, T. Huynh-The, S. Bhattacharya, P. K. R. Maddikunta, and M. Liyanage, "Federated Learning for Big Data: A Survey on Opportunities, Applications, and Future Directions," *arXiv:2110.04160*, Oct. 2021.

[6] K.-Y. Lin and W.-R. Huang, "Using federated learning on malware classification," in *2020 22nd International Conference on Advanced Communication Technology (ICACT)*, pp. 585–589, IEEE, 2020.

Challenges with Sensitive Data in Distributed Graph Settings

Olivera Kotevska*, Chris Stanley, J. Robert Michael, Bill Kay,
Anand Sarwate[$], Ramki Kannan, Gina Tourassi
Oak Ridge National Laboratory,[$] Rutgers University
Corresponding author: kotevskao@ornl.gov

Challenge

Data availability faces distinctive challenges in DOE environments on applications such as life sciences, critical energy infrastructure sciences, and national security. It is in the interest of federal agencies such as DOE, NIH, and CDC to make communal use of the data that cannot be moved or revealed due to privacy concerns. Some of these data are relational in nature and are thus readily represented as graphs. For example, in computer security, the network of user interactions for a particular platform can naturally be viewed as a graph whose nodes are users and whose edges represent client-to-client interactions. Another example would be social network graphs, whose nodes are users and edges represent relationships. In each case, the node contains private user information to be protected, while the natural encoding scheme for the data is a graph. Graphs are a natural way to represent data whose pairwise relationships are of interest, and provide additional advantages to better data understanding, flexible schema and identifying the hierarchical flow of information.

However, we can not leverage the benefits of graphs fully due to sensitive data and privacy concerns. For instance, an outstanding issue for sensitive domains, such as health and medicine, is preserving privacy while analyzing this shared data to obtain global insights. Removing personal identifiers and confidential details is insufficient, as we lose accuracy, and an attacker can still impute the missing data using techniques such as in [4]. Additionally, serious threats are encountered in collective AI endeavors that aggregate data from different sources since the weakest link establishes the overall security level for the global analysis.

Therefore the major bottleneck of applying graph-based models to some of the applications or sharing the data and models among parties is data privacy [5]. This has also been identified in two most recent DOE reports [9, 8].

For agencies to analyze the data distributed across different providers, we need sophisticated privacy-preserving algorithms with privacy guarantees. To unleash the power of AI/ML to mine personal and population data, it must retrieve information while preserving an extensive list of privacy concerns.

Mining knowledge graphs (which capture relationships between objects of interest) for particular subgraphs has shown great promise in the field of natural language processing[2]. However, analyzing the knowledge graph with sensitive data is a major concern for unsupervised learning.

When we combine AI/ML, graphs, and privacy, the main challenge is finding the optimal trade-off between privacy loss and data utility in distributed graph learning algorithms and other statistical analyses.

Background

The voluminous data generated from multiple sources (humans and machines) can be processed and analyzed to support graph-based decision-making models. However, these models are prone to privacy violations. There is a vast opportunity for privacy preservation as it enables efficient data sharing, analysis, and decision-making on various data types and systems. In comparison, the usual approach is applying privacy methods on a centralized data repository which requires the data to be moved between systems at a centralized location. We call the models built at this centralized data repository as "global" model and results from the analysis as "global" results.

Introduced by Dwork et al. in 2006 [6], differential privacy (DP) has become a widely studied framework for providing privacy-sensitive results from data analyses. DP algorithms introduce noise or randomness into computations to ensure that it is hard to detect the presence or absence of any individual's data from the output of the algorithms.

There are two parameters (ϵ, δ) for DP algorithms that are used to bound the probability of obtaining the same outcome with or without any particular user, representing the privacy risks of the procedure. In general, DP algorithms have been developed and deployed by Google [7], Apple [3], and the U.S. Census Bureau [1]. For decentralized learning with privacy constraints, DP can be incorporated into federated learning approaches, particularly cross-silo learning: we call this distributed differential privacy (DDP).

In this way, we can leverage the huge amount of data (structured and unstructured), perform graph analysis and get insights into the relationships and patterns while preserving the privacy of individuals or other entities. For example, the unsupervised graph learning approach trains directly from the local node data. The quality of the distributed learning will be closer to global sequential results performed on the aggregated data. The risk assessment for performing analysis on protected data distributed across multiple servers is not well understood, and it is an open problem to ensure this protected data is never communicated to another server, even inadvertently.

Opportunity

In the differential privacy framework, the privacy risk (of re-identification) increases as more information is revealed. Therefore repeated releases increase the privacy risk since they (in combination) allow for more accurate inferences by individuals. Differential privacy can produce dramatic changes in population counts for racial/ethnic minorities in small areas and less urban settings, significantly altering knowledge about health disparities in mortality.

However, we can counterbalance this through the randomness in *uniform* sampling that can sometimes mitigate the privacy loss. An essential area of investigation is a sampling process that can guarantee ethical AI models through a fair label representation of racial groups, rare diseases for differential privacy. The problem exacerbates in distributed differential privacy settings.

Therefore, we need to focus on novel graph samplers with mathematical guarantees to protect associations among entities while providing guarantees for ethical properties with an unbiased, fair representation.

There are several key research challenges for developing DDP algorithms for unsupervised graph learning. First, we need to define a privacy model appropriate for knowledge graphs and understand the translation of the traditional model of neighboring data sets to this setting. For example, there is a body of work on releasing graphs with differing notions of edge and node privacy, but these are unexplored in the differential privacy setting. For the graph learning problems, we should also consider new notions for hypergraphs that are needed to be developed.

Also, we need new DP methods that operate on knowledge graphs. Much of the work in DP has been on supervised learning, and more recently, on deep learning. However, these approaches will not translate directly to unsupervised learning in knowledge graph structures. For example, although simple approaches to spectral clustering using matrix factorization are possible, a central lesson in applied differential privacy is that exploiting prior knowledge of the graph structures or imposing constraints can significantly improve the empirical performance of these methods. This knowledge comes from close collaboration between domain experts and privacy algorithm designers. Finally, to allow for decentralized processing, we must identify what data derivatives need to be shared. Unlike deep learning that often uses simply noisy gradients for privacy, the information fusion for graph-structured representations shall require different types of information to be shared. Especially if the graph architectures differ between collaborating sites, we may need to design methods to align graphs in a privacy-preserving way. This will not be as simple as adding noise to gradients in deep learning.

In addition to developing novel message structures, limiting the number of rounds of communication can help mitigate the overall privacy risk.

Timelines and maturity

The need for privacy preservation of sensitive information will enable more graph-based models to be deployed in the real world without the risk of exposing sensitive information. Also, it will increase the number of gathered diverse data without compromising the accuracy of the models and overcome the potential bias in the models.

This leads to creating graph models trained across shared data from hundreds of providers that will be of better quality over hundred different models trained separately by data providers. The good globally trained model respecting privacy will benefit federal agencies like DOE, NIH, CDC, and the data providers. DDP is an important cornerstone to build confidence in data providers that (a) it will preserve anonymity and (b) any analysis with malicious intent cannot reveal the identity of the data.

The potential outcome will increase the discoveries in medicine and healthcare, improve the confidence in critical national infrastructure, and so forth. It will also provide secure data infrastructure for performing learning so that the policymakers can understand the societal trends and hospitals can improve diagnostics by maintaining patient privacy.

References

[1] John M. Abowd. The u.s. census bureau adopts differential privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, London, UK, August 19–23 2018.

[2] KM Annervaz, Somnath Basu Roy Chowdhury, and Ambedkar Dukkipati. Learning beyond datasets: Knowledge graph augmented neural networks for natural language processing. *arXiv preprint arXiv:1802.05930*, 2018.

[3] Apple Differential Privacy Team. Learning with privacy at scale. *Apple Machine Learning Journal*, 1(8), 2017.

[4] Jiajun Bu, Shulong Tan, Chun Chen, Can Wang, Hao Wu, Lijun Zhang, and Xiaofei He. Music recommendation by unified hypergraph: combining social media information and music content. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 391–400, 2010.

[5] James Curzon, Tracy Ann Kosa, Rajen Akalu, and Khalil El-Khatib. Privacy and artificial intelligence. *IEEE Transactions on Artificial Intelligence*, 2021.

[6] Cynthia Dwork, Frank Mcsherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC '06: In Proceedings of the 3rd Theory of Cryptography Conference*, pages 265–284, 2006.

[7] U. Erlingsson, V. Pihur, and A. Korolova. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, Scottsdale, Arizona, USA, November 03–07 2014.

[8] Sean Peisert. Ascr cybersecurity for scientific computing integrity-research pathways and ideas workshop: Doe workshop report. 2015.

[9] Rick Stevens, Valerie Taylor, Jeff Nichols, Arthur Barney Maccabe, Katherine Yelick, and David Brown. Ai for science. Technical report, Argonne National Lab.(ANL), Argonne, IL (United States), 2020.

Secure Compartmentalization and Partitioning of System Software

Jack Lange[1]

[1]Oak Ridge National Laboratory

1 Introduction

The next generation of supercomputing resources is expected to greatly expand the scope of HPC environments, both in terms of more diverse workloads and user bases, as well as the integration of edge computing infrastructures and sensitive/protected data repositories. The effect of this expansion will be that the users of a given system will likely come from multiple organizational units with their own local security policies and identity management. In addition, the data accessed on the systems will also likely have independent access control policies that must be enforced across organizational boundaries. It is possible that in the future a HPC system will be running a software environment that the local system administrators will not be permitted to access or inspect. Supporting these use cases will require new mechanisms and approaches at the Operating System level to support these broader classes of workloads along with their different security requirements. We claim that a key mechanism needed for these workloads is the ability to securely compartmentalize entire system software stacks executing on a given node.

To develop these capabilities we envision leveraging current and future hardware capabilities that will likely be available on next generation system architectures. These include hardware support for Trusted Execution Environments (TEEs), availability of low level resource virtualization support via firmware/software hypervisors, and distributed/disaggregated hardware architectures including co-processors and DPUs. These system features will provide system software with a rich ecosystem of functionality that can be leveraged to provide security isolation at multiple levels of the system software stack. A central challenge to enabling robust security capabilities on future platforms will be developing the system software frameworks to effectively and efficiently allow environmental isolation required by inter- and intra-organizational security policies. We expect that fully supporting the breadth of security policies will require a multi-layer approach that incorporates a range of isolation capabilities from user level library isolation (ala Intel SGX) to full OS/container level isolation via secure partition managers/hypervisors.

Trusted Execution Environments (TEEs) have recently received a significant amount of attention due to their capabilities of establishing secure and trustworthy computational environments on otherwise untrustworthy hardware and software environments. While originally designed for end-user devices and commodity system platforms, there has been gnificant interest in adapting these technologies to server class systems. However, so far relatively little work has gone into exploring how to deploy TEEs in an HPC platform setting, mainly due to the scale and performance constraints that these environments introduce. However, as these features become more capable, their utility for HPC platforms will increase and present promising approaches for integrating rich and distributed security policies in future environments. The research challenge with deploying these features will be how best to integrate them into HPC system software environments and application workflows, especially as the hardware platforms become increasingly heterogeneous with complex hardware topologies and diverse architectures.

In addition to the application library and RPC service based TEE approaches, we claim that future systems should also support the compartmentalization of full system software stacks in order to allow full Operating System/Runtime environments to execute in isolated security partitions. This capability is becoming increasingly available via hardware extensions from AMD (SEV [1]) and ARM (TrustZone [6]). While these hardware based approaches has so far not been fully exploited by existing software stacks, they present the opportunity to deploy extensive security partitioning capabilities at the OS layer. Each provides the means of deploying a trusted hypervisor that can create hardware level memory partitions enforced using a combination of encryption and/or address filtering by the memory controller. Current approaches have been limited by existing software architectures and architectural assumptions, for instance SEV has limited applicability to host based virtualization approaches (such

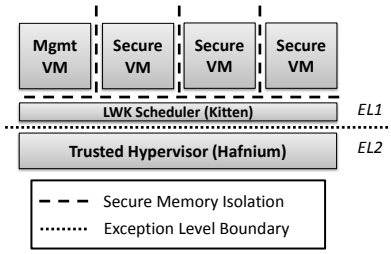


Figure 1: Virtualization based Compartmentalization using the Hafnium Hypervisor

as KVM) due to the requirement that the VMM be fully trusted, similarly TrustZone based approaches have generally assumed that the software executing in the secure world would be significantly limited in its scope and utility.

Finally, other approaches for supporting secure system partitions exist as well, such as those implemented at the firmware/hardware layer by system vendors as well as isolating virtualization solutions that execute as type 1 hypervisors beneath every OS on the system. While these approaches do exist, they are either not directly accessible by system users or have not been fully adapted for use in HPC contexts. These shortcomings can be remedied through either the exposure of firmware APIs by system vendors, the adaptation or existing virtualization solutions, or clean sheet designs of HPC focused hypervisors.

Therefore, while it is possible with modern system architectures to provide limited partitioning and security isolation capabilities, we have not yet reached the point where general purpose security isolation is a fully supported and available feature on modern systems. While secure function evaluation and limited data protection are possible at present, we instead envision a future where full stack container environments (complete OS and user space environments) are able to be dynamically instantiated in trustable and secure execution environments and provided with hardware level resource isolation (CPUs, memory, and I/O devices). With this vision in place, it will be possible to ensure that data and computation can move between HPC platforms as needed while ensuring that their own internal security policies are honored and enforced.

2 Proof of Concept

To demonstrate our vision, we have developed a virtualization based approach as a proof of concept, which allows secure isolation of full OS/R environments on ARM64 systems both with and without TrustZone [4]. Our prototype system software stack is based on the Hobbes Exascale OS/R environment [3] and the Hafnium secure partition manager (SPM) [2]. Our approach combines a secure partition manager framework such as Hafnium with an HPC oriented lightweight kernel scheduling OS to execute full HPC applications inside securely isolated virtual machine environments. A high level overview of our approach is shown in Figure 1. While our current implementation is severely limited due to constraints at both the hardware and software levels (i.e. static partitions defined at boot time, limited I/O capabilities, and limited guest OS features), we believe that these are surmountable challenges and will likely resolve themselves with future hardware iterations.

3 Challenges

While existing solutions provide a path forward, there are still a number of challenges that must be addressed at both the hardware and software layers. The most pressing issue for HPC environments is the need to both increase the partition capacity (notably in terms of memory size) and support dynamic resizing of partitions at runtime. While this is primarily a challenge for the underlying hardware implementations, there will also need to be support integrated into the system software frameworks to effectively manage those partitions. In addition the ability to partition hardware needs to be extended to additional compute and device resources beyond CPU and memory, preferably at sub-device granularities. Again this is mostly a hardware challenge, but will also require system software support. On the software side, management interfaces and secure cross partition communication channels are likely to be necessary to support system administration and application composition requirements. Finally, these features will need to be introduced without negatively impacting the security guarantees of the partitions themselves, either via direct or side channel attack vectors.

References

[1] Secure Encrypted Virtualization Key Management. `https://developer.amd.com/sev/`.

[2] Hafnium Hypervisor. `https://www.trustedfirmware.org/projects/hafnium/`.

[3] B. Kocoloski et al. System-Level Support for Composition of Applications. In *International Workshop on Runtime and Operating Systems for Supercomputers*, 2015.

[4] J. Lange, N. Gordon, and B. Gaines. Low Overhead Security Isolation using Lightweight Kernels and TEEs. In *International Workshop on Runtime and Operating Systems for Supercomputers*, 2021.

[5] AMD TrustZone. `https://developer.arm.com/technologies/trustzone`.

Position Paper
Oct. 12. 2021
David Lawrence Ph.D.
davidl@jlab.org
Thomas Jefferson National Accelerator Facility

Federated Analysis of Experimental Data

Analysis of Experimental Nuclear Physics (ENP) data generated at accelerator facilities requires significant compute resources. Increasing network bandwidth and connectivity make it possible to process this data at sites remote from where the data was generated. Experiments such as STAR at BNL and GlueX at JLab have used the NERSC facility to process significant portions of experimental data sets[1][2]. The Electron Ion Collider(EIC) is anticipated to make even greater use of remote compute facilities to process the data in near-real time. There is benefit in diversifying the facilities used to process data such as:

- Less dependence on single sites when servicing real-time or near-real-time workflows
- Smaller fraction of each single site needed for a specific workflow, allowing them to maintain greater diversification in the jobs they service
- Better access to heterogeneous resources that may not exist at all sites (e.g. one site may be GPU heavy while another is FPGA heavy)

One of the issues encountered with using multiple HTC sites for processing GlueX data is the need for multiple forms of, periodic re-authentication. GlueX processes experimental data primarily at three remote sites (in addition to the farm at JLab where the data originates). These are NERSC, the Pittsburgh Supercomputing Center (PSC) and Indiana University Big Red3. The workflow requires that an automated system be used to transfer the experimental data to the remote site, submit a job to the remote sites' system for each file as it arrives, and then transfer the resulting output files back to the home institution. For GlueX, processing at NERSC, this requires the following authentications and frequencies:

Authentication Type	Method	Duration
Globus (OAuth token for use with CLI)	Browser-based login*	?
JLab DTN from Globus (jlab#gw12 outgoing)	Browser-based login	30 days
JLab DTN from Globus (jlab#scidtn incoming)	Browser-based login	30 days
NERSC DTN from Globus	Browser-based login (MFA)	11 days
PSC bridges-2 DTN	Browser-based login	10 days
IURT - Slate	Browser-based login	8 days
Remote shell access to NERSC	ssh key	30 days
Remote shell access to PSC	ssh agent	varies
Remote shell access to IU	ssh agent	varies

To add to the complication, some of the connections use a personal account (e.g. Globus) while others (e.g. NERSC, JLab) use group accounts. A group account is appropriate since long workflows require multiple admins. Ideally, there would be a system that would allow a single authorized member of a

group to reauthenticate the group to all edge facilities with a single action. For example, responding to an alert that the group's authentication will expire in 24 hours by using MFA or biometrics to renew. It is critical that the authentication is multi-directional since raw data files will typically be read from the experimental facility and the outputs written to a long-term storage facility that could be a 3rd site.

In addition to federating the authentication, a system to advertise system capabilities as well as current and anticipated availability would be very useful for the heterogeneous era. Similar to the OSG, instead of targeting jobs to a specific site, the jobs are submitted locally with requirements and preferences. The remote systems would then match the jobs and pull them rather than having the local site push them. The end effect would be jobs seeing the multiple federated computing sites it is authenticated to use as a single, large resource pool. Special hardware (e.g. FPGAs) would be in either the requirements list or preferences list if the software has optional support of it built-in.

References:

[1] "*STAR Data Production Workflow on HPC: Lessons Learned & Best Practice*",M. Poat, et al. , **ACAT2019**, https://indico.cern.ch/event/708041/

[2] *"Offsite Data Processing for the GlueX Experiment",* D. Lawrence, **EPJ Web of Conferences 245**, 07037 (2020) https://doi.org/10.1051/epjconf/202024507037

Federated Tensor Models for Data Integrity

Cannada Lewis (canlewi@sandia.gov)*, Eric T. Phipps, and Hemanth Kolla

October 14, 2021

1 Topics (Data Integrity, Federated Learning, Distributed Error Detection)

This paper seeks to address issues with **error detection** and **data integrity** for **distributed data** that is either hard or impossible to collect in a single processing location. One method we believe will be useful for this challenge is the use of **federated learning**, specifically for the tensor decompositions including the optimization of the generalized canonical polyadic (GCP)[5] tensor decomposition.

2 Challenge (Distributed Data and Input Trust)

Smart power grids and accurate weather prediction are critical for the efficient function of transportation, energy availability, and advanced notice of severe events. Both rely on a geographically distributed network of sensors and data where processing power may be limited either due to location, time constraints, or sheer volume.

Weather prediction models depend on accurate and timely inputs [10, 6, 2], that assimilate a huge amount of distributed raw information into a validated well defined output. Part of this process is a complicated step that involves predicting the accuracy and possible error in raw inputs based on a combination of historical data, background measurements and previous models [3].

Similarly, the power grid must be constantly adjusted to provision resources, increase resiliency, and reduce climate impact [4]. As critical infrastructure, the power grid is closely monitored for issues, threats, and attacks. Different levels of the power grid communicate via a variety of different standards, from wireless communication protocols at customer sites to dedicated infrastructure, but what they all have in common is a need for security and data integrity. However, monitoring must balance the need for information with the privacy and security of the customer's data.

Both the security of the energy grid and timely and accurate weather prediction present computational challenges due to

- Distributed and remote inputs
- Vulnerability to erroneous or manipulated sensors
- Large data and/or low bandwidth.

While these industries have strategies that somewhat mitigate these challenges, future demands for both are expected to grow and new techniques for data assimilation and analysis will be required.

*Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. Sand Number: SAND2021-12935 O

3 Opportunity And Timeliness (Tensor Decompositions Can Meet These Needs)

Fields such as weather prediction and smart power grids present a difficult challenge, but provide an opportunity to exploit federated models[8] for data prediction and integrity. Federated learning is a technique that trains models while avoiding running the training on centralized hardware. Federated learning allows for training even in situations where data must not or can not be moved either due to low bandwidth or security concerns. Typically models are trained by sending the model or part of the model to the data and doing the training locally. Recently, there has been an increase in interest in the ability to build models from large, heterogeneous, distributed data, with the goal of limiting data movement and preserving privacy. Two of the authors (E. Phipps and C. Lewis), have been working on federated learning for GCP [7]. We have also conducted work to show that tensor based methods can be used to generate compressed models for distributed data, which can be used for anomaly/error detection [1].

As our ability to generate raw data outgrows our ability to centralize and store it, we will need the ability to build models that are constructed at the point of collection, and that alert us to erroneous or invalid data, without relying on a centralized process. We believe that federated generalized tensor decompositions trained using methods such as FedAvg [9] or FedOpt [11], could meet both of these goals, even in the presence of low-bandwidth networks or large data. Thus future funding calls should take into account the potential for tensor decompositions trained with federated methods as part of a holistic strategy for cyber security and scientific data integrity.

References

[1] K. Aditya, H. Kolla, W. P. Kegelmeyer, T. M. Shead, J. Ling, and W. L. Davis IV, *Anomaly detection in scientific data using joint statistical moments*, Journal of Computational Physics, 387 (2019), pp. 522–538.

[2] S. G. Benjamin, S. S. Weygandt, J. M. Brown, M. Hu, C. R. Alexander, T. G. Smirnova, J. B. Olson, E. P. James, D. C. Dowell, G. A. Grell, et al., *A north american hourly assimilation and model forecast cycle: The rapid refresh*, Monthly Weather Review, 144 (2016), pp. 1669–1694.

[3] A. M. Clayton, A. C. Lorenc, and D. M. Barker, *Operational implementation of a hybrid ensemble/4d-var global data assimilation system at the met office*, Quarterly Journal of the Royal Meteorological Society, 139 (2013), pp. 1445–1461.

[4] Z. El Mrabet, N. Kaabouch, H. El Ghazi, and H. El Ghazi, *Cyber-security in smart grid: Survey and challenges*, Computers & Electrical Engineering, 67 (2018), pp. 469–482.

[5] D. Hong, T. G. Kolda, and J. A. Duersch, *Generalized canonical polyadic tensor decomposition*, SIAM Review, 62 (2020), pp. 133–163.

[6] D. T. Kleist, D. F. Parrish, J. C. Derber, R. Treadon, W.-S. Wu, and S. Lord, *Introduction of the gsi into the ncep global data assimilation system*, Weather and Forecasting, 24 (2009), pp. 1691–1705.

[7] C. A. Lewis and E. T. Phipps, *Low-communication asynchronous distributed generalized canonical polyadic tensor decomposition*, in 2021 IEEE High Performance Extreme Computing Conference (HPEC), IEEE, 2021.

[8] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, *Federated learning: Challenges, methods, and future directions*, IEEE Signal Processing Magazine, 37 (2020), pp. 50–60.

[9] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, *Communication-efficient learning of deep networks from decentralized data*, in Artificial intelligence and statistics, PMLR, 2017, pp. 1273–1282.

[10] F. Rawlins, S. Ballard, K. Bovis, A. Clayton, D. Li, G. Inverarity, A. Lorenc, and T. Payne, *The met office global four-dimensional variational data assimilation scheme*, Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography, 133 (2007), pp. 347–362.

[11] S. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečnỳ, S. Kumar, and H. B. McMahan, *Adaptive federated optimization*, arXiv preprint arXiv:2003.00295, (2020).

Towards Privacy-Preserving and Robust Federated Learning

Hai "Helen" Li, Professor
Electrical and Computer Engineering Department, Duke University
Email: hai.li@duke.edu

Federated learning (FL) is an emerging distributed machine learning paradigm that enables massive clients to train a shared model in a federated fashion without transferring their local data. In general, a central server coordinates the FL process, where each participating client communicates only the model parameters with the central server while keeping local data private. FL has been widely applied to many critical applications, such as electric load forecast [1], EV-based energy demand prediction [2], digital health [3], etc. Privacy preservation is a critical property offered by FL. Unfortunately, recent studies [4] have revealed that Fl could be vulnerable to various inference attacks, e.g., property inference attacks and model inversion attacks, due to the sharing of model updates or gradients. Such privacy leakage from FL may result in catastrophic consequences. For example, the leakage of power traces will lead to grid vulnerabilities being exposed and targeted, affecting the local economy and even national security. In addition, recent works have shown that FL could also be vulnerable to other forms of attacks, e.g., model poisoning attacks and adversarial transfer attacks. Such vulnerabilities of FL will either result in degraded machine learning models or leave an attacking interface to adversaries. Thus, there is an urgent need to improve the privacy guarantee and robustness to enable the applications of FL to more safety-critical applications in practice. Here, we propose to develop privacy-preserving and robust FL solutions from the following perspectives.

Privacy-Preserving FL. FL is vulnerable to inference attacks, including property inference attacks and model inversion attacks [5]. Here the property inference attack infers sensitive properties of training data using the model updates, while the model inversion attack attempts to reconstruct training data using model gradients. The causes of such privacy leakages have not been thoroughly investigated or fully understood. Defenses techniques, such as differential privacy (DP), secure multi-party computation (MPC), and data compression, have been developed to prevent privacy leakage. Since these defensive approaches are not specifically designed for the privacy leakage from the communicated local updates, they usually incur significant computational overhead and/or unignorable accuracy loss. Our latest study reveals that privacy leakage is essentially caused by the data representations embedded in the model updates. Based on this observation, we propose to design a defense scheme against model inversion attacks from the representation perspective. The key idea of our defense is to identify the part of representation that is most vulnerable to model inversion attack and perturb this part of representation during local training.

Defending against adversarial transfer attacks in FL. Deep learning models are vulnerable to adversarial perturbations – small additive noises that could fool the model to yield malicious behavior. The FL scenario could make the adversarial attacks even more accessible. For example, a malicious user could craft adversarial examples against the global model that is shared across all clients. The situation poses significant security threats to benign users and sparks the urgency of developing a defense against such attacks between clients. One defense strategy is to reduce the attacker's knowledge of other clients' models by training personalized local models instead of a single global model. In our recent work LotteryFL [7], we let each client learn a unique binary mask, which will be multiplied by the global model to produce local models. Future studies include investigating the robustness of LotteryFL against cross-client attacks and designing a more effective and organized mask training algorithm. Moreover, the transferability of adversarial attacks is a concern, while personalized models might not be sufficient to defend it. Previously, we developed DVERGE [8], a training algorithm for preventing adversarial transfer between models. Here, we plan to explore robust training of personalized models by combining DVERGE with LotteryFL.

Defending against model poisoning attack in FL. Model poisoning attacks can be conducted in untargeted or targeted ways. Untargeted model poisoning attacks attempt to increase the error rate of the global model indiscriminately. The objective of targeted model poisoning attacks [6] is to make the global model generate attacker-desired misclassifications for some particular test samples. Several studies have been done to improve the robustness of FL against model poisoning attacks through robust aggregations, clipping local updates, and leveraging the noisy perturbation. These defensive methods fail to guarantee robustness when attacks are strong. More importantly, we observe that as long as the global model is polluted, the impact of attacks on the global model will remain in subsequent rounds even if there are no subsequent attacks. As such, an additional defense is needed to mitigate the poisoning attacks that cannot be eliminated by robust aggregation and will pollute the global model. We propose to design a defense against model poisoning attack that can identify the parameter space where long-lasting attack effect on parameters resides during local training and squeeze the dimension of that space. Our client-based defense should be complementary to the current server-based defense and can enhance the robustness of FL against the model poisoning attack, especially against the extremely strong attacks that can not be mitigated during the aggregation.

Improving robustness against out-of-distribution data in FL. Besides the aforementioned attacks that are artificially crafted, detecting the natural Out-of-Distribution (OOD) data (i.e., abnormal or novel inputs that do not come from the training distribution) is crucial for a robust system to operate in the open world. FL provides both opportunities and challenges for enhancing the robustness against OOD samples. While the non-IID characteristic of FL suggests the potential of leveraging other clients' data as outliers for training better OOD detectors, the key problem is how to exploit and exchange useful information between clients without compromising their data privacy. Developing an effective and privacy-preserving method to enhance OOD detection is another meaningful step towards robust FL.

Reference

[1] Taïk, Afaf, and Soumaya Cherkaoui. "Electrical load forecasting using edge computing and federated learning." *2020 IEEE International Conference on Communications (ICC),* 2020.

[2] Saputra, Yuris Mulya, et al. "Energy demand prediction with federated learning for electric vehicle networks." *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019.

[3] Rieke, Nicola, et al. "The future of digital health with federated learning." *NPJ digital medicine* 3.1 (2020): 1-7.

[4] Lyu, Lingjuan, Han Yu, and Qiang Yang. "Threats to federated learning: A survey." *arXiv preprint arXiv:2003.02133* (2020).

[5] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. *The 22nd ACMSIGSAC Conference on Computer and Communications Security*, 2015.

[6] Sun, Z., Kairouz, P., Suresh, A. T., & McMahan, H. B. (2019). Can you really backdoor federated learning?. *arXiv preprint arXiv:1911.07963*.

[7] Li, Ang, et al. "Lotteryfl: Personalized and communication-efficient federated learning with lottery ticket hypothesis on non-iid datasets." *arXiv preprint arXiv:2008.03371* (2020).

[8] Yang, Huanrui, et al. "DVERGE: Diversifying Vulnerabilities for Enhanced Robust Generation of Ensembles." *Advances in Neural Information Processing Systems* 33 (2020).

Toward Effective Security/Reliability Situational Awareness via Concurrent Security-or-Fault Analytics (SoFA)

Mingyan Li (lim3@ornl.gov), Robert A. Bridges (bridgesra@ornl.gov), Pablo Moriano (moriano@ornl.gov), Christian Engelmann (englmannc@ornl.gov), Feiyi Wang (fwang2Wornl.gov), Ryan Adamson (adamsonrm@ornl.gov)

Oak Ridge National Laboratory, Oak Ridge, TN, USA

Topic: Adversarial modeling & simulation. Integrity & provenance. Secure data architecture. Graph algorithms.

Challenge: Modern critical infrastructures (CI) and scientific computing ecosystems (SCE) are complex and vulnerable [1]. The complexity of CI/SCE, such as the distributed workload found across ASCR scientific computing facilities, does not allow for easy differentiation between emerging cyber security and reliability threats. It is also not easy to correctly identify the misbehaving systems. Sometimes, system failures are just caused by unintentional user misbehavior or actual hardware/software reliability issues, but it may take some significant amount of time and effort to develop that understanding through root-cause analysis. On the security front, CI/SCE are vital assets. They are prime targets of, and are vulnerable to, malicious cyber-attacks. Within DoE, inter-disciplinary and cross-facility collaboration (e.g., ORNL INTERSECT initiative, next-gen supercomputing OLCF6), traditional perimeter-based defense and demarcation line between malicious cyber-attacks and non-malicious system faults are blurring. Amidst realistic reliability and security threats, the ability to effectively distinguish between non-malicious faults and malicious attacks is critical not only in root cause identification but also in countermeasures generation.

Today's segregated fault diagnosis and attack detection approaches leave much room for improvement. Anomalies during attack detection could very well point to system faults (thus the problematic high false-positives with root-cause identification failure). Similarly, fault diagnosis anomalies could be from attacks (dangerous false negatives). Realistic situational awareness requires a wholistic analytics approach, counting in both security and faults contexts, and concurrently, to achieve effective situational awareness and realistic root-cause identification.

Opportunity: Concurrent & parallel security/fault analytics improves attack/failure root-cause analysis quality. There are two implications – proactive/adaptive analytics and countermeasure generation. There are no guarantees that when anomalies were first detected, sufficient evidence has already been collected and analyzed, particularly in the real-time scenarios where attacks/faults are progressing. Analyzing, and adaptively collecting additional relevant data pertinent to scenario developments, on both malicious and non-malicious fronts, helps to properly exonerates/confirms suspected attack/fault developments. Examples of additional data include increased-fidelity telemetry streams, directed endpoint detection and response (EDR) queries, or even information normally kept by third party organizations relevant to a suspicious CI/SCE workflow. With accurate root-cause identification, effective countermeasures (repair or defend) can then be devised and applied. Such an improved fault/attack paradigm could serve to improve and protect DoE CI/SCE where detection and defense of a compromised workflow component at one facility could prevent the spread of a malicious payload to another. One potential approach is to model system's behavior with (temporal) graphs and using the interactions between its different states to uncover complex modes of operation, including faults and attacks [2]. We can further employ model-driven checkpoints based on scientific and data workflow to proactively monitor, collect and detect anomalies, in both data privacy and federated learning contexts.

Figure-1 conceptually illustrates concurrent attack/fault analytics. On the left is an example fault tree [3], or it

could be a state graph with nodes represent behavioral states and edges represent transitions between states. Observed events are mapped into perspective nodes (colored dots). Scenarios are analyzed horizontally across, denoted by the directed curve. On the right, a conceptual adversarial model (an attack strategy goal/subgoal model here – could also be ML-based models) [4], also with horizontal scenario analytics curves. Here, concurrent analytics takes place. If called for, in real-time environment, the detection system can adaptively adjust auditing behavior to proactively look-ahead to monitor/search for relevant data per suspicious scenario developments (e.g., dual arrows on the right). Otherwise, in offline mode, retroactively retrieve required data locally or externally for assessment. Such concurrent attack/fault analytics improves root-cause attribution effectiveness thus leads to improved situational awareness and countermeasure generation [5].

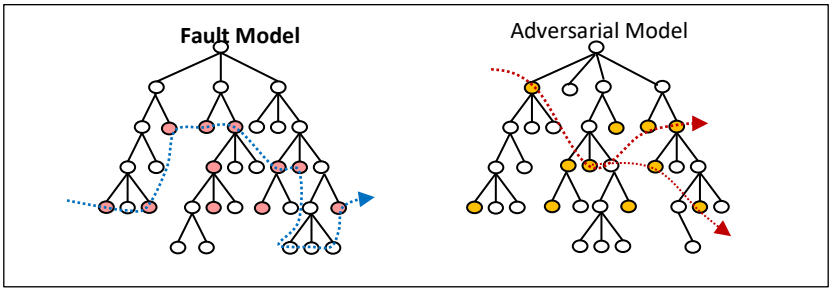


Figure-1 Concurrent fault & attack analytics conceptual illustration

Proactive adaptive auditing, i.e., dynamic data ingestion, further helps to alleviate the "information impoverished" problem in big data collection. Adaptively audit/collect only contextual relevant data, per scenario development, reduces the "needle in the haystack" scalability challenge in real-life applications.

Timeliness/Maturity: Both failure/fault diagnostic and security analytics (e.g., Host-based/network-based Intrusion Detection Systems, and ML-based anomaly detection systems) technologies have been investigated in the past, but two stove pipes did not really come together to realistically address the real-life situational awareness challenge. Past attempts have been made to build/utilize declarative security attack as well as fault models, but they were severely limited by scalability in terms of declarative knowledge capturability. Recently, work on streaming telemetry and log events has matured within DoE supercomputing facilities, and platforms exist to enable dynamic software-defined data collection. With today's machine learning based or graph-based data science approaches, there lies the potential of realization of such integrated multi-modal approach.

References:

[1] National Strategic Computing Reserve, https://www.whitehouse.gov/wp-content/uploads/2021/10/National-Strategic-Computing-Reserve-Blueprint-Oct2021.pdf

[2] B. Bowman, C. Laprade, Y.Ji, and H. Huang. "Detecting Lateral Movement in Enterprise Computer Networks with Unsupervised Graph {AI}." In 23rd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2020), pp. 257-268. 2020.

[3] Z. Gao, C. Cecati, and S. X. Ding, "A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part I: Fault diagnosis with model-based and signal-based approaches && Part II: Fault Diagnosis with Knowledge-Based and Hybrid/Active Approaches", IEEE Trans. Ind. Electron., vol. 62, no. 6, Jun 2015

[4] Y. Deldjoo, T. D, NOIA, and F. A. Merra. "A survey on Adversarial Recommender Systems: from Attack/Defense strategies to Generative Adversarial Networks", ACM Comput. Surv., Vol. 54, No. 2, Mar 2021

[5] G. Tertytchny, N. Nicolaou, and M. Michael "Classifying network abnormalities into faults and attacks in IoT-based cyber physical systems using machine learning", Microprocessors and Microsystems, Sept 2020.

Research Opportunities for Ensuring HPC Data Integrity and Provenance

Chunhua Liao[1], Matthew Sottile, Tristan Vanderbruggen, Steve Chapin.
Lawrence Livermore National Laboratory

Introduction:

Verification of trustworthiness in computing often focuses on computations instead of the data itself. Even when data is considered, the focus is generally on the computations related to handling the data. Trustworthy science requires data verification to be treated as equally important as computations. Data integrity is the maintenance of, and the assurance of, data accuracy and consistency over its entire life cycle. Aiming to prevent unintentional changes to information, it is foundational to trust in simulation models and their predictions. Advances in data integrity verification are necessary to ensure that data is not tampered with, provenance is tracked through the full lifecycle of a scientific modeling activity, and the semantics of the data conform to the requirements of modeling and analysis methods.

Use cases:

We identify two typical scenarios in which data integrity is highly relevant to High-Performance Computing (HPC). First, traditional scientific computing using supercomputers is a data-heavy process that depends on data integrity to allow researchers to produce trustworthy results. Secondly, as machine learning techniques are being widely deployed in HPC, large numbers of datasets and AI models are becoming available for sharing and reusing in the community. Robust techniques for ensuring data integrity and provenance are necessary to give stakeholders confidence that they can trust data made available.

We believe that data integrity can be approached in at least three aspects. The first concern is **well-formedness** – does the data conform to the I/O requirements of tools. An example requirement is that an input data file should conform to a predefined data schema stored in a preferred standard file format (e.g., XML). Second, is the data **meaningful** at the application domain level – do physical, numerical, and geometric properties assumed by the application hold? Third, what is the **provenance** of the data – can a simulation result or AI models be traced in a trustworthy and secure fashion back to the exact authors and software/hardware configuration used to generate it? All of these aspects mentioned above have obvious cybersecurity implications.

Challenges:

---

[1] Corresponding Author: liao6@llnl.gov This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL-ABS-827739

While existing schema-oriented input validation techniques are well established to achieve the first goal of ensuring well-formedness of data, schemas often are not rich enough to capture domain level semantics. For example, an application may impose geometric constraints on a mesh that require a validator to not only examine the representation of the mesh as a collection of integers and floating-point numbers, but also examine properties of the higher-level geometric structure that the collection denotes. Such constraints are outside the reach of validators today due to a lacking specification mechanism for expressing domain level data constraints and a lack of tooling to check such specifications. As to data provenance, current scientific workflows for generating data do not automatically provide secure and verifiable provenance information.

Research Opportunities:

A promising research direction is compile-time, runtime, and offline steps to automatically check data integrity-related assertions throughout the life cycle of data generation, storage and retrieval. New forms of assertions will be needed to express rich semantic constraints of high-level domain concepts. They may also involve properties of an entire dataset (beyond just at the record or individual array level). These assertions can form the basis for offline verification of formal theories of data consistency and integrity using external theorem provers and satisfiability solvers. A number of formal methods tools that allow users to work at the level of mathematical theories provide the building blocks for writing mathematical assertions and theories, but to our knowledge no tool exists to connect them to data schemas.

HPC programming models have traditionally been focusing on performance, correctness, and energy efficiency. There is an opportunity to incorporate data integrity as a first-class citizen into the design goals of programming models. Directive-based programming models (such as OpenMP) provide a flexible framework for experimental language, compiler and runtime extensions concerning data integrity and provenance. We also need to extend programming models to manage high-level semantics associated with complex (and potentially distributed) data structures. Doing so will allow data semantics to be analyzed and verified in concert with computational semantics to support data integrity-related logic to co-exist in the presence of parallelism and compiler optimizations.

Blockchain-like techniques should be exploited for improving data provenance. To detect data tampering, all data must be signed and verified with low overhead. We need to study scalable and multi-institutional mechanisms to allow researchers from all over the world to manage their digital keys. As data is automatically generated by software, we also need to explore the possibility of treating software as agents and grant independent signing power to running software instances. So, software-generated data can automatically have built-in data integrity and provenance information to enable data forensics.

Toward Automated Defense Against The Adversarial Attacks and Forgeries in National Security Video Surveillance

Yuewei Lin (ywlin@bnl.gov), Shinjae Yoo (sjyoo@bnl.gov)
Computational Science Initiative
Brookhaven National Laboratory, Upton, NY, USA

Risk and Challenge

In the national security arena, video/images are used for a wide variety of purposes, including physical security surveillance for sensitive facilities, remote monitoring and verification of the Non-Proliferation Treaties, and video surveillance in nuclear safeguards. In the context of treaty monitoring and verification, the main primary case of video surveillance as a treaty verification tool is to serve as alternative or supplement to direct access by inspectors, especially in areas that are sensitive to national security such that a treaty party wants to restrict access by inspectors. However, the monitored party is able to expand direct access to video surveillance equipment or to review videos/images post facto, which increases the probability of success of any attempt to conduct an adversarial attack or forgery to the surveillance video. If video surveillance technology is to be included in the verification measures for future arms control treaty negotiations, it must have confidence in the reliability of the method of detecting adversarial attack and tampering to ensure that the determination of authenticity is accepted by all parties.

Figure 1 shows several attack/forge techniques in the natural image/video area. Adversarial attack aims to fool the machine/deep learning models by adding the intentionally designed but visually imperceptible perturbations to the video/images, while the forgery is the generated or manipulated fake video/images. All the attack/forge techniques could be straightforwardly used on security surveillance videos. Adversarial attack and manipulation forgery have potentially serious effects, because such attacks/forgeries are difficult to detect by the human eye.

Defending against such attacks and forgery is challenging. Due to security reasons, custom-designed hardware solutions may not be acceptable by both parties to the treaty, while commercial off-the-shelf (COTS) hardware will quickly change and become obsolete. Security concerns may also restrict data transfer outside the monitored site. Machine/deep learning models are expected to be used to defend against (machine learning-based) attack/forgery. For image level, existing works usually utilize adversarial training for adversarial attacks, and frequency domain information for either local or global manipulation detection. At the video level, time series analysis techniques, such as long-shot item memory (LSTM) or Transformer, are used to detect time inconsistencies. However, the generalization ability of the existing methods is poor. They solve one type of attack/forgery by using ad hoc techniques. Therefore, without knowing the actual attack/forgery type, it is necessary to exhaust all types of attack/forgery techniques to ensure that the system runs well. Designing a deep neural network (DNN) specifically for this task is another challenge. Although CNN-based models have been successfully applied to many tasks, directly borrowing neural architecture from other tasks may lead to inferior performance as it fails to take into account the specific characteristics of this specific task. On the other hand, manually designing a neural architecture is a trial-and-error process that is intractable and time-consuming.

Opportunity

In order to defend against different possible adversarial attacks and forgeries in the national security arena, we must develop a framework with the following three characteristics: 1) *Generalizability:* This framework should be able to defend against different types of attacks at the same time, it should be general enough to enhance the model robustness against adversarial attacks, detecting multiple types of tampering, including deep learning-generated forgeries, classic image processing, before the lens tampering, and video splicing. This framework should also be hardware agnostic, *i.e.,* it can be flexibly adapted to many different camera models. 2) *Online/offline module switch:* The framework is expected to be able to work continuously during image capture and/or review previously recorded shots by the inspector. This provides potential end-users (*e.g.,* NA-243) with great flexibility as they

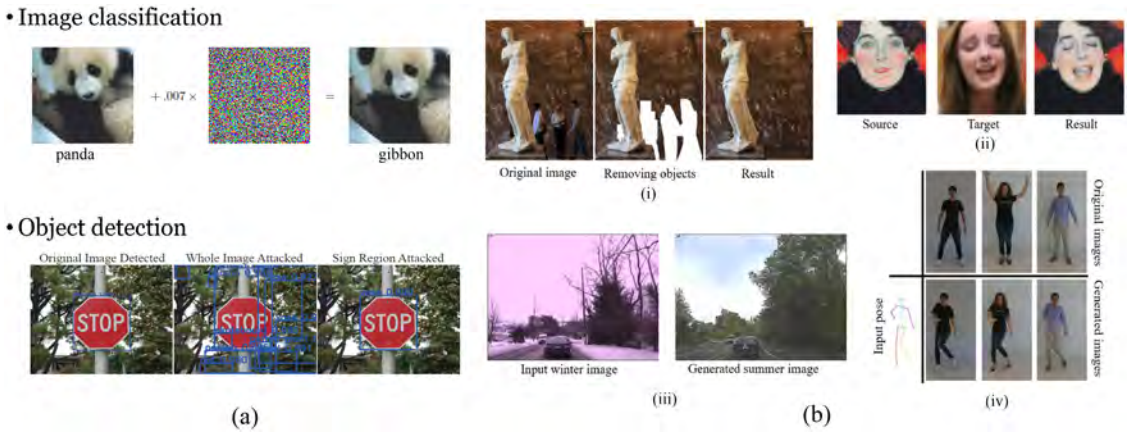


Figure 1. Illustrations of (a) adversarial attacks on image classification: a "panda" image will be misclassified as "gibbon" with high confidence by only adding visually imperceptible perturbations (note the factor .007), and object detection: the attack can let the model either couldn't the find the stop sign, or misclassify the detected stop sign to be a "vase"; and (b) forgeries: (i) image inpainting: removing objects (the people) from an image and fill the regions with the background, (ii) preserving the source image's identity but manipulates its intrinsic attributes (*e.g.*, pose and expression) by using target image's attributes, (iii) image-to-image translation for scene transfer, (iv) pose transfer: generating a sequence of the person in the original image performing specific motions given by a pose sequence.

seek to put a complete video surveillance system for treaty verification into field use. 3) ***Interpretability:*** This framework can identify potential manipulated (spatial) regions and/or (temporal) segments for further review and inspection. Considering that the general defense model is very different from other tasks, and facilities usually have limited computing resources, we also need a model to jointly optimize the neural network architecture and model size for the defense model.

We are making effort in this area. We have developed a model [1] that utilizes attention-guided knowledge distillation and bidirectional metric learning to defend against adversarial attacks and improve the adversarial robustness of the model. The extensive experiments show that our proposed model has better adversarial robustness than the state-of-the-art approaches. We have developed a differentiable neural architecture search (NAS) model, which can automatically optimize the neural structure of face forgery detection tasks [2]. It can help us achieve a good balance between prediction accuracy and model size.

Impact

Successful outcomes in this area will provide a universal and robust framework for defense against adversarial attacks and forgeries. It solves the shortcomings of the existing models, and the generalization of the algorithm will enable users to defend against various adversarial attacks and forgery methods. These technologies will also be camera hardware agnostic and adaptable to the future evolution of potential adversarial attacks and counterfeiting. AI/ML analysis can be used on its own to analyze previously captured footage, or it can be incorporated into a complete video surveillance system that complements other forms of defense against the adversarial attacks and forgeries as part of a defense in depth approach. The technology developed in this area will help NNSA prepare for new treaty verification and monitoring. In addition to treaty verification, it has potentially a wide range of uses throughout the national security field, including video surveillance implemented by the IAEA as part of safeguards containment and surveillance, as well as numerous nuclear and homeland security applications.

References

[1] H. Wang, Y. Deng, S. Yoo, H. Ling, and Y. Lin*, "AGKD-BML: Defense against adversarial attack by attention guided knowledge distillation and bi-directional metric learning," *IEEE International Conference on Computer Vision*, 2021.

[2] P. Liu, Y. Lin, Y. He, Y. Wei, L. Zhen, T. Zhou, and R. Goh, "Automated deepfake detection," *arXiv:2106.10705*, 2021.

Fine-Grained Containerization for Data Authentication, Security, and Portable Workflows

Sandia National Laboratories, Univeristy of Tennessee, Knoxville
Jay Lofstead (gflofst@sandia.gov), Paula Olaya (polaya@vols.utk.edu),
Andrew Younge (ajyoung@sandia.gov), Michela Taufer (taufer@acm.org)

Position: Data workflows have long dealt with offering data sets for others to use as a foundation for determining trust in outcomes. In general, data sets are offered either as a file set or as an archive and the processing is according to the user needs. However, precisely and uniquely identifying a version of a data set is manual and arbitrarily defined. At best, a community has a stable publication and management process with carefully tracked versions and dates. At worst, there is no reliable way to determine what version of a data set was used.

Container technology has revolutionized the application space as it enables precisely packaging particular versions of libraries and software offering a mostly effective way for people to run the same tools again. However, the data process is still subject to the issues described above.

Our work has focused on expanding the idea of containers from strictly application encapsulations into true "containers". This means that they are simply vessels for holding something. The file system in a file format for container images offers tremendous potential for handling data as well as applications. Through extending container runtimes and how container systems work, we believe we can take advantage of the encapsulation, along with digital signatures, to offer true, unique, precise identification, customizable security envelopes, and provenance tracking for data products.

Background: Application runtime management has evolved extensively with the introduction of containers by enabling each application instance to have a private space for any dependencies eliminating system conflicts due to varying or missing library versions. However, all of the attention has focused on strictly how to make the application management efficient and effective. Tools like digital signatures, such as is offered by Singularity, offer ways to certify that a container is genuine and using the associated hash value reveals that it is the particular one used for a particular purpose.

The general assumption is that the data is transient for the vast majority of these environments and is simply something used to generate the next model or response. Long-term data management for purposes such as reproducibility, authenticity, and provenance has never been a priority.

Complicating this environment, mounting multiple containers into a single space is not always easy. In general, a single container is run and various local system resources are mapped into the container namespace to provide the variation required for each run. The application within the container can monitor a mapped-in directory for new data to process or wait for the next data packets to arrive via a network connection or other interface. Not to discount how successful this model is, but small extensions can offer a new model with interesting, useful features.

Data Pallets for Containerized data: Our previous work called Data Pallets [1] and the follow-on thesis work [2] is exploring this idea. In Data Pallets, we demonstrated the idea of using containers as a way to store the data generated for a workflow output as a first step towards a top-to-bottom containerized environment for both applications and data.

In this work, we enabled the capability of digitally signing the container through the use of Singularity's digital signature functionality offering a way to trust that a data set is genuine. Further work is examining automatic signing and differential privacy to reduce the barrier to these capabilities as well as controlling who can access sensitive data, such as detailed identity information for the application scientist. We recognize that politicized areas, such as climate research, want to widely release data sets for scrutiny, but the scientists need to protect their identity except to select users. By using differential privacy, we can encode both the data and identity information in the container, but only reveal the pieces a user has authority to access. With the public digital signature from an institution, the data authenticity can be verified while managing potential tampering. Ideally, since the container can be mounted in the local system, this in no way negatively affects data use.

Olaya's thesis work has extended these ideas demonstrating with scientific applications with various production workflows. The standard configuration uses an input data set, a runtime configuration, an application, and a generated output data set. Additionally, it incorporates automatic provenance generation and annotation to track *h*ow data was created.

Recently we held the First Workshop on Reproducible Workflows, Data, and Security (ReWorDS) at

eScience 2021. The theme of this workshop, as the title suggests, is to explore the ideas of portable workflows, accurate and portable provenance tracking, and workflow data management. The workflow data management is strongly focused on issues around how to manage data for authenticity and validation as well as tracing data creation from the final product back to the source code via the intermediate processing steps, compilers, third party library, system settings, and anything else that can affect the final generation. As mentioned above, we sought to incorporate security as a fundamental aspect of the workshop to raise awareness that we have a dual problem of not just data authenticity, but also of the privacy issues as described above. Based on the success of the first instance, we will hold a second instance at eScience 2022.

Our ongoing work is extending the concepts we promote by optimizing how containers work and examining the provenance requirements capabilities. For example, containers are write-only constructs. At best, a container can be created based on the final set of files placed on storage. In order to use this in the next workflow component, it is necessary to extract the data into the native storage system breaking any chance of automatically tracking the provenance information.

Impact The environment we are striving to enable offers many profound impacts on the practice of science and more generally any workflows incorporating data articles:

1. **Improved Data Longevity**. By invisibly and portably annotating data with full provenance information, discovering not just want the data represents, but how it was created is now possible. Since it is attached to the data and does not use fragile information, such as machine names and absolute or relative directory paths to identify artifacts, this provenance is now fully portable to other systems.

2. **Automatic Data Authenticity**. By using a data packaging mechanism that can hold a digital signature, we can automatically attach such a signature enabling others to verify the data set authenticity.

3. **Privacy and Security**. Our proposed solution offers privacy and security guarantees to ensure that the system can allow users to protect sensitive and proprietary information without compromising The ability to share results, where possible, in a meaningful way. A system such as the Singularity Image Format that offers multiple partitions within the container image file enables storing different data using different access keys without negatively impacting the data usability. Simply by providing the keys as part of the runtime we will be able to mount the various partitions that can be unlocked and they become visible, trusted data in the environment.

4. **Independent, Portable**. By using a user-space construct, such as containers, we avoid requiring any system-specific extensions for our functionality. For example, a common approach for tracking metadata associated with a data set is via a system-level database. However, this requires intercepting all potential actions that can affect the data location, names, or contents to maintain the proper provenance. Attaching this in container partitions eliminates this system dependence and gains the same advantages. While data containers are the focus of the work, consider that an application is little more than a set of source code (data) processed by an application (a compiler) according to an input configuration (build setup) generating a data product (executable). All applications and data can be handled the same way.

5. **Support for All Applications**. Given the general approach of using containers, we have demonstrated this using gnuplot in the Data Pallets work and then with SOMOSPIE (hydrology) and visualization tools in the thesis work.

References

[1] J. F. Lofstead, J. Baker, and A. Younge. Data pallets: Containerizing storage for reproducibility and traceability. In M. Weiland, G. Juckeland, S. R. Alam, and H. Jagode, editors, *High Performance Computing - ISC High Performance 2019 International Workshops, Frankfurt, Germany, June 16-20, 2019, Revised Selected Papers*, volume 11887 of *Lecture Notes in Computer Science*, pages 36–45. Springer, 2019.

[2] P. Olaya, J. F. Lofstead, and M. Taufer. Building containerized environments for reproducibility and traceability of scientific workflows. *CoRR*, abs/2009.08495, 2020.

Negotiating Privacy Utility Trade Offs under Differential Privacy

Ashwin Machanavajjhala, Tumult Labs & Duke University, ashwin@tmlt.io
Gerome Miklau, Tumult Labs & UMass Amherst, miklau@tmlt.io
Michael Hay, Tumult Labs & Colgate University, michael@tmlt.io

Abstract: We describe the challenge of negotiating privacy-utility trade offs that many federal agencies face when releasing sensitive data for research, policy making and social good. We present a case study of deploying differential privacy at the IRS and highlight the importance of this negotiation. We conclude with a discussion of open challenges in this area.

Problem & Use Case: Many US federal and state agencies release data products based on highly sensitive and regulated microdata about individuals, establishments and transactions. These data products span a variety of domains -- demographic, administrative, health, finance, energy, etc. -- and are used for diverse purposes like scientific research, policy making, enabling transparency, and are essential for high stakes allocation problems like funds allocation, apportionment, disaster recovery and redistricting. The rise of data science and AI methods as well as new mandates for increased data sharing have strained the existing processes for assessing the privacy loss encountered by fine-grained data releases. Hence, we need new methods to assess privacy and utility, and user-friendly tools to navigate this tradeoff.

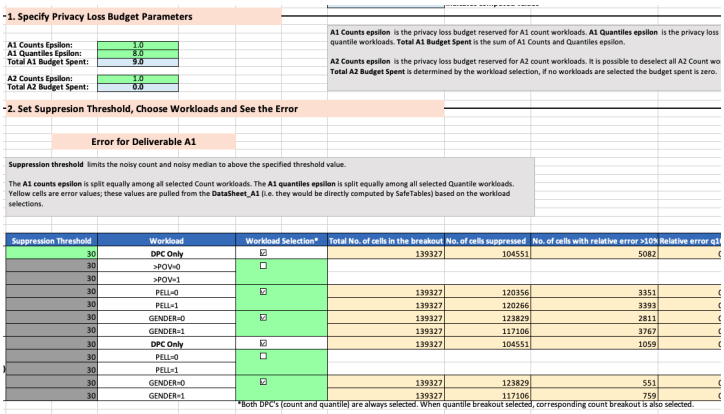
Case Study: Cross-agency data sharing to support educational accountability: The U.S. Internal Revenue Service (IRS) manages hundreds of millions of personal records from individuals and corporations. The IRS needs to be able to release statistical summaries to other U.S. government agencies to advance research and support evidence-based policymaking. However, these data releases must adhere to stringent privacy constraints encoded in U.S. law. For instance, the IRS releases income data of students in support of the Department of Education (ED)'s College Scorecard website. The website informs college students about post-graduation outcomes like median salaries broken down by institution and degree program. Starting with IRS tax returns of 5-8 million students, tens of millions of income statistics are generated to power the College Scorecard website. Starting in 2020, the IRS asked Tumult Labs to design and implement a unique solution to safely release this data.

The Need for Differential Privacy: Prior to 2020, IRS had used legacy disclosure avoidance techniques to release data to ED. However, the suppression and ad hoc distortion techniques resulted in over 70% of the data being suppressed. Moreover, as ED asked for more detailed and overlapping statistics, it was becoming impossible to verify the privacy ensured by their suppression-based approaches. Hence, IRS decided to adopt differential privacy in 2020.

Differential privacy (DP) has emerged as a mathematically proven answer to getting the most out of sensitive data. No matter the kind of data -- structured administrative records, health records, financial transactions, images and video collected by cameras -- or the kind of analysis performed on the data -- releasing aggregate statistics, ML, synthetic data -- differentially private algorithms can be used to protect the privacy of individuals. DP also provides a measure of privacy loss (often called *epsilon*) that quantifies the worst case privacy loss an individual may

experience and that can be used to track privacy loss across multiple releases of sensitive data. Thus, DP provides a solid foundation and language for negotiating privacy-utility trade offs.

Negotiating Privacy-Utility Trade Offs: The development of DP algorithms is a major focus of academic research across computer science and statistics. And, startups like Tumult Labs, initiatives like OpenDP and companies like Google are developing hardened libraries for DP. But algorithms and software alone cannot solve the problems faced when deploying DP in high stakes situations like in the IRS-ED use case. The most important challenge is the *privacy-utility negotiation.* In the IRS-ED use case there are two parties: the data steward (IRS), who is responsible for ensuring the privacy of the individuals and the data analyst (ED), who cares about the utility of the released data. The analyst, of course, would like the most detailed statistics with the highest accuracy. But, theoretical results like the Fundamental Law of Information Recovery state that if enough statistics are released with sufficient accuracy then one can reconstruct the underlying sensitive database. The US Census Bureau famously performed such an attack using the published results of the 2010 Decennial Census.


1. Specify Privacy Loss Budget Parameters

A1 Counts Epsilon: 1.0
A1 Quantiles Epsilon: 8.0
Total A1 Budget Spent: 9.0

A2 Counts Epsilon: 1.0
Total A2 Budget Spent: 0.0

A1 Counts epsilon is the privacy loss budget reserved for A1 count workloads. A1 Quantiles epsilon is the privacy loss quantile workloads. Total A1 Budget Spent is the sum of A1 Counts and Quantiles epsilon.

A2 Counts epsilon is the privacy loss budget reserved for A2 count workloads. It is possible to deselect all A2 Count wo
Total A2 Budget Spent is determined by the workload selection, if no workloads are selected the budget spent is zero.

2. Set Suppresion Threshold, Choose Workloads and See the Error

Error for Deliverable A1

Suppression threshold limits the noisy count and noisy median to above the specified threshold value.

The A1 counts epsilon is split equally among all selected Count workloads. The A1 quantiles epsilon is split equally among all selected Quantile workloads. Yellow cells are error values; these values are pulled from the DataSheet_A1 (i.e. they would be directly computed by SafeTables) based on the workload selections.

Suppression Threshold	Workload	Workload Selection*	Total No. of cells in the breakout	No. of cells suppressed	No. of cells with relative error >10%	Relative error q1
30	DPC Only	☑	139327	104551	5082	
30	>POV=0	☐				
30	>POV=1					
30	PELL=0	☑	139327	120356	3351	
30	PELL=1		139327	120266	3393	
30	GENDER=0	☑	139327	123829	2811	
30	GENDER=1		139327	117106	3767	
30	DPC Only	☑	139327	104551	1059	
30	PELL=0	☐				
30	PELL=1					
30	GENDER=0	☑	139327	123829	551	
30	GENDER=1		139327	117106	759	

*Both DPC's (count and quantile) are always selected. When quantile breakout selected, corresponding count breakout is also selected.

So a key question arises: how much of an analyst's "wish list" of statistics can be answered, and with what accuracy, for reasonable settings of privacy? To address this question, we at Tumult developed an "Interactive Decision Support Tool" that shows (a) the set of statistics that the analyst wants to release, (b) privacy settings for a DP algorithm, and (c) accuracy achieved by the DP algorithm to answer these statistics. The analyst could use this tool to (a) set privacy loss parameters (epsilon) and (b) interactively include and exclude statistics in the final data product and visualize the error that they could expect in the release. In the IRS-ED use case, the tool was used by the data steward (IRS) to estimate the effect of releasing statistics on privacy loss, and the data analyst (ED) used the tool to identify the set of statistics that can be released with reasonable error under a privacy constraint.

Conclusion: As more organizations deploy differential privacy, we believe we need tools to permit and ease the negotiation between data stewards and analysts. We highlighted one case study from federal statistical agencies where such tools were crucial. There are several interesting technical and research challenges that arise when enabling such a negotiation. We highlight a couple below: First, there may not exist ways to estimate tight error bounds for a DP algorithm without actually measuring the error on the real data. This is especially true when releasing statistics like quantiles where error is dependent on the data distribution. So one should be careful about the error statistics that are released if the analyst is untrusted. Second, while epsilon is a nice way to quantify privacy loss, it is not very interpretable by non-experts. Further research is needed to contextualize the privacy loss parameters in DP.

AN INFORMATION-THEORETIC VIEW OF LEARNABLE PRIVACY-UTILITY TRADE-OFF FOR SCIENTIFIC DATA

SANDEEP MADIREDDY*, PRASANNA BALAPRAKASH

Argonne National Laboratory

Topics: Cybersecurity and Privacy of Scientific Computing Ecosystems

Challenges and Opportunity: In this age of "Big-Data", the scientific discovery has undergone a new reckoning. Large volume of data are being collected and processed at several of the DOE user facilities such as APS/light source and leadership-class high-performance computing systems, as well as in application areas such as fusion energy, and cosmology. However, due to small amount of sensitive/private information within these datasets, they are not being able to be shared broadly across the community. Broad adoption, however, is key to make transformative progress both for improving the facilities and scientific discovery. For example, the compute facilities run a diverse set of applications each day which interact with the compute, storage, and network systems; this data can give a wealth of information on designing the future systems. However, they have to contain a sensitive user information. In another example, the light source facilities such as the Argonne Photon Source are used to characterize several materials each day. In both these cases, able to obfuscate the sensitive information and share the data without any sacrifice in its utility will be the key for scientific discovery. In an ideal scenario, the user should also be able to choose the degree of trade-off between privacy and utility based on the application and user-specific needs. On the other hand, machine learning methods are increasingly being used in scientific applications [1]. In addition to the need for privacy-preservation, this explosion in scientific data collection and the adoption of machine learning poses new privacy threats: the public datasets may be deanonymized with only a few queries [4], a machine learning model's parameters and outputs can reveal sensitive information about the dataset [5].

To this end, privacy and security have been traditionally seen from the cryptography point-of-view, where the goal is to prevent unauthorized information extraction from communication over insecure channels. However, the privacy goal in many of these data-driven applications is not to ensure zero information leakage but to *ensure a provable level of privacy, while achieving a target level of utility*. The adversary in this case seeks to observe the shared data and tries to infer private information from it. Hence, designing techniques that can obtain application-specific privacy-utility trade-offs is a fundamental challenge that needs to be addressed by privacy research [3]. Privacy is usually ensured via a privacy-preserving mechanism, which is an algorithm that randomizes data or its functional transformation to prevent unwanted statistical inferences. Privacy metric on the other hand, is used for evaluating the effectiveness of the privacy-preserving mechanisms. Here, we will discuss an information-theoretic view on privacy-preserving mechanisms and metrics.

Information-Theoretic Mechanisms for Privacy-Preservation: In recent days, information theory has emerged as a popular lens through which theoretical properties of machine learning systems are being studied extensively, which has also been highlighted in the recent workshop on data reduction for science [2]. This can be attributed to the strong mathematical rigor of information-theoretic quantities as well as the recent advances in their estimation in high-dimensions [7]. From an information-theoretic point of view, the problem of privacy-utility

E-mail address: smadireddy@anl.gov.

trade-off can be posed as follows: let $X \in \mathcal{X}$ be the observed data, $S \in \mathcal{S}$ be the application-specific sensitive/private attributes (e.g., user information), and $Y \in \mathcal{Y}$ be the shared data. The goal of information-theoretic privacy is to determine $P(Y|X)$ as a privacy-preserving mechanism to share data without leaking private information. This can be posed as a constrained optimization problem, where we seek to find a representation S that has minimal mutual information (MI) with Y while be maximally informative about X: $\min_S \mathcal{I}(S;Y)$ s.t. $\mathcal{I}(X;S) \geq C$. However, this optimization problem is usually intractable due to the difficulty of computing the MI integrals. Nonetheless, it is possible to carry out an approximate form of this optimization by maximizing a tractable lower bound on $\mathcal{L}_{PF}$ with the IB Lagrangian: $\mathcal{L}_{PF} = \mathcal{I}(S;Y) - \beta\mathcal{I}(S;X)$. This is known as *Privacy Funnel* approach, and has strong connections to the popular *Information Bottleneck* [6], information-theoretic rate-distortion and remote source coding. The objective of privacy funnel approach can also be extended to encompass maximum information leakage formulations [3].

Due to the pure data-driven formulation of this approach, it can be challenging to impose the application-specific privacy constraints, partly because estimation of the mutual information can be notoriously difficult, especially in high dimensional datasets that are common to scientific data and machine learning. However, recent theoretical advances in the estimation of mutual information metrics in high dimensions, development of alternative entropy formulations [7] where, scalable and differentiable MI estimation is being done using parametric models such as deep neural networks to construct variational MI estimators and estimate MI lower bounds, faster Bayesian inference through advances in variational inference, Markov chain Monte Carlo techniques, and probabilistic programming languages provides hope for accurate and efficient information-theoretic privacy estimation.

Information-Theoretic Metrics for Privacy-Preservation: The privacy leakage is defined as the amount of information about the private feature that can be observed through shared data. Information-theoretic privacy leakage quantities are used in both differential privacy (DP) and the information-theoretic privacy (mentioned above) alike. In differential privacy, the loss metric is used to measure the divergence between the original data and the perturbation produced by a privacy-preserving (randomized) mechanism. Strong guarantees given by DP and its variants are hinged on the ability to calculate this divergence in high dimensions. For example, the Rènyi DP uses the Rènyi divergence, derived from an alternative information-theoretic entropy. This signifies that broad applicability of the information-theoretic measures and its role as a unifying aspect of various privacy-preserving approaches.

References

[1] N. Baker, et al. Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence. USDOE Office of Science (SC), Washington, DC, 2019.

[2] S. Klasky, et al. "Data reduction for science: Brochure from the advanced scientific computing research workshop",Tech. rep., USDOE Office of Science (SC)(United States) (2021)

[3] Hsu, Hsiang, et al. "A Survey on Privacy from Statistical, Information and Estimation-Theoretic Views." IEEE BITS the Information Theory Magazine (2021).

[4] M. Abadi, et al. "Deep learning with differential privacy," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 308–318.

[5] A. Narayanan and V. Shmatikov, "Robust deanonymization of large sparse datasets," in 2008 IEEE Symposium on Security and Privacy (sp 2008). IEEE, 2008, pp. 111–125

[6] Goldfeld, Ziv, and Yury Polyanskiy. "The information bottleneck problem and its applications in machine learning." IEEE Journal on Selected Areas in Information Theory 1.1 (2020): 19-38.

[7] Yu, Shujian, Luis Sanchez Giraldo, and Jose Principe. "Information-Theoretic Methods in Deep Neural Networks: Recent Advances and Emerging Opportunities

[8] Madireddy, Sandeep, et al. "A modular deep learning pipeline for galaxy-scale strong gravitational lens detection and modeling." arXiv preprint arXiv:1911.03867 (2019).

Addressing the Limitations to Distributed Learning Containing Sensitive Data

J. Robert Michael[1], Christopher Stanley, Ryan Adamson, Olivera Kotevska
Oak Ridge National Laboratory

Topic: Federated Learning, Differential Privacy, Distributed Computing, Secure Data Architectures

Challenge: Artificial Intelligence (AI) and Machine Learning (ML) have been used to answer questions about data where previous solutions have failed. AI/ML are regularly used to analyze sensitive data in secure environments and DOE is performing these types of analyses at scale for a variety of distinct partners. While AI/ML continue to gain traction with their increasing ability to answer questions accurately and efficiently about data, it becomes difficult to identify at what point the model, itself, may constitute sensitive data. This challenge primarily arises because AI models begin to intimately 'learn' aspects of the training data itself that can be unintentionally leaked in downstream applications. Whether by membership inference attacks, model inversion, gradient stealing, or novel vectors yet to be created, it is possible for a potential attacker to reconstruct sensitive information from the underlying data [1].

This 'model as sensitive data' problem is of particular interest when one considers the need to train large scale models across disparate data that cannot be mixed, for which Federated Learning (FL) is an attractive solution. FL [2] is an elegant implementation whereby multiple sensitive datasets, which cannot be shared, are leveraged to train a single model by having each data owner train iterations of the model on their local data and share only the model updates. This offers implicit security advantages given that multiple datasets are not being mixed and opens the door to multiple DOE partners collaborating in ways which are currently impossible due to restrictions on data sharing. However, the 'model as sensitive data' problem raises new questions regarding the security of FL relative to the success of one federation member's ability to extract sensitive data from another member using only the shared model.

FL is regularly implemented in industry with highly distributed, low power, and centrally governed devices such as phones [3]. These practices are being adapted to research settings where multiple institutions act as clients and iteratively train models. However, while industry performs FL in a centrally secure manner, there is much greater risk with multiple independently managed entities acting as a federation. In the former case, the central authority manages the full stack of clients to govern the compliance and security of devices, ensuring that untrusted third parties cannot gain access to the models. In the latter, each independently operated federation partner maintains its own security posture and the entire federation is only as secure as the weakest link. **The risk of a centrally governed FL stack scales as O(1) with the number of clients while the risk of independently managed partners scales as O(N).**

Opportunity: To securely share sensitive AI and ML models we must investigate the ability to extract information from these models, evaluate the risk, implement technical safeguards, and revisit policy. Examples of paths forward toward to this end are:

- Attempt to extract information from both partial and fully trained sensitive models. While this will not prove the security of models, the use of scalable systems to perform systematic black-box attacks, such as membership inference and model inversion, could show success where other attempts may fail. Vital to this attempt is to utilize world class computational capabilities to ensure that a limitation of resources does not yield a false sense of security.
- Evaluate the ability for a compromised federation partner to extract information from a shared model in an FL setting or successfully implement backdoor poisoning attacks. This is tied to the above opportunity and could inform technical design of FL solutions to mitigate the risk of compromised partners. This will assist with revisiting policy and educating partners on the risks associated with implementations of FL to ensure that data owners are making informed decisions.

[1] Corresponding Author: michaeljr@ornl.gov

- Modify models and model deployment to make them shareable with untrusted parties. Differential Privacy has made advances in this domain, but enhanced privacy is achieved at the cost of reduced accuracy. Further work is needed in this area to deploy models to untrusted parties. Efforts here could advance FL approaches as well as improve our ability to share previously private results with a wider audience for greater scientific impact.

The above examples are among a variety of options that should be pursued before sharing models trained on sensitive data or performing shared model training via FL. This is not an exhaustive list and cannot be, as no work to date optimizes model accuracy, risk mitigation, and computational efficiency. Further investigation is needed into evaluation of current solutions which trade-off between these three goals and creation of novel methods which improve upon them.

Timeliness and Maturity: The timeliness of this topic cannot be overstated. There is an ever-growing number of AI/ML methods today which are becoming both more accurate and easier to train. While this seems like an obvious win, the application of these methods in the medical community and other areas where sensitive data is leveraged are an afterthought in terms of risk assessment and secure deployment.

The combination of the following three problems pose significant risk during this era for shared computing of sensitive data. First, *data* production is outpacing our ability to *analyze* it, particularly in the biomedical field where novel instruments are producing larger amounts of data than ever before. Second, development of generalized AI/ML methods *for analysis* is outpacing our ability to evaluate and *deploy* them for implementation with sensitive data. Third, solutions for *deployment* of AI/ML methods for sensitive data analysis is drastically outpacing our *policy* development. With this in mind, it is time for us to reverse the order of priority here to put the development of *policies* for *deployment* of *analysis* tools using secure AI/ML methods first and foremost to deal with the exponentially growing amount of sensitive *data*.

Aside from these lagging components of the AI/ML development process for secure data analysis, we face new challenges for ensuring security of models due to increasingly sophisticated hardware architectures. Data breaches (both in general and with AI models) could have previously been deemed ‘low risk’ due to the limited computational power which adversaries may have utilized. However, with novel architectures for AI and the dawn of Quantum Computing, defining and evaluating risk for extracting sensitive data from models needs to be revisited in a whole new way.

Finally, the increase in data volume creation is mirrored by the revelation among data owners of its value. Increasingly data owners are refusing to share their data. Rather than for technical reasons (data volume, for example), increasingly this is becoming due to either the inherent value, privacy, or both which the data holds. Based on this, data owners who wish to build robust models now find themselves in the dichotomous position of wanting to keep their data siloed while also collaborating with competing or untrusted entities to heighten the value of their data. This is where technologies like FL are currently employed, but with all the challenges that we face today (and the rapid pace at which these challenges become worse), the time is upon us to revisit and improve upon these tools and policies.

References:

[1] M. Veale, R. Binns and L. Edwards, "Algorithms That Remember: Model Inversion Attacks and Data Protection Law," *Philos Trans A Math Phys Eng Sci,* vol. 376, no. 2133, 2018.

[2] P. Kairouz, *et al* "Advances and Open Problems in Federated Learning," *Foundations and Trends in Machine Learning,* vol. 14, no. 1-2, pp. 1-210, 2021.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson and B. Aguera y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017.

Graph Algorithms for Quality and Security of Scientific Computing Ecosystems

Pablo Moriano (moriano@ornl.gov), Oak Ridge National Laboratory, Oak Ridge, TN, USA

Topic: Graph algorithms, AI-driven software, scientific computing ecosystems, trustworthy computational science

Challenge: Today's scientific software has become increasingly complex. Scientific software requires the contribution of thousands of developers, include tens of millions of lines of code, and require advanced mathematical abilities, strong programming skills, and a good understanding of the science problem being solved. This complex collaboration landscape makes (1) developers more likely to introduce defect-prone changes that lead to software faults; and (2) advanced adversaries well-positioned to introduce vulnerabilities that could result in exploits. On the one hand, software defects are detrimental to software quality and have a negative impact in the scientific enterprise and subsequent ecosystems, especially when they lead to software failures that produce incorrect results. On the other hand, vulnerabilities have the potential of threaten nation's energy, economic, and security infrastructure undermining the trustworthiness of computational results. Determining when these defect/vulnerable-prone changes are introduced has proven challenging, and using traditional machine learning (ML) methods to make these determinations seems to have reached a plateau [1]. Thus, there is a huge incentive to advance the state-of-the-art on AI-driven and, in particular, graph algorithms to improve the quality and security of scientific computing ecosystems [2]. Significant use cases for quality and trustworthy scientific computing ecosystems include the following:

- Characterization and early detection of defect/vulnerable-prone changes in open source software used in scientific software.
- Characterization and early detection of defect/vulnerable-prone changes in the software used by scientific instruments.

Background: Previous research on software quality assurance focuses on either module-level [3] or Just-In-Time (JIT) defect prediction [4]. The module-level approach uses ML models trained on historical data obtained from software characteristics, including code churn, change metadata, and complexity metrics. Defect prediction models detect defect-prone software modules (e.g., files, subsystems). Defect prediction models are then used to identify software modules that likely contain faulty code. These models can also help prioritize software quality assurance efforts, such as code reviews and pre-release testing. The JIT approach, in contrast, focuses on change-level defect prediction. This means that the focus is on software changes (i.e., commits) rather than on modules. Among these, JIT has shown important advantages over module-level defect prediction [4].

Current JIT defect prediction models use software characteristics to inform commonly used, supervised ML models. Traditional features used in this task are related to the diffusion, size, purpose, history, and experience dimensions of the changes [4]. Recent models also add context to these features by leveraging the semantic information and syntactic structure hidden in source code [5]. Once this set of features has been computed for the targeted software commits, different ML models are used for JIT defect prediction, including logistic regression and more sophisticated models, such as ensembles and deep learning. Obtaining large amounts of accurate historical data is a prerequisite for good performance in JIT defect prediction. However, this data can be difficult to obtain because the nature of code commits/changes tends to evolve during the development cycle, which can impact the performance of JIT defect prediction [6]. In addition, as shown recently [1], even when using sophisticated ML models, such as ensembles, the achievable performance for JIT defect prediction still has much room for improvement (i.e., it currently reaches about a 30% average F1 score for predicting early exposed defects).

Opportunity: Graphs are a key abstraction for representing complex interactions such as those derived when creating scientific software. Structuring data and knowledge representations using graphs

and leveraging their structural properties have shown to be critical for obtaining better results in tasks such as classification, clustering, and predicting missing data [7]. Incorporating graph-based capabilities to determine when defect/vulnerable-prone changes are likely to be introduced in scientific software is empowered by graph-based ML [8]. The key notion is on representing this complex collaborative landscape through contribution graphs. Contribution graphs are bipartite graphs in which nodes represent developers and modules (source code files) [9]. Edges in the contribution graph capture interactions between developers and modules, thereby representing software changes. Edges in these graphs can be labeled to distinguish clean commits from bug-introducing commits using the Sliwerski-Zimmermann-Zeller (SZZ) algorithm [10]. This approach is novel for JIT defect prediction in that it assigns a probability score to each new code change (i.e., an unlabeled edge in the graph) that indicates the likelihood of that change being defect-prone. This idea can be operationalized using edge classification. Edge classification refers to the problem of classifying unknown edge labels in a graph. Here, the notion of an edge appearing in the future is quantified as a score that measures the likelihood of it being a defect-prone change. Future work will include examining the effectiveness of the proposed approach using an inductive embedding framework, such as GraphSAGE [11]. The approach proposed would benefit from an inductive framework that can efficiently generate node embeddings from previously unseen graph data by aggregating features from node-local neighborhoods, including node attributes that may represent the level of expertise of developers and intrinsic characteristics of source files.

Timeliness or maturity: Reducing the number of software defects and vulnerabilities through quick and automatic identification would improve the quality and security of scientific applications. This is critical in areas where the results obtained by scientific applications are used to ensure national security and inform public policy, as well as in the "smart laboratory." AI-driven algorithms and, in particular, graph algorithms have the potential to help in this endeavor by exploiting the complex nuanced collaboration landscape of modern scientific software development process that was not previously obtainable by traditional ML algorithms. These approaches are also certain to enable new opportunities for DOE scientific computing ecosystems by helping to delineate principles to be integrated more quickly and efficiently in scientific program synthesis.

References

[1] Y. Tian, N. Li, J. Tian, and W. Zheng, "How well just-in-time defect prediction techniques enhance software reliability?" in *QRS*, 2020.

[2] H. Finkel and I. Laguna, "Report of the Workshop on Program Synthesis for Scientific Computing," https://anl.app.box.com/s/z8lw1hs1ypnpawvum39ob9nc0rar0cip, 2020.

[3] A. E. Hassan, "Predicting faults using the complexity of code changes," in *ICSE*, 2009.

[4] Y. Kamei, E. Shihab, B. Adams, A. E. Hassan, A. Mockus, A. Sinha, and N. Ubayashi, "A large-scale empirical study of just-in-time quality assurance," *IEEE Trans. Softw. Eng.*, vol. 39, no. 6, pp. 757–773, 2012.

[5] M. Kondo, D. M. German, O. Mizuno, and E.-H. Choi, "The impact of context metrics on just-in-time defect prediction," *Empir. Softw. Eng.*, vol. 25, no. 1, pp. 890–939, 2020.

[6] S. McIntosh and Y. Kamei, "Are fix-inducing changes a moving target? a longitudinal case study of just-in-time defect prediction," *IEEE Trans. Softw. Eng.*, vol. 44, no. 5, pp. 412–428, 2017.

[7] R. Stevens, V. Taylor, J. Nichols, A. B. Maccabe, K. Yelick, and D. Brown, "Report on the Department of Energy (DOE) Town Halls on Artificial Intelligence (AI) for Science," https://anl.app.box.com/s/f7m53y8beml6hs270h4yzh9l6cnmukph, 2019.

[8] J. Bryan and P. Moriano, "Graph-Based Machine Learning Improves Just-in-Time Defect Prediction," 2021, arXiv:2110.05371.

[9] P. Moriano, J. Pendleton, S. Rich, and L. J. Camp, "Insider threat event detection in user-system interactions," in *CCS Workshops*, 2017.

[10] J. Śliwerski, T. Zimmermann, and A. Zeller, "When do changes induce fixes?" *Softw. Eng. Notes*, vol. 30, no. 4, pp. 1–5, 2005.

[11] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NIPS*, 2017.

A Platform For Controlled Access to Heterogeneous Computing for High Energy Physics

Andrew Norman (anorman@fnal.gov), Fermilab

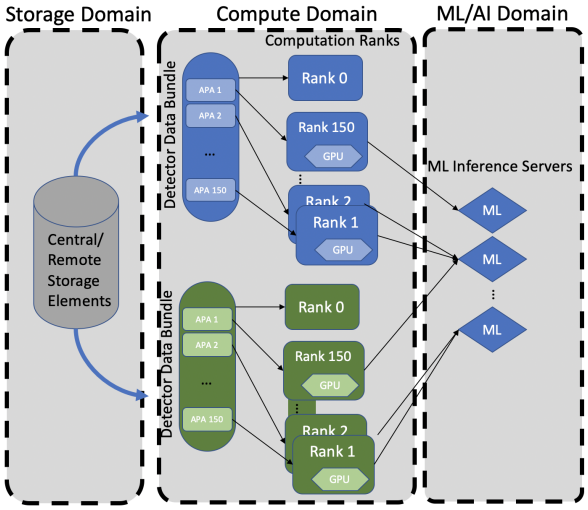
The high energy physics (HEP) community features scientists from a diverse international community that focus on research related to the fundamental structure of the subatomic world and on how that structure influenced the development of our universe. The HEP community is organized into large 1000+ research collaborations which operate the world's most advanced particle beams and experimental detector facilities. These collaborations focus on fundamental particle physics at the very highest energy scales currently available at the world's particle accelerators, and on precision measurements which focus on specific structures of the subatomic world. The data taken at these facilities, at the Large Hadron Collider in Geneva Switzerland, at Fermi National Accelerator Laboratory in the United States, and at other sites both in the U.S. and around the world need to be analyzed using computationally complex techniques ranging from simulation of the experimental detectors, to advanced statistical computations, to AI/ML based techniques for transforming, reconstructing, and identifying the data into representations of the underlying subatomic particle interactions. These data analysis workflows will present a near-exascale data access and analysis challenge over the next decade as a new generation of experiments and increased accelerator beam powers become available. This data analysis challenge can not be meet with traditional, localized, homogeneous computing resources, but instead will require access to a highly diverse and distributed set of computational and storage resources which can only be met through a combination of high performance, high throughput, and advanced acceleration computing. Bringing these resources together to bridge their access technologies, security strategies, and architectural diversity is the challenge we need to consider.

In particular, the HEP community has the need to execute workflows from different major experimental collaborations (CMS, DUNE, Rubin Observatory), which differ significantly in their underlying software structure, their needs to access data sources (e.g. databases holding experimental apparatus information, data streams of the measurement data, auxiliary calibration measurements, etc…) and their access to local accelerators and remote accelerators (alongside different topological mappings of computational the ranks to those accelerator systems.) Each of these workflows requires special security considerations owing to the manner in which data are accessed and distributed between components. **In the realm of HPC computing facilities, the differences in data access security, workflow submission and authorization, and network isolation of HPC site components limit and prevent the HEP community from effectively leveraging resources at specific sites, prevent co-scheduling of resources at sites and across site boundaries, and convolute data access/authorization and transmission between experimental and computational facilities.**

We outline here the combined use case for HEP computing, which has been in use through the HEPCloud scientific computing portal. The HEPCloud portal system uses an intelligent "Decision Engine" to compare workflows submitted by HEP scientists to resources that are available to the research from DoE HPC facilities, Open Science Grid Enclaves, Grid computing facilities at national labs and major universities, and commercial cloud resources. The HEPCloud Decision Engine then performs an optimization based on various "cost" metrics and makes a decision on where to provision resources across all the available sources. An HTCondor based queueing system matches the queued workflows to the provisioned resources as they become available using a "virtual cluster" model which is an aggregation of resources from across all the available sites and resource types. This model works extremely well for optimizing the HEP

community's use of diverse resource pools, since it can grab in an on demand fashion the type, quantity and configuration of resources that a particular workflow or computation requires.

However, this model has difficulty navigating computing ecosystems where the authentication, authorization, or trust system differs significantly between resource types, sites, or communication channels. This needs to be addressed, and a system for common authorization and for trusted data sharing needs to be established that the HEP community can use at the top level of the portal, which can then be propagated down to the site specific resources. In a similar vein, for co-scheduling of resources to work properly a method for multi-system or ecosystem wide authentication/authorization needs to be checked and provisioned against, and then system of trust needs to be instantiated between the component's communication channels to peremiti efficient data sharing.



As a concrete example of proposed HEP workflow use cases, we consider the DUNE neutrino experiment's data processing model. In this model large (~6 gb) readouts of the experiment's detectors are to be analyzed by a number of physics algorithms, which include a series of ML/AI inference evaluations. Each readout is independent and can be analyzed in parallel with other readouts (typical execution places a few thousand readouts in flight at a given time with the problem shown to potentially scale into the low 10,000's) . The detector data to be analyzed exhibits a natural 150-way top level data parallelism which allows it to be split across multiple computational nodes (this also splits the memory footprint), and the algorithms are threadable allowing for additional parallelism at they node level. At the lowest level the analysis performs a large number of Fourier style transformations and data manipulations which can be highly accelerated by GPU techniques (node local), and the code needs to be able to make external calls to a preconfigured inference engine. However the execution time of the code varies dramatically between the CPU only computations, the GPU accelerated computations and the ML inference evaluation. As a result the workflow would ideally be executed using the following topology: 1 data atom split across 151 ranks (with a rank 0 being used for reduction operations and other aggregation operations) with each rank running O(64) threads and having access to a local GPU accelerating. Each rank then additionally would have a communications channel to an external ML inference server, where multiplexing is possible with ratios of approximately 100:1. The pattern is then repeated for each concurrent data atom. The difficulty with this type of model and workflow is the security concerns in crossing between they different storage, compute and ML domains, along with the mechanisms that are needed to securely pass data across those boundaries and then back to the science teams. We propose a demonstrated need for research into these problems of distributed access and co-scheduled computing as a topic for consideration.

https://computing.fnal.gov/hep-cloud/
Holzman et al. Comput Softw Big Sci (2017) 1:1 dio:10.1007/s41781-017-0001-9
Mhashilkar et al. EPJ Web of Conferences **214**, 03060 (2019).

Trustworthy Computing with Edge Devices

Authors: Peter Pirkelbauer[1], Chunhua Liao

Lawrence Livermore National Laboratory, Livermore, CA.

Topic-Area: Computing with Edge Devices

Introduction

Future scientific experiments will be conducted on large-scale computing environments connected to edge devices. This combination of scientific experiments and edge devices poses significant challenges and risks. By definition, an edge device is operating in a less-controlled and outside-facing environment. Due to economic factors, edge devices typically operate under tight resource constraints in terms of processing speed, available memory, and power budget. Also, edge devices are connected to some computing environment through the internet. Such a setting makes edge devices a welcome target for a large number of attacks scenarios.

An important requirement for harnessing edge devices for scientific, large-scale experiments is that the data reported by an edge device can be trusted. The absence of trustworthy data threatens the validity of any scientific experiment. The development of techniques and tools that simplify the secure deployment and operation of edge devices is critical to the successful integration of edge devices into scientific computing.

Common Cybersecurity Threats

Edge devices communicate with a central system through the internet. This makes them vulnerable for many known cyber attack scenarios, including DDOS attacks, repeated probing, exploiting common software vulnerabilities with the goal to install malicious software (*e.g.*, traffic monitors, backdoors, data theft, data fabrication).

Even if verified or well tested software is used, attackers may be able to launch side channel attacks, such as timing attacks, to obtain critical information in the form of cryptographic data that can be used to infiltrate a device.

Use Cases

The notion of *digital twins* describes a dual system where a physical instance is connected with a digital model of the physical world. The digital twin receives real-time input from sensors on connected edge devices and updates its state. Digital twins can be used to play through what-if

[1] Corresponding Author, pirkelbauer2@llnl.gov. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL-TR-827852.

scenarios and to report observed data that falls outside an established norm. According to Taylor et al. (2020) "Digital twins can become another tool to be used by scientists to advance science discovery." To enable the practical use of digital twins for scientific discovery, it is imperative that the data produced by edge devices can be trusted.

Research Directions

The security of edge devices is an urgent issue that needs to be addressed. No single solution will be able to address all concerns related to the security and integrity of an edge device. Thus it is desirable that protection mechanism can be composed to enhance the security. Issues that need to be addressed include:

- Development of scalable monitoring techniques for resource-constrained systems. Monitoring software may guard against common software vulnerabilities, or perform statistical analysis that an edge devices software is operating within an established norm.
- Automated tools that rewrite software (source and binary) into more secure forms.

To solve these problems further research is needed. Possible research directions for software-oriented solutions include:

- Hybrid static and light-weight dynamic analysis. Compile-time analyses identify code locations that potentially exhibit security vulnerabilities (*e.g.*, memory accesses) whereas dynamic checks will be used to guard these locations at runtime. To deal with a resource constrained environment, sparse instrumentation techniques that bound the overhead in terms of execution time, memory use, and power consumption will need to be developed.
- Automatic source-code transformations to make software more secure against side-channel attacks. Side channel attacks often exploit differences in runtime that stem from branching or caching behavior. By automatically rewriting critical code (*e.g.*, by eliminating branching, delay randomization) software can be made more secure against such attacks, Similar techniques can be explored to guard against side-channel attacks that monitor a device's power consumption.
- Study the composability of technqiues that address a subset of concerns.

Reference

Valerie Taylor, Xingfu Wu, Zhiling Lan: Digital Twins for Future Scientific Discoveries, ORAU Whitepaper, 2020. https://www.orau.gov/2020doe-coi/whitepapers/lotf2020-paper124.pdf

SKETCHING ALGORITHMS IN DISTRIBUTED SYSTEMS

ARVIND PRASADAN

ABSTRACT. In this position paper, we discuss exciting recent advancements in sketching algorithms applied to distributed systems. That is, we look at *randomized* algorithms that simultaneously reduce the data dimensionality, offer potential *privacy* benefits, while maintaining *verifiably* high levels of algorithm accuracy and performance in multi-node computational setups. We look at next steps and discuss the applicability to real systems.

1. MOTIVATION

A distributed system is one wherein the data and/or processing is spread over several devices. For example, we might imagine a physically distributed sensor network that is collecting data at various locations; the power grid, with multiple stations generating, transforming, and distributing power, and several millions of houses and businesses receiving it; or, computer simulations that generate so much high dimensional data that they cannot be stored or processed on a single machine, among other applications.

In this position paper, we focus on the setting in which the quantity and dimensionality of the data is prohibitive. In particular, whether the distributed systems is physically present or not, we have no choice but to perform computations in a distributed manner—without *full data access*. However, even with the distributed setup, it may be infeasible to perform computations on all of the data as is. For example, a simulation of a molecular interaction might generate terabytes of data in seconds or a simulation of a climate phenomenon might require hundreds of nodes and generate terabytes of data for every hour of time simulated. Beyond the data generation, several algorithms (e.g., regression or prediction) might require passing of data between computational nodes. If the data size is large, this can quickly become a bottleneck. A *scalable* solution in this setting would necessarily reduce the scale and size of the data as well as that of transmissions between nodes.

Additionally, we also focus on the setting in which the data may be sensitive and may not be transmitted. For example, simulations studying the properties of key national infrastructure might generate highly classified data. Or, data that contains personally identifying information might also be sensitive or non-transmittable for legal reasons [6]. Alternatively, to preserve data fidelity and guard against malicious alterations, data access and transmission might be limited. It might be desirable to perform distributed computations that simultaneously reduce transmission or *privacy risks* in a manner analogous to federated learning.

2. A BRIEF SURVEY OF SKETCHING ALGORITHMS

Sketching is a probabilistic or *randomized* data compression technique that enables us to work with a smaller dataset. In its simplest form, p-dimensional data vectors are multiplied with a $d \times p$ random matrix to produce d-dimensional vectors, where d is generally much smaller than p. That is, we may view sketching as applying a random projection to reduce the dimensionality of the data. A sketching algorithm is one that employs sketching as a preprocessing step. For example, we might imagine a sketched linear regression or sketched covariance estimation wherein sketching greatly speeds up computation. Of course, nothing in life is free and this dimensionality reduction comes at a cost: sketching is a randomized or probabilistic procedure, and its output is hence random. Nonetheless, there are theoretical guarantees about how random or how variable the output will be, even if there is additional uncertainty that is introduced. Averaging over multiple sketches can verifiably reduce this uncertainty, and if the dimension reduction is high enough, can still be a massive computational speedup.

In the distributed setting, there have been several recent exciting advancements. One recent work proposed and analyzed a distributed sketching setup for linear regression wherein each node has a different sketching matrix and the final estimate is done by averaging the estimates from each node [1]. That is, a smaller sketched least squares problem is solved at each computational node and the results are aggregated at the end—this is similar to the federated learning paradigm [4]. The authors found that for appropriate choices of the sketches and sufficient amounts of data at each node, they were able to derive *sharp, exact error bounds* for the error relative to the un-sketched estimator. Moreover, they also found that this system offered strong *differential privacy guarantees*, i.e., that the usage of different sketches at each node and the central aggregation led to data privacy guarantees. In such a system, giving outside parties query access or model access would still protect the data that the model was trained on.

Another recent work studies a distributed sketching setup for covariance estimation [3]. Here, the data are assumed to be extremely massive to the point that storage on a single node is impossible and that communication between nodes is a major bottleneck. Once again, sketched estimators at each node are aggregated at the end. The authors found a provably lower communication overhead, that is, better *scalability*, while maintaining a boundedly low error relative to the un-sketched estimator or relative to an estimator on a down-sampled dataset.

In a related work, a group showed that a distributed sketching procedure applied to stochastic gradient descent led to better communication efficiency [2]. That is, the data are once again distributed across several nodes and some function is being optimized via stochastic gradient descent. The authors showed that transmitting sketches of the gradient reduced communication overhead while maintaining good performance. In critical systems, we might imagine a hybrid approach wherein a generic sketching procedure like this is used to seed, initialize, or speed up convergence by getting 'close enough', and then a more specialized sketched or a non-sketched approach would be used to finish the job and guarantee good performance.

3. Future Work and Conclusions

While sketching is no longer a new idea, it has yet to be fully understood for a wide range of statistical methods. In particular, there is very limited work on sketched regularized regression, and even less for sparsity inducing regularizers or for distributed settings [7, 5]. These methods are extremely popular in modern statistical analysis and machine learning, and more research in this area is necessary. We believe that there is important work left to do in understanding the performance of distributed sketches for a wider range of methods—including what we have just mentioned—and that this direction is a natural next step given what has been done so far.

We discussed a distributed sketching procedure applied to stochastic gradient descent. This work looked at applications to training a machine learning model; beyond machine learning, it would be interesting to consider sketching in other domains. For example, can we *verifiably* apply sketching to physical simulations or to the solution of large physics-based systems? Many of these settings involve calculating and passing gradients across nodes, and a positive answer to this question would lead to much greater *scalability* of physical simulations and would greatly accelerate the scientific process.

We have discussed the application of sketching in distributed computation setups. Sketching enables better *scalability* of simulations and computations while offering some *privacy* benefits. Moreover, while it is a *randomized* algorithm, it offers *verifiable* performance guarantees. While the state of the art is impressive, it is still limited: too few varieties of algorithms and problems have been studied in this framework. We believe that more research in this area is both necessary and doable, and that new developments would greatly speed progress and push the frontiers of modern science.

References

[1] Burak Bartan and Mert Pilanci. "Distributed sketching methods for privacy preserving regression". In: *arXiv preprint arXiv:2002.06538* (2020).

[2] LingFei Dai et al. "A Distributed SGD Algorithm with Global Sketching for Deep Learning Training Acceleration". In: *arXiv preprint arXiv:2108.06004* (2021).

[3] Zengfeng Huang et al. "Communication-Efficient Distributed Covariance Sketch, with Application to Distributed PCA". In: *Journal of Machine Learning Research* 22.80 (2021), pp. 1–38.

[4] Zaoxing Liu et al. "Enhancing the privacy of federated learning with sketching". In: *arXiv preprint arXiv:1911.01812* (2019).

[5] Vu Pham and Laurent El Ghaoui. "Robust sketching for multiple square-root LASSO problems". In: *Artificial Intelligence and Statistics*. PMLR. 2015, pp. 753–761.

[6] Paul Voigt and Axel Von dem Bussche. "The EU general data protection regulation (GDPR)". In: *A Practical Guide, 1st Ed., Cham: Springer International Publishing* 10 (2017), p. 3152676.

[7] Shusen Wang, Alex Gittens, and Michael W Mahoney. "Sketched ridge regression: Optimization perspective, statistical perspective, and model averaging". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 3608–3616.

Sandia National Laboratories, Livermore, CA
Email address: aprasad@sandia.gov

Communication-free Secure Multi-Party Computation for Deep Neural Network Training using HPC

Yihui Ren[1], Tal G. Malkin[2], Shinjae Yoo[1]
1: Computational Science Initiative, Brookhaven National Laboratory
2: Department of Computer Science, Columbia University
{yren, sjyoo}@bnl.gov and tal@cs.columbia.edu

Thrust: Algorithms for secure, scalable, privacy-enhancing technologies

Motivating Use Cases

Many scientific and national security applications are exploring deep learning AI technologies. However, as a data-driven method, deep learning models are known for its data-hunger nature. Researchers have shown that by simply feeding more and more data, performance of the same model can be improved [6]. This is known as "unreasonable effectiveness of data". However, real data are not easily available, many of which are under strict regulations. Examples include but not limited to medical records, genome data, IAEA (International Atomic Energy Agency) surveillance data and so on. On top of private data availability, IAEA surveillance system requires inspection around the globe nuclear facilities, which implies that there are many sites have 1) limited computational capabilities (i.e. GPU), 2) limited storage capabilities and 3) limited network bandwidth. Such scenarios are not uncommon among other applications. For instance, many medical clinical sites tend to have good storage capacity but not computational capabilities, so expensive model training and even inference have to be outsourced. Also the network connection to many of such medical sites is reliable and bandwidth is usually limited, leading to communication intensive training mechanisms to be inefficient. Therefore, our fundamental research question is **"How can different data holders jointly train a model without revealing the data and without maintaining a live communication?"**

Challenges

There are many existing research directions such as secure multiparty computation, secure federated learning, differential privacy, and so on. Each has their own advantages and disadvantages. For example, due to added noise, the performance of differential privacy models tends to degrade. Secure federated learning requires activate communications and decent amount of computation resources at each site. By comparison, homomorphic encryption (HE) requires no communication except sending encrypted data at the beginning. It is considered "quantum safe". However, the computation overhead is also the largest. HE [5] is a form of encryption that permits computations on its ciphertext without decryption. The resulting ciphertext, if decrypted, will match the result of the same computation performed on the original plaintext. Prior to 2009, only a certain amount of computation could be applied due to the limited noise budget associated with HE schemes. Gentry [3] proposed a new procedure, homomorphic recryption (a.k.a. *bootstrapping*), to refresh the noise budget without decryption using lattice-based cryptography, so an unlimited amount of computation can be applied on ciphertext. This created the Fully Homomorphic Encryption (FHE) era. However, due to the high computation cost of bootstrapping, FHE had been deemed impractical.

Opportunities

Since Gentry's original FHE work, various FHE schemes have been proposed such as BGV, BFV, FHEW and TFHE. The most advanced and relevant to AI is the so-called CKKS scheme, named after its inventors, which supports real number arithmetic, and has been used for AI model inference. In a model inference as a service setting, the model could be trained in plaintext and held by a central party, a data provider can encrypt their sensitive data and send the ciphertext along with evaluation keys, to the central party. The central party applies the model to the encrypted data using FHE and send the output, which remains in ciphertext, back to the data provider. As AI model inference requires limited number of operations, it avoids the expensive bootstrapping operation.

In order to make joint model training possible, one needs to tackle the seemingly impracticality of bootstrapping. One way is to device more efficient bootstrapping algorithms. Comparing to the original CKKS bootstrapping algorithm and implementation [2], newer algorithms have dramatically reduced the

computation costs [4, 1] upto 56 times. Another way is to modify the bootstrapping algorithm to fully utilize leadership high performance computing facilities.

Future Research Directions

Automated AI model optimization and computation circuit design for FHE

CKKS scheme supports a limited set of arithmetic operations such as addition and multiplication. Other operations, including division, exponentiation and comparison, have to be approximated using various algorithms. Most of algorithms involve trade-offs between the number of computation levels needs to be consumed, the precision of approximation and the range of the approximation. Even better, we could design neural network models optimized for FHE without using complex functions, similar to the research direction on normalization-free networks. Another intricacy of FHE is the leveled computation circuit design: assign a fixed amount of noise budget for each computation level, and within the level, perform as much calculation as possible under the noise budget. How to achieve automated circuit design and schedule bootstrapping for a given network model is an important research direction.

Towards HPC-accelerated large-scale FHE computation

Most of FHE software libraries such as Palisade, HEAAN, SEAL, Lattigo and HElib, are research oriented and do not aim for an HPC environment or a distributed computing environment. Many of them do not have support for bootstrapping operation due to aforementioned alleged impracticality for single machine to compute. However, such seemingly expensive computation task involved in bootstrap can be distributed onto multiple nodes, and further be accelerated by GPUs. Since a neural network model contains multiple ciphertexts, there will be enough computation workloads for a large-scale computation. How to optimize the communication and computation is an interesting research direction. Another related direction is to boost the usability and accessibility of FHE neural network training. Since FHE calculations are highly vectorized, it might be possible to extend exsiting popular deep learning frameworks, such as PyTorch, Tensorflow and JAX, to support FHE operations.

References

[1] Jean-Philippe Bossuat et al. *Efficient Bootstrapping for Approximate Homomorphic Encryption with Non-Sparse Keys*. 1203. 2020. URL: https://eprint.iacr.org/2020/1203 (visited on 02/10/2021).

[2] Hao Chen, Ilaria Chillotti, and Yongsoo Song. "Improved Bootstrapping for Approximate Homomorphic Encryption". In: *Advances in Cryptology – EUROCRYPT 2019*. Ed. by Yuval Ishai and Vincent Rijmen. Vol. 11477. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 34–54. ISBN: 978-3-030-17655-6 978-3-030-17656-3. DOI: 10.1007/978-3-030-17656-3_2. URL: http://link.springer.com/10.1007/978-3-030-17656-3_2 (visited on 10/15/2020).

[3] Craig Gentry. *A fully homomorphic encryption scheme*. Stanford university, 2009.

[4] Kyoohyung Han and Dohyeong Ki. "Better Bootstrapping for Approximate Homomorphic Encryption". In: *Topics in Cryptology – CT-RSA 2020*. Ed. by Stanislaw Jarecki. Vol. 12006. Series Title: Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, pp. 364–390. ISBN: 978-3-030-40185-6 978-3-030-40186-3. DOI: 10.1007/978-3-030-40186-3_16. URL: http://link.springer.com/10.1007/978-3-030-40186-3_16 (visited on 10/15/2020).

[5] R. L. Rivest, A. Shamir, and L. Adleman. "A method for obtaining digital signatures and public-key cryptosystems". In: *Communications of the ACM* 21.2 (Feb. 1, 1978), pp. 120–126. ISSN: 0001-0782. DOI: 10.1145/359340.359342. URL: https://doi.org/10.1145/359340.359342 (visited on 07/09/2021).

[6] C. Sun et al. "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2017, pp. 843–852. DOI: 10.1109/ICCV.2017.97. URL: https://doi.ieeecomputersociety.org/10.1109/ICCV.2017.97.

Secure and Trustworthy Machine Learning for Life Sciences: Opportunities and Challenges

Bin Li, Chen Song, Parisa Ghasemi, and Xinghua Shi, Temple University.

With the advent of big data in life sciences, we envision a new era of scientific discovery propelled by rigorous development in Artificial Intelligence (AI) and machine learning (ML) for science. New methods are called upon to harmonize and integrate various types of scientific data across diverse computing infrastructure including high performance computing (HPC). As summarized in the "AI for Science Report" [1], the synergy between big data and powerful computing that is readily available at the DOE Office of Science domains, drives AI-empowered strategies for next generation scientific discovery and technological innovations.

In life sciences especially healthcare and biomedicine, the need of trustworthy computing to support AI-empowered analytics is prominent yet challenging. For example, a cancer patient may get her genome sequenced to help determine the subtypes of her tumor based on her genotypes or molecular markers to receive personalized therapy and care. An AI algorithm can be used to take in her genome, and calculate a risk score, classify her tumor into particular subtypes, determine treatment strategies, and predict clinical outcomes. To provide disease prediction aforementioned in precision medicine, sharable large-scale data resources that include a large number of collection of patients data from diverse populations, geographical and ethical groups would be desirable. However, such data are usually distributed and not sharable across different cancer institutions, healthcare systems, and computing infrastructures. With these constraints, it is challenging and sometimes infeasible to aggregate and share a large amount of data that is sufficient and necessary to develop robust, secure and trustworthy AI systems for biomedical research and clinical practices.

Looking forward, we anticipate significant breakthroughs in the development of algorithms, computing, and systems to enable secure and trustworthy computational data science. In short, we anticipate numerous opportunities and challenges in doing so from the following three perspectives.

I. Sharing and Provenance of Data and AI Models:

Data and models are core components in AI-empowered science, which are critical to be shared to promote broader adoption of AI models and knowledge transferring across science domains. New and concerted mechanisms are essential to enable the "found, accessed, and are interoperable and reusable (FAIR)" sharing of not only data but also the trained models. Currently, when models and/or data are shared, they are mostly shared decoupled and it is hard to track the linkage between data and models, not to mention the details about data usage and model optimization. Therefore, it is challenging yet desirable to provide a streamlined system where data (including input data, meta-data, intermediate data if useful) and model (including training procedure, hyperparameter tuning, meta-data about model details) are monitored and shared as a unified entity.

There are several strategies to enable wide-sharing of data and models at different levels, scales, and granularities. First, provenance is a powerful tool to facilitate the tracking and sharing of scientific data. Nonetheless, the provenance of AI models is still an emerging area where rigorous research is needed. Second, containers such as DockerSingularity, provide a convenient way for researchers to package and run AI models with pre-configured dependencies and libraries that allow for other scientists to reuse, share, and replicate those models. However, it is yet an underexploited field to build secure container-based provenance for scientific analytics centered around AI/ML on HPC. Third, adversarial simulation and enhancement of scientific data arises as an effective artificial instrument to generate a huge amount of synthetic data that mimic or emulate real data. For example, simulated data produced by generative adversarial networks (GANs) and their variants, can significantly enhance the quantity and quality of scientific data in situations where data is hard to gather or measure, contain data with very low signal-to-noise ratio, and too sensitive or private to share.

II. Transparent and Explainable AI:

Explainable AI (XAI)[2] and interpretable ML[3] have become active research fields where the goal is to provide new AI/ML methods that generate interpretable results and provide reasoning while maintaining high performance (e.g. prediction accuracy). Although explainable DL models have been investigated, it is still challenging to develop interpretable DL methods and AI systems for robust and scalable predictive modeling of scientific data. Generally, AI researchers consider interpretability to be the capability of a model in describing its decision-making mechanism. In an ideal scenario, the decision making process of an AI system is expected to be transparent and explainable so that decisions made by AI/ML models can be accountable and interpretable. In addition to being explainable and interpretable, responsible AI/ML systems are in great need so that their

output is rational, fair, generalizable, and extracts useful knowledge not based on accidents, biases, artifacts or noises.

III. Secure, Trustworthy and Federated Learning:

Although modern AI for science calls for widely sharing and aggregating of a large ensemble of scientific data, privacy and security are critical concerns for sharing such data and associated AI/ML models. Various studies have shown that not only data, but also AI/ML particularly cutting-edge DL models, can inevitably leak sensitive information and are thus subject to adversarial attacks. Additionally, provenance information itself can disclose data privacy and introduce security concerns to not only data and AI/ML models, but also the computing systems and environments. Therefore, it is of growing importance to provide rigorous privacy preservation and security guarantee when scientific data and AI/ML models are shared.

There are various algorithms and techniques to provide secure and trustworthy AI[4] for science, and we witness that these areas are quickly converging toward hybrid solutions that combine multiple algorithms or strategies. Here, we emphasize on three complementary strategies that stem from formal methods and programming language, data privacy and cryptography, networks and systems, and recently federated learning and distributed ML. First, formal methods should be utilized to develop verification-based systems to ensure trustworthy AI/ML. Scalable and robust verification systems that have shown great success in other areas (e.g. programming languages and systems) can be built into an AI/ML model so that the model is easily verified to be accessible, reliable, safe, secure, and privacy preserving. Second, various secure and privacy preserving algorithms (e.g. differential privacy, homomorphic encryption and/or secure multi-party computing) can be used standalone or in combination to guarantee the security and privacy of scientific data and AI/ML models. Third, Federated Learning (FL)[5] and secure distributed ML have emerged as effective infrastructures to support secure and trustworthy ML. FL is a distributed system to utilize privacy-related data deployed at multiple computing parties to train ML models. In federal training process, parameters could exchange across each party with differential privacy to establish a virtual shared model with reinforced data privacy. FL can be deployed among different types of devices (e.g. computers, smartphones, routers, and embedded chips) and computing infrastructures (e.g. HPC, cloud computing, edge computing, Internet of Things). Algorithms for deploying computer networks and ad hoc networks, and methods such as compression or encryption, can be utilized in FL to reduce the extremely high overhead when communicating model parameters among parties. In distributed ML settings, new approaches such as swarm learning[6] can be employed to support decentralized ML training on distributed datasets and computing nodes. Additionally, other technologies such as blockchain and software guard extension (SGX) provide alternative computing infrastructure for deploying secure AI/ML-based analytics for science.

In summary, secure and trustworthy AI for science calls for innovative development of algorithms and approaches with knowledge extracted from a wide range of computer and information sciences including formal methods, computer systems, networks, security, privacy, AI, HPC and ethic computing. We anticipate that research in these areas will catalyze the wide adoption of exascale AI/ML methods on DOE computing environments in diverse domains of relevance to the mission of the Office of Science and the Advanced Scientific Computing Research (ASCR) Program. Such investigations will provide unprecedented opportunities for embedding AI/ML into scientific discovery that offers remarkable potentials to accelerate the development of secure and resilient AI systems. These advances will eventually revolutionize research and applications in bioenergy, ecosystem, smart grid, high energy physics, climatology, cosmology, healthcare and disease treatment.

References

[1] R. Stevens, V. Taylor, J. Nichols, A. B. Maccabe, K. Yelick, and D. Brown, "Ai for science - report on the department of energy (doe) town halls on artificial intelligence (ai) for science," 2020.

[2] D. Gunning, "Explainable artificial intelligence (xai)," *Defense Advanced Research Projects Agency (DARPA), nd Web*, vol. 2, no. 2, 2017.

[3] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Communications of the ACM*, vol. 63, no. 1, pp. 68–77, 2019.

[4] J. M. Wing, "Trustworthy ai," *Communications of the ACM*, vol. 64, no. 10, p. 64–71, 2021.

[5] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint*, p. arXiv:1610.05492, 2016.

[6] S. Warnat-Herresthal, H. Schultze, K. L. Shastry, S. Manamohan, S. Mukherjee, V. Garg, R. Sarveswara, K. Händler, P. Pickkers,

N. A. Aziz, *et al.*, "Swarm learning for decentralized and confidential clinical machine learning," *Nature*, vol. 594, no. 7862, pp. 265–270, 2021.

Challenges Towards Efficient Anomaly Detection for Improving HPC Data Integrity

Seung Woo Son
University of Massachusetts Lowell
seungwoo_son@uml.edu

ABSTRACT

We discuss the need for unsupervised streaming anomaly detection mechanisms in the current and future HPC workflow for improving data integrity. We describe challenges in realizing an efficient anomaly detection for HPC datasets and then discuss the potential research directions to tackle these challenges.

1 KEY CHALLENGES

The massive volumes of datasets generated by modern high performance computing (HPC) applications of significant importance can help scientific discoveries reach better and faster decisions if scientists leverage the datasets correctly. However, the potential compromise in the datasets produced by HPC applications due to uncertain inputs, incomplete models, incorrect implementations, silent hardware errors, silent data corruption, malicious tampering, etc., can adversely affect the integrity of scientific interpretation. These anomalies in exascale HPC applications can no longer be ignored as they become more frequent for various reasons [3]. One solution to overcome such unwanted situations is to use anomaly detection techniques. An efficient and effective anomaly detection empowers scientists to take timely actions to correct anomaly situations. However, the unpredictability caused by anomalies, combined with the growing data size and complexity of HPC systems, makes effective use of anomaly detection challenging.

Anomalies, or outliers, are data artifacts that do not align with *normal* metrics in a dataset. As the normal data points, in most cases, are based on some long-standing assumptions made by comparing the data points to the normal group, identifying such classification and grouping for HPC datasets is a challenging task. One apparent reason is that the volume and velocity of the data generated at an unprecedented rate within many HPC systems make it difficult for typical algorithms to scale and retain their real-time characteristics [1]. Furthermore, it is practically infeasible to inspect every data point for learning anomaly models and detecting anomalies. Lastly, computing nodes are inherently in unfavorable circumstances for performing additional computations like anomaly detection. Nonetheless, effective anomaly detection on the HPC data stream is critical.

Anomaly detection has traditionally relied on the experience and expertise of human operators. Therefore, domain experts need to define the anomaly rule suited by application characteristics and deploy the defined rule-based anomaly detection algorithms to the target HPC systems, which is a significant burden to both systems and users. There is also much discussion on the types of algorithms applied to anomaly detection, such as shifting from supervised algorithms to unsupervised learning ones. In supervised machine learning (ML) based anomaly detection models, obtaining large-scale datasets with proper labels is challenging because the annotation process by domain experts like scientists is laborious and error-prone [4]. Thus it is not easy and often impossible to obtain a large labeled data sets required for typical supervised approaches in HPC.

2 FUTURE RESEARCH DIRECTIONS

To address the challenges in improving HPC data integrity, we propose that unsupervised real-time anomaly detection in a streaming manner should be a starting point. In the following, we elaborate on potential future research directions for achieving this goal.

2.1 Anomaly Detection in Streaming Data

There is much similarity between how data moves in HPC systems and streaming time-series data models.

Data points in large volumes of datasets produced by HPC applications typically follow smooth spatial patterns, like the collective trend or seasonal or cyclic patterns in the time-series data. This statistical property of data can represent the original data in a sparse signal using a proper transform. We argue that future research towards efficient anomaly detection should take advantage of streaming anomaly detection [2]. Treating HPC data as streaming data can open up new capabilities to current and future HPC workflows in improving data integrity. For instance, we can exploit this property to characterize profiles of HPC datasets and use those to estimate anomaly boundaries regarding datasets to classify anomalies accurately. The training step can utilize sparse sampling rather than using entire data points, which is more efficient and effective. Acquired raw stream of HPC data can be easily partitioned into blocks for exposing spatial correlation better, which would help improve the model's performance.

To build a robust time-series anomaly detection on HPC datasets, we need to consider at least three aspects of correlation: correlation across data points, time, and series. Many HPC datasets exhibit the first correlation due to smooth data patterns among neighboring data points, i.e., spatial correlation. Since many HPC datasets are generated or collected in a time-stepped manner through simulation or experiment, the anomaly detection model needs to exploit such temporal correlation. However, unlike spatial correlation, exploring temporal correlation requires more computation, memory, and storage. Lastly, since there are many variables (i.e., series) in a typical HPC application, we need to derive correlation among variables to build more robust anomaly detection models. Again, data in HPC systems and workflows behave similar to time-series data, so we need to exploit well-explored algorithms and mechanisms to realize robust data anomaly detection in HPC workflow, thereby improving data integrity.

2.2 Efficient Unsupervised Anomaly Detection

Unsupervised algorithms detect anomalies solely based on intrinsic properties of the unlabeled data points [1]. The unsupervised learning algorithms should be able to learn features useful to detect anomalies without relying on labeled data. However, the growing data scale of HPC systems makes this task even more difficult. Wang et al. [5] mentioned the statistics-based techniques such as curve fitting or autoregressive integrated moving average (ARIMA) model rely on the fact that an anomaly point value manifests itself as a large gap from the value of neighbors. Hence, these techniques ignore small errors that are unlikely to lead to an incorrect interpretation [5]. There has been significant progress in anomaly detection techniques based on recent advances in machine learning and deep learning, but they are not tailored to HPC workflows: they incur significant resource usage in training to achieve high detection performance. The new research is required to demonstrate the robustness of unsupervised anomaly detection algorithms, superior training speed, and high anomaly detection performance with minimal overhead to existing HPC workflows.

For a valid comparison with the state-of-the-art unsupervised anomaly detection techniques, there needs a systematic mechanism to inject anomalies into HPC datasets. There has been extensive research on anomaly generators on time-series data, which we can utilize in the HPC research community. The injection module needs to consider several factors, such as anomaly type (point, collective, or contextual), contamination ratios, etc. In many data-intensive HPC applications, the data integrity varies as the use cases differ from application to application. Therefore, the machine learning model should be designed in close collaboration with domain scientists and workflow users.

REFERENCES

[1] A. Borghesi, A. Libri, L. Benini, and A. Bartolini. Online Anomaly Detection in HPC Systems . In *IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2019.

[2] A. Moon, X. Zhuo, J. Zhang, S. W. Son, and Y. J. Song. Anomaly Detection in Edge Nodes using Sparsity Profile. In *IEEE International Conference on Big Data (Big Data)*, pages 1236–1245, 2020.

[3] S. Peisert, G. Cybenko, S. Jajodia, D. L. Brown, C. L. DeMarco, P. Hovland, S. Leyffer, C. Matarazzo, S. Prowell, B. Tierney, and V. Welch. ASCR Cybersecurity for Scientific Computing Integrity, 2015. Department of Energy Office of Science.

[4] H. Ren, B. Xu, C. Y. Yujing Wang, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, and Q. Zhang. Time-Series Anomaly Detection Service at Microsoft . In *KDD*, 2019.

[5] C. Wang, N. Dryden, F. Cappello, and M. Snir. Neural Network Based Silent Error Detector. In *IEEE International Conference on Cluster Computing (CLUSTER)*, pages 168–178, 2018.

Practical Privacy-Preserving Algorithms for Efficient Knowledge Dissemination at DOE Facilities

Topic: Algorithms for secure, scalable, privacy-enhancing technologies and frameworks
Authors: Christopher Stanley, Srikanth Yoginath, Mathieu Doucet, Debsindhu Bhowmik

Challenges

Improvement or enhancement in the understanding of a domain problem either by an artificial intelligence (AI) system or a scientist depends exclusively on the data results from continuous experiments. When such experiments are performed in independent competing silos, significant time and resources are lost due to redundant and uncommunicated efforts. Hence, at the core of accelerating scientific innovations lie the challenge of exchanging ideas and information, while continuing to protect and foster one's intellectual property to maintain a technical edge. Collaboration has always been at the core of scientific endeavors. As the scientific community grows, and as experimental techniques become more specialized, it becomes even more important to share data across institutions. Key scientific discoveries are often made using several techniques, often performed at different laboratories. For this purpose, lowering the threshold for data and knowledge sharing is an important element for accelerating science. Such needs have been identified for some time. For example, the large quantity of data produced in high-energy physics led to projects to distribute the storing and sharing of data [1]. Closer to us, the Globus project [2] was started more than two decades ago to allow data transfer between DOE institutions. More recently, as AI becomes more prominent as a tool to support scientific discovery, the sharing of data and knowledge generated from machine learning (ML) has been identified as a key component of our vision for how science will be done in the decades to come [3]. We envisage scientists to be able to share training data across laboratories and share trained models. Even before contemplating the design of a data sharing platform, the question of privacy comes to mind. In a highly competitive field, there is little incentive for scientists to share their data before having completely exhausted their usefulness. This data is often difficult to acquire and the description of the data itself, like the materials that were used in a particular study, can be perceived as a window on the scientific team's thinking and therefore something to be kept secret until publication. The benefits of a shared data platform for ML therefore need to outweigh those privacy risks.

Opportunity

In a U.S. Department of Energy (DOE) Laboratory setup, the user facility instruments collect large and ever-increasing datasets. While the historical data comprises that of many individual user experiments, the potential value in performing analytics across the aggregate data is immense, both for the facility and the users. For example, ML model training, which benefits from such large datasets to make proper generalizations, could be used to assist in facility decision making [4] and even to drive new experiments. However, an immediate concern is to ensure the security of each individual user data when considering any applications beyond the original intended scientific measurements. This is primarily because of the time-sensitive nature of the data, which still may be unpublished in the open literature and constitutes a liability on the facility. In addition, the data may be inherently sensitive or proprietary, depending on the sample, scientific details, and user affiliation (e.g., industry). Since trained ML models are capable to learn, and then unintentionally leak, private information about the training data, it is essential to include appropriate privacy and security measures for such applications. These applications may range from assisting the user facility in optimizing the performance of instruments, deciding future instrument upgrade plans and science directions, to directly informing back to guide the individual measurements of users. The overall goal is to leverage AI to advance facility operations and science while still preserving data security of individual researchers.

Currently, the industry is using privacy-preserving technologies like, differential privacy, federated learning, secure multi-party communication, homomorphic encryption and so on, to secure ML models and data. These innovations are significant and could be efficiently leveraged and enhanced to meet the DOE facility's precision, scale and data volume requirements. We believe these challenges can be addressed by

consistently evaluating industry developed privacy-preserving algorithms for DOE use cases and by deriving or creating new scalable algorithms to efficiently exploit advanced computing architectures to meet the DOE facility specific precision, scale and volume challenges. In this regard, we have studied the performance behaviors of privacy-preserving algorithms like differential privacy and secure MPC on high-performance computing (HPC) and private Cloud systems [4]. While we demonstrated the feasibility of knowledge dissemination while preserving privacy, we noted severe computational performance constraints that could make direct adoption of industry developed privacy enablers impractical. Furthermore, there exists different methods to ensure privacy, for example: Clipping and randomizing stochastic gradient descent updates and private aggregation of teacher ensembles (PATE) are two different methods to realize differential privacy-based ML model training. Since the domain is still nascent and active, ample room for further algorithmic innovations exist. Further, we consider that privacy concerns are at the center of scientific knowledge exchange and hence expect most of the data or models of the future to incorporate privacy-preserving algorithms. This will also aid in efficiently handling the automated experiment scheduling and operations for optimal exploration of a domain space, and automated tuning of scientific instruments for optimal performance. For example, privacy-preserving algorithms within an AI agent training environment would only be able to access the generic information without divulging into the intrinsic details of the experimental results. However, automation of control using methods such as reinforcement learning are sequential, slower and any expectation of privacy-preservation algorithms in this learning environment can be expected to make the process extremely slow. Hence, the applicability of privacy-preserving methods, the extent to which they can be incorporated in supervised, unsupervised and reinforcement learning methods, and their practical limitations in the DOE setup will guide the privacy-preserving algorithmic development research effort.

Timeliness

In recent times, the fields of ML, HPC, cloud computing, service-oriented architectures, continuous-integration and continuous-deployment methods and privacy-preserving algorithms have seen significant development and maturity. These innovations are very significant since they together provide all necessary technologies to realize the knowledge extraction and their dissemination from the instruments or simulations on large scale facilities at the DOE laboratories to the researchers across the world [4]. As the DOE labs aim to seamlessly integrate different facilities like scientific instruments, HPC systems, edge systems, cloud computing systems and so on, to realize integrated and automated workflows like ORNL's INTERSECT initiative [5], it becomes necessary to acknowledge the significance and centrality of privacy. Timely introduction of privacy-preserving algorithms within large-scale multi-facility and/or multi-lab workflows in their design phases, are extremely necessary. This can play a significant role in timely scientific knowledge dissemination, which essentially results in coordinated scientific exploration resulting in accelerated scientific innovations.

References

[1] Lamanna, M., "The LHC computing grid project at CERN, Nuclear Instruments and Methods in Physics Research" A 534 (2004) 1–6

[2] Foster, I., "Globus Online: Accelerating and Democratizing Science through Cloud-Based Services," *Internet Computing*, IEEE, May-June, 2011.

[3] Ratner, D., et al. "BES Roundtable on Producing and Managing Large Scientific Data with Artificial Intelligence and Machine Learning." United States.

[4] Yoginath, S.B., Doucet, M., Bhowmik, D., Heise, D., Alamudun, F., Yoon, H., Stanley, C. "Secure Collaborative Environment for Seamlessly Sharing of Scientific Knowledge," SMC 2021.

[5] ORNL's INTERSECT Initiative: https://ldrd.ornl.gov/drd/intersect

Leveraging Fault Tolerance for Secure Scientific Computing

Keita Teranishi (knteran@sandia.gov)*, Hemanth Kolla, and Jackson R. Mayo
Sandia National Laboratories, Livermore, CA

Thrust areas: Algorithms, Platforms

1 Motivation Prevalent use of edge and cloud computing will create new methodologies for computational science, tailoring real-time collection of empirical data, data analytics, and large-scale simulations together to deliver new scientific insight through federated learning. This architecture is enabled by numerous small edge computing clients, cloud, and on-premises HPC resources coordinating data exchange and computation. However, this complex and highly distributed computing future not only entails a widely understood need for resilience to accidental faults, but also poses a major concern for cybersecurity to ensure the integrity, availability, and confidentiality of the data and results. Tightening the security policy could negatively impact performance, making it difficult to deliver results within a given time window. On the other hand, lack of security could damage the level of trust in scientific data, even if attacks do not succeed in altering the final outcome of an analysis.

To enhance security for the new applications and analysis at ASCR, we propose leveraging fault tolerance techniques already being developed for HPC systems and their applications. Large-scale computing systems have been vulnerable to variety of *inherent hardware and system faults* due to their size and complexity. Fault tolerance is intended for mitigating faults and failures of computing systems and applications, keeping desirable system and application behaviors amid loss or undesired alteration of computing resources and data. We see many commonalities between cybersecurity and fault tolerance in HPC. Since the exact patterns of faults and cyberattacks are not known beforehand, methods for both need to be proactive – anticipating various kinds of faults and attacks and designing for them – rather than reactive. With hardware and software stacks becoming deep, heterogeneous, and diverse, the likelihood of interacting and compounding faults and attacks makes the mitigation strategies even more complex. In such situations, fault tolerance and cybersecurity at or close to the application level could be the final layer of protection. For the past two decades, the fault tolerance community in HPC has made significant progress in adapting a variety of techniques for system configurations and applications in HPC. Similar approaches can be effective for the new cybersecurity needs of future HPC systems and applications.

Fault tolerance and security are different paradigms and there are limitations in applying solutions for one to the other. Cybersecurity aims to maintain desired behavior in the presence of attackers, focusing on three facets: integrity (data remains correct), availability (data remains accessible and usable to authorized personnel), and confidentiality (data remains private from unauthorized personnel). Fault tolerance primarily aims to equip computations to handle random localized errors in the hardware and/or software stack. Fault tolerance techniques can help address integrity and availability, but are less applicable to confidentiality. The major challenges are (1) technical feasibility of transforming fault tolerance techniques to cybersecurity, (2) performance portability from edge to supercomputers, and (3) adaptivity to a variety of numerical properties and formats of scientific data.

*Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

2 Proposed Approaches

Approaches inspired by fault tolerance can enhance cybersecurity in three main ways.

Employing fault tolerance techniques and features Fault tolerance can be utilized by leveraging the R&D on the techniques, and/or by leveraging specific features in a computing system. A host of algorithm-based fault tolerance (ABFT) techniques [3] have been developed that achieve scalable and efficient fault detection and recovery using domain-specific properties. Examples include physics-based checksums that detect when state has been corrupted, and communication-minimizing algorithms that ensure convergence in the face of missed or corrupted messages. These techniques apply for some scenarios of deliberately corrupted application state. Such techniques have also been complemented by a variety of system-specific features: error correction code (ECC) for memory subsystems, storage approaches for data redundancy (e.g., checkpointing), triple modular redundancy, resilient asynchronous many-task runtimes, dynamic resource allocation, batch and thread scheduling. Together, these techniques and features can be used to manage corruption and damage due to cyberattacks – helping with detection, or at least supporting a recovery response from the surviving resources if attacks are detected in other ways. ABFT techniques can be applied to the recent domain-specific or adaptive data service and file system middleware such as ADIOS [2] and Seacas [4] for checking application-specific data integrity.

Embracing nondeterminism From a cybersecurity perspective, determinism is an attacker's friend as it allows the attacker to study the system, understand patterns, and cover their tracks to evade identification. Nondeterminism via diversity and randomness, on the other hand, makes it hard to devise a stealthy, targeted attack [1]. Heterogeneity in HPC can enhance nondeterminism and support detection and recovery from cyberattacks, since a vulnerability in one chip, protocol, or algorithm may not be shared with another. Randomization of resource scheduling may confuse attackers, though it has ramifications for the reproducibility of computation. Fault tolerance requires handling nondeterminism due to random faults. Accordingly, an application being driven by fault tolerance can be beneficial for security as well.

Applying containment and partitioning A key challenge in the ready use of fault tolerance techniques is that they are designed for random, relatively infrequent, and typically uncorrelated hardware faults, whereas a well-designed cyberattack could be designed to eventually distribute across the whole system. However, fault tolerance techniques can still be useful if they are complemented by containment and partitioning in the system, such as "sandboxing" [1]. Partitioning the system allows localizing the effect of an attack and limiting its spread, so that the remainder of the system remains reliable and can perform recovery.

References

[1] R. C. Armstrong, J. R. Mayo, et al. Complexity science challenges in cybersecurity. SAND2009-2007, Mar. 2009. wiki.cac.washington.edu/download/attachments/7478403/Complexity+Science+Challenges+in+Cybersecurity.pdf.

[2] W. F. Godoy, N. Podhorszki, et al. Adios 2: The adaptable input output system. a framework for high-performance data management. *SoftwareX*, 12:100561, 2020. ISSN 2352-7110.

[3] J.-Y. Jou and J. Abraham. Fault-tolerant matrix arithmetic and signal processing on highly concurrent computing structures. *Proceedings of the IEEE*, 74(5):732–741, 1986.

[4] G. D. Sjaardema. Seacas. https://gsjaardema.github.io/seacas-docs/sphinx/html/index.html, 2021. [Online; accessed 12-October-2021].

Differential Privacy is not *Privacy*

Vandy Tombs (tombsvj@ornl.gov)
Oak Ridge National Laboratory, Oak Ridge, TN, USA

Topics: Algorithms, Data

Challenge: Privacy. The word privacy can mean vastly different things to different people. For example, some individuals might say their privacy is violated if a single picture of them is taken without consent while others might only say their privacy was violated if images taken of them at a certain location are released; others still might say that privacy requires that they can control any image taken of them and another that might say the image just can't be traced back to them. As Daniel Solove has put it, "privacy suffers from an embarrassment of meanings" and it is "far too vague a concept."

Despite this widespread disagreement in the English language, the technical world has embraced one definition of privacy: differential privacy. This mathematical definition of privacy, introduced in 2006, formalizes privacy using a parameter ε (and later δ as well) to measure how inclusion/exclusion of an individual in a database effects the output of an algorithm using that database. Or in other words, how likely an adversary is to determine if a specific instance of a record was included in the database just from observing the algorithm's output. There are several variations of differential privacy, and all are quite similar in what they measure. This definition has been well-explored and is seeing increasing utilization across various scientific domains. For example, the US Census has announced that it will be using differential privacy when releasing statistics for the 2020 Census.

While differential privacy has several strengths it is only a single definition and cannot hope to represent what every individual or government means when they discuss or require keeping data private. Using the incorrect privacy definition can lead to misunderstanding and could potentially provide incorrect privacy guarantees or worse, lead to privacy vulnerabilities and data leakage. Of course, not every English definition of privacy can or should have a corresponding formal mathematical definition; however, there are cases where a different mathematical definition of privacy should be used. For example, differential privacy may provide stronger privacy guarantees than is needed for a specific use case and thus introducing the noise needed to achieve differential privacy would needlessly decrease the accuracy of the algorithm. There are also cases where differential privacy is insufficient or would require too much noise to achieve. In both these cases, another privacy definition should be used.

There are several other technical definitions of privacy and even frameworks for developing unique privacy definitions. However, these other mathematical definitions are underutilized and under explored. Thus, the methods and mechanism for achieving these other privacy definitions may be immature or not exist at all. Developing these mechanisms cannot and should not fall on the shoulders of practitioners or users of a specific definition of privacy but needs to be done by research teams and facilities who have researchers in several different scientific domains as well as the computer science and mathematical expertise to develop these mechanisms for individual domains or use cases. DOE is a unique position to develop and increase access to these other definitions of privacy. In summary, there are two main limitations that prevent algorithm developers from choosing the correct privacy definition:

1) lack of awareness and/or understanding of what these definitions guarantee, and
2) the mechanisms for achieving these definitions are not well developed or may not exist at all.

Opportunity:

Challenge 1: One approach for making practitioners aware of more privacy definitions is to develop more privacy definitions using a mathematical framework like Pufferfish [1], which can be used to create

unique privacy definitions. These definitions are probabilistic in nature, much like differential privacy (in fact, differential privacy is one definition that can be produced from this framework). However, creating meaningful definitions from this framework requires expert knowledge about how data within a domain changes over time, as well as an understanding of what properties of the data should be considered private. Frequently, individuals with that kind of expertise lack the experience in mathematics to be able to utilize a framework like Pufferfish. Domain experts and privacy experts need to work together to develop one or several privacy definitions that should be used within a given field. These definitions should cover a wide range of use-cases within the field and should create a tight bound on what is considered private.

There are also syntactic privacy notions that have formal definitions and there has been work done to formulate games that an adversary must win to break the privacy notion (e.g see [2]). Understanding the goal of an adversary helps practitioners understand different privacy notions and what notions are the most important for their domain. This can help guide the privacy definitions that should be used or produced for a given field.

Challenge 2: Once an agreed upon set of privacy definitions have been selected for a given domain or use-case, the methods for achieving these privacy definitions need to be developed. This may look like the noise mechanisms developed for differential privacy (e.g., the Gaussian mechanism). These mechanisms could be introduced into various algorithms much like how current mechanisms are used. This is the best approach for algorithms that are deterministic in nature. However, many algorithms that should achieve privacy are probabilistic in nature. Thus, if the privacy definition is probabilistic the algorithm may already have some privacy attributes or achieve some privacy. Understanding the inherent privacy attributes of different classes of algorithms under different privacy definitions would decrease the amount of trade-off between privacy and algorithm performance. Mathematical proofs detailing these attributes would be needed and would decrease the amount of noise that needs to be introduced to achieve a certain privacy level within a specific privacy definition.

Timeliness: Algorithms are under increasing scrutiny to be private both by governments and by individuals. At the same time, our reliance on algorithms is also increasing. To satisfy these privacy requirements, algorithm developers will likely continue to just use differential privacy due to where it is in its current stage of development. Now is the time to bring other privacy definitions to the same stage. Until these mechanism and methods for achieving a wide range of privacy definitions are developed, algorithm developers and practitioners will likely continue to use differential privacy without any thought about whether it is the correct definition for their case. We need to stop equating privacy with differential privacy in order to develop and facilitate the use of the correct privacy definition based on the specific field or use-case.

References:

[1] Kifer, Daniel, and Ashwin Machanavajjhala. “Pufferfish: A Framework for Mathematical Privacy Definitions.” ACM Transactions on Database Systems 39, no. 1 (January 2014): 1–36. https://doi.org/10.1145/2514689.

[2] Ankele, Robin, and Andrew Simpson. “Privacy Games for Syntactic Privacy Notions,” 2017. http://eprint.iacr.org/2017/1126.

Machine Learning for Identifying Vulnerable Libraries in Binary Code

POC: Tristan Vanderbruggen, LLNL, vanderbrugge1@llnl.gov
Co-Authors: Steve Chapin, Nathan Pinnow
Topics: Machine Learning for Graph – Vulnerability Detection in Platform

Introduction

Commercial-of-the-shelf software is omnipresent in our cyber-infrastructure. From the firmware of grid devices to mundane inventory software, most software used on our networks is delivered in a binary form that is opaque to inspection. Unfortunately, it means it potentially includes defective components, sometime making it vulnerable to cyber-attacks. Worse, in many cases, we know that these components are vulnerable but are unable to detect them. Finally, this does not require malicious actors: it only requires an oversight at one stage of a complex supply chain. In this paper, we describe a research topic with two main benefits: developing the state-of-the-art in machine learning applied to graphs, and securing DOE's infrastructure with a particular emphasis on edge computing for large-scale experiments. These same techniques can be applied to analyze firmware on control devices in areas of high concern for DOE, including energy infrastructure such as power grids and natural gas pipelines.

Use Cases

DOE facilities, supercomputing and others, use a significant amount of software. In many cases, this software is packaged in manners that are opaque to inspection. We discuss two potentials issues below: one could affect many networked hardware systems (especially edge devices collecting data for large scale experiments), while the other could affect any software distributed in binary form. Both cases could create serious security vulnerability in DOE's infrastructure.

In April 2021, the U.S. Cybersecurity and Infrastructure Security Agency (CISA) issued an advisory warning of multiple vulnerabilities in the OpENer EtherNet/IP stack that could expose industrial systems to denial-of-service (DoS) attacks, data leaks, and remote code execution. This library is used by manufacturers of "I/O adapter devices" meaning all sorts of physical devices that need to be controlled through a network. Now, how likely is it for components manufactured before this advisory (hence using a vulnerable version of this stack) to be remove from a supply chain? In this case, we hope it would—given the high visibility of this incident (with five major flaws) b—but many incidents do not get as much publicity.

Another, more mundane, example is the Expat XML parser. This library is used by a large number of applications, libraries, and hardware to ingest (very) large XML documents. A search of Common Vulnerabilities and Exposures (CVE) related to "libexpat" returns six results with the four most recent ones (2017-2019) scoring a high severity (> 7.5). With its permissive licensing, its vulnerable source-code could be directly integrated into software packages without any mention of it. It is also a very portable C code meaning that compiler optimization and toolchain can easily be tweaked. Once integrated in the binary distribution of another software package, conventional techniques would be hard pressed to detect it.

Software in all areas of computing is increasingly distributed as binary packages through package management systems which free developers from maintaining their software's dependencies. In these environments, local security measures check individual packages for blacklisted software.

This approach still suffers from a flaw: the software package might silently include vulnerable components (such as an older version of libexpat) that are invisible to the local checks.

Challenges

As a matter of fact, software provided in binary form—without source code, proper identification of dependencies (libraries and versions), or any information about the toolchain used to produce the binary—makes the detection of known security weaknesses difficult. Traditionally, security experts rely on hashing techniques to match software components (or pieces of these components) within databases of known vulnerable software. However, these techniques are extremely sensitive to the noise introduced by variables as benign as the choice of compiler, or even the optimization level used when creating the binary. These issues are compounded by the large number of toolchains, hardware targets, and even the large optimization space offered by modern compilers. For many years, security researchers and engineers have been looking for techniques to identify known vulnerable pieces of binaries. Some of the most advanced techniques rely on de-assembly, followed by control flow analysis (including generation of a Semantic Control-Flow Graph (SCFG)), and semantic annotations. Problems arise when trying to match the extracted SCFG for a specimen under examination with those contained in a database of known components. Techniques to compute similarity between these graphs fall in the category of subgraph isomorphism which are NP-Complete. As it is customary to build SCFG at the function granularity, computing matches between a target binary's functions and a database of functions from vulnerable libraries becomes prohibitively expensive.

Research Opportunities

Recent advances in machine-learning provide a path to build resilient and fast matching algorithms for these semantic control-flow graphs. We recommend immediate and significant investment in the study of self-supervised geometric deep-learning as a technique to build embeddings of semantic control-flow graphs. This domain, identification of vulnerabilities in binaries, has both a high intrinsic value and the ability to provide valuable datasets for theoretical and practical studies in machine learning. In particular, the study of geometric deep-learning, which started with graph neural networks, is emerging as a general framework to unify the bestiary of deep-learning architectures and techniques. However, graph neural networks are mostly used for either protein folding (many small graphs) or social networks (a single huge graph). Comparatively, the analysis of binaries has the possibility to provide a variety of graph structures and sizes. Finally, these datasets can, in most cases, provide clean training targets, using known vulnerable code, permitting us to compare conventional supervised learning with self-supervised techniques. Self-supervised learning considers the data itself as a learning target and was instrumental in advancing other areas, including natural language processing, to their current levels. Techniques to apply self-supervised learning on graphs are appearing and will be instrumental to build resilient graphs matching techniques. Finally, DOE is uniquely positioned, relative to private industry, in having both the responsibility to protect the nation's energy infrastructure that is put at risk by binary software of unknown provenance and composition, and the supercomputing facilities to address the computational cost of exploring the tuning space associated with establishing new deep-learning architecture for emergent use cases.

Blockchain and the Scientific Method

James A. Evans,[1] Kweku Opoku-Agyemang,[2] Krishna Ratakonda,[3] Kush R. Varshney,[3] Lav R. Varshney[4,5]
[1]University of Chicago, [2]University of California, Berkeley, [3]IBM, [4]University of Illinois Urbana-Champaign,
[5]Brookhaven National Laboratory (lvarshney@bnl.gov)

I. Introduction

As more subtle, large-scale, or complex phenomena are being studied, scientific knowledge claims are becoming less trusted and alarms of a 'crisis in science' have been raised. The basic experimental hypothesis-testing process of scientific research that crystallized five centuries ago, however, has largely remained intertwined with trust embodied in institutions such as learned societies, universities, journals, and funding agencies. Such institutional trust enabled the scaling of the scientific enterprise beyond small social networks of participants with long-term relationships, but may be insufficient when the file drawer problem, p-hacking, hypothesizing after the results are known (HARKing), insufficient computational replicability, corrupted reagents, overstressed peer-review systems, misaligned incentives, biases in favor of particular interests, opaque misconduct investigations, and other such factors have eroded trust. In addition to eroded trust, science is also facing a challenge of reduced productivity. For example, the number of disruptive papers and novel concepts per unit cost have been slowing considerably. Known as Eroom's Law (the reverse of Moore's Law), the cost of drug discovery is doubling every nine years. Nearly all of these challenges are coming to the forefront during the scientific community's response to the COVID-19 pandemic; scientists and institutions are often bypassing their prevailing practices and culture in favor of expedience. These actions are being taken informally without attendant changes in the supporting infrastructure and tools.

Novel cognitive and communication tools have a general-purpose ability to accelerate scientific progress. Chalk and slate changed scientists' cognitive processes; modern statistical and computational tools enable discovery within large datasets; and artificial intelligence technologies are poised to suggest novel hypotheses. The printing press changed scientific communication by standardizing format in journals and facilitated citing previous publications; photocopying enabled a shift from editorial review to peer review; and preprint servers allowed timestamps to establish priority and credit, while also making knowledge public without delay. New blockchain technologies—originally developed to underpin cryptocurrencies but having recently found use in domains as diverse as global trade, political elections, food safety, and property rights management—offer an alternative mode of technology-mediated distributed trust. As such, blockchain has been touted as a panacea to numerous of science's ills. Several prototypes have been deployed for clinical trials, journals, etc. Yet, many argue that a simple database is all that is needed when blockchain is deployed. Here we argue that there are several problems in scientific discovery for which blockchain is appropriate.

At a high level, we propose the following paradigm for scientific research. A funding entity solicits research on a general topic of inquiry or specific goal and earmarks a purse that is to be awarded to contributors in advance to pursue certain goals or afterwards once those goals are met. Contributions in all parts of the scientific method are recognized, not only the final discovery. For example, background research, hypothesis generation, experimental design and preparation, data collection, data analysis, exposition of results, and simplification of conclusions for policymakers and the general public are all disaggregated individual steps of the scientific method that may receive awards for an eventual discovery. All contributions are treated as transactions and posted to a permissioned blockchain in appropriate formats without haste, perhaps in the form of non-fungible tokens (NFTs). They are immediately available for public inspection and expert review. Smart contracts trigger automatic solicitation of peer review as well as meta-analysis and assimilation of new contributions into the body of knowledge. Reproduction and replication are also incentivized in award schemes. In the case of post hoc prizes and bounties, the final rewards are also triggered by smart contracts once a goal has been met, reviewed, and sufficiently replicated. Findings can then be background research for a new cycle of the scientific method.

Technology is never a complete solution in the absence of cultural change, but here we argue that blockchain technologies have the potential to act as joint cognitive and communication (collective intelligence) tools to accelerate trusted science. Indeed, blockchain permits a reimagining of the scientific process in the digitally connected age, where anyone anywhere may contribute to a collectively evolving body of knowledge and be rewarded for it: an alternative scientific culture and practice of representation whose virtues could be accentuated in distributed discovery across numerous branches of science. We provide some further detail in the sequel.

II. Specific Challenges

It is clear that blockchain technologies may mitigate reproducibility and replicability challenges in science, but here we focus on other challenges.

A. Disaggregation of Contribution

Scientific papers are the key currency in the economics of science. Yet, there is a lot of scientific activity around building digital tools including data and code that is not credited by such a system. Given the growing importance of such tool building in science, however, it is important to ensure mechanisms of credit for workers engaged in these activities. It is clear that if an immutable ledger of smart contracts is used to keep track of analysis protocols, then all contributions will be recorded. This is not surprising since blockchain itself is an offshoot of cryptocurrency systems like bitcoin, which must keep records of all contributions as transactions. Engineering effort may be required to make things work in the context of human behavior, which itself will suggest new theoretical research questions, e.g. in the game theory of mechanism design. Moreover, although there are debates as to whether publications should be tours de force or least publishable units, scientific publications are certainly 'chunky' and only reveal the hypothesis and data *if* the findings 'work out.' This means that many hypotheses, datasets, and findings are omitted from the literature. A potential blockchain remedy is to 'publish' and link to hypotheses, data, methods, findings, and relevance, while yielding more equitable incentive structures.

B. Open, Fair, and Rapid Dissemination

Although many journals are named *Transactions*, current scientific publishing in scholarly conferences and journals is said to be slow and closed. One might wonder if blockchain-based mediums for dissemination may replace journals, building on current preprint mandate trends that have been motivated in part by mechanisms for establishing priority by researchers and in part as remedies for paywalls for readers. A promising innovation that is already influencing the scientific method is using blockchain to add a layer of rigor to peer-review by making the process more transparent than is currently possible. Such a solution improves on the status quo in terms of promoting reviewer recognition while lessening ethical compromises. Such innovations can help journals and outlets protect the integrity of their internal processes and potentially limit the concern of being biased in favor of better-resourced researchers, or researchers with whom they share professional relationships. Although there are protocols to prevent such outcomes, many fields are sufficiently small that overlap is difficult to entirely avoid in practice. We argue that such issues are inherently challenging for many disciplines and blockchain may help professions to be more open to novelty. Any datastore based on blockchain must be trusted by the entire academic community to scale, which is why we advocate for systems that are fully owned by the communities they hope to serve. Blockchain-based review and dissemination may also increase trust in fast-moving crisis situations such as the COVID-19 pandemic, and may facilitate dissemination to policymakers and the general public.

C. Post Hoc Filtering and Analysis

Information overload is a central concern in scientific knowledge, and this may cause an especially strong lock-in of paradigms in large and growing fields. Moreover, just because the data stored in a blockchain is immutable does not imply that scientific knowledge is immutable. In fact, scientific knowledge is mutable and continually evolves as new investigations are carried out. The same empirical results in the blockchain may be interpreted by different scientists differently, since they may have differing priors. Here we argue that crowdsourced and semi-automated annotation of meta-data with findings could allow the Bayesian update of facts from throughout the scientific literature. In fact, one might automate meta-analyses from raw data directly, or using AI techniques to read papers and perform meta-analyses. Such constant assessment and incorporation of new results can accelerate science.

D. Funding and What to Work On?

Scientific research, whether motivated by public-interest or curiosity-driven considerations, often requires funding. Many considerations enter into decisions about what types of scientific studies to fund, including striking a balance between exploratory and confirmatory research. Both safe and bold research help move science forward, but there is bias against novelty in science funding. Often these decisions informally reflect the mission, goals, and values of the funding agency. Inducement prizes allow strong expression of values, but they are inefficient as all-pay auctions. Grants have an information asymmetry / principal-agent problem. Contracting approaches are usually used for principal-agent problems, but better monitoring alleviates it also. Blockchains may provide an approach to monitoring that gives stronger ability for funding agencies to express these values.

Besides the mission, goals, and values of the funding agency, the current state of knowledge within a field of study is also important in funding decisions. Moreover, exploratory research is more susceptible to non-replication, whereas confirmatory research is less likely to uncover exciting new discoveries. Fields investigating potentially ground-breaking avenues will produce results that are less replicable, on average, than fields that investigate highly likely, almost-established results. Indeed, a field could achieve near-perfect replicability if it limited its investigations to prosaic phenomena that were already well known. As such, the *uncertainty* could be one blockchain-enabled value that could be operationalized using prior probability. Tension can arise between replicability and discovery, specifically between the replicability and the novelty of the results. Hypotheses with low a priori probabilities are less likely to be replicated.

Further, smart contracts issued by funding agencies could enable programmatic articulation of their goals. Smart contracts could also support funded markets for ideas, conjectures, and other contributions with programmatic expression of goals.

A Flexible Framework for Privacy-Preserving Scientific Computing

Cory Hauck (ORNL, Email: hauckc@ornl.gov), Stanley Osher (UCLA, Email: sjo@math.ucla.edu),
Thomas Strohmer (UC Davis, Email: strohmer@math.ucdavis.edu),
Bao Wang (Univ. of Utah, Email: bwang@math.utah.edu)

A primary driver behind modern data-driven scientific computing is the availability of useful data. However, numerous reasons make access to data and data sharing cumbersome: because the data contain sensitive information of some kind or are considered a proprietary trade secret, or due to legislation related to privacy protection. These barriers to data sharing radically reduce the value of big data and are increasingly impacting computational science. Therefore, it is imperative to find mechanisms to make sensitive data available for advancing scientific discovery while adhering to the imposed stipulations such as privacy preservation.

Yet, scientific data come in all kinds of forms—the data may be unstructured, heterogeneous, and multimodal. This makes it difficult or sometimes even impossible to use off-the-shelf techniques, such as open-source differential privacy libraries, to achieve the required data protection necessary for sharing.

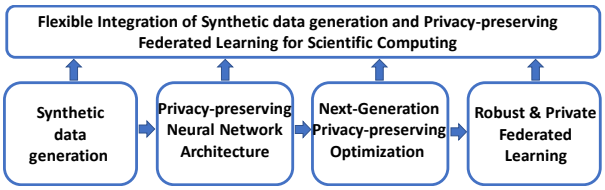


Figure 1: Workflow of our flexible framework for privacy-preserving scientific computing.

Hence, what is urgently needed to overcome these limitations is the creation of a principled and flexible framework for privacy-preserving scientific computing, shown in Fig. 1. Developing tools for such a framework requires a community effort to ensure an effective feedback loop between theory, algorithms, and applications. We highlight several topics that such a framework needs to encompass and point out some promising research directions.

Differential privacy and beyond. Differential privacy (DP) is a statistical technique which guarantees that the outcome of any data analysis is essentially equally likely independent of whether any individual record has been included or not included in the dataset [1]. DP thus ensures a certain type of privacy. DP is usually implemented by injecting noise to enforce a "privacy budget". However, applying DP is currently challenging, requiring a high degree of expertise and effort [2]. Moreover, the injection of noise to the data or the learning procedure reduces utility and introduces bias.

Thus, there is a pressing need for new algorithms and analytical tools that enable more precise privacy accounting and higher utility in DP. For instance, recent work has shown that it is possible to achieve DP without adding noise (via a technique called "*private sampling*" [3]), but much more research needs to be done to explore such directions. For example, which other randomized algorithms can achieve DP without injecting noise into the data? Moreover, it is a priori not clear that the form of privacy provided by DP is suitable for all use cases of scientific computing. It would be extremely useful to develop additional forms of privacy protection tailored for various computational science scenarios (beyond existing and generally not strong enough anonymization techniques).

Synthetic data generation for scientific computing. *Synthetic data* is a promising concept to overcome various limitations of current privacy protection strategies, by reconciling data innovation with data privacy [4]. The goal of synthetic data is to create an as-realistic-as-possible dataset—one that not only maintains the nuances of the original data—but does so without risk of exposing sensitive information. The concept of synthetic data holds great potential as privacy-enhancing technology for scientific computing. Many of the deep learning based methods for making synthetic data come without any utility guarantees and thus are delicate to use in mission-critical scientific computing settings.

An important challenge in this context is thus the development of an algorithmic framework for accurate and privacy-preserving generation of heterogeneous, unstructured, dynamic synthetic datasets. Here, the notion of accuracy should be tied to the tasks of the subsequent data analysis.

Privacy-preserving neural architecture design. Neural networks play an increasingly prominent role in scientific computing. In this context, machine learning (ML) with DP guarantees is obtained through noisy training data, training loss perturbation, or noise-injected gradient. Under these mechanisms, even though DP is achieved, the accuracy of ML models drops severely. Thus, what is needed are methods that can significantly improve upon the current privacy-utility tradeoff. Some possible directions include: Developing network architectures that are less sensitive to accuracy loss caused by noise injection. Initial results demonstrate provably improved accuracy and adversarial robustness of the resulting ML models, indicating the potential of this direction [5]. In particular, neural ODE-style continuous-depth models, which are especially suitable for learning complex dynamics and for scientific discovery, seem an interesting concept to explore. Since graph data arise in various scientific computing applications, it seems critical to develop privacy-preserving *continuous-depth graph neural networks* for deep learning with graph data. In particular, it seems feasible to design privacy-preserving and expressive graph neural networks by parameterizing diffusion equations on graphs with neural ODEs [6] and leveraging tools from partial/stochastic differential equations. It is also interesting to explore if the neural network parameterized graph diffusion protects data privacy due to its diffusive nature.

Robust and private federated learning. Federated learning (FL) is an ML setting where a massive number of entities (clients) solve an ML problem collaboratively without transferring raw data, under the coordination of a central server. FL trains ML models by exchanging the model parameters between clients and the central server. Therefore, this method has undeniable appeal for scientific computing, since FL decouples the model training from the need for collecting or direct access to the private training data. However, FL has been shown to be vulnerable to model inversion attacks, external linkage attacks, and other privacy invading methods. FL has some other drawbacks: the communication cost between the server and clients can be excessive since a large number of clients are involved in a practical FL system; the failure of the server would disrupt the training process of all clients.

As such, developing a *decentralized federated learning* (DLF) system is crucial, which replaces client-server communication with peer-to-peer communications. Several critical problems to resolve: How to design a robust overlay network that guarantees fast convergence and reliable generalization of ML? How to design a client-client communication protocol that can automatically account for potential communication failures? How to improve local training algorithms to optimize privacy-utility tradeoff? To address these problems, we need a hardware and software co-design.

Another challenge lies in how to fully utilize the computational power of the exascale high performance computing centers that DOE built for privacy-preserving federated learning. Integrating synthetic data, privacy-preserving federated learning, and the use of hardware trusted execution environments [2] is another envisioned direction that holds great potential.

References

[1] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2014.

[2] S. Peisert, "Trustworthy scientific computing," *Commun. ACM*, vol. 64, no. 5, pp. 18–21, 2021.

[3] M. Boedihardjo, T. Strohmer, and R. Vershynin, "Private sampling: a noiseless approach for generating differentially private synthetic data," *arXiv preprint arXiv:2109.14839*, 2021.

[4] S. M. Bellovin, P. K. Dutta, and N. Reitinger, "Privacy and synthetic datasets," *Stan. Tech. L. Rev.*, vol. 22, p. 1, 2019.

[5] B. Wang, Z. Shi, and S. Osher, "ResNets ensemble via the Feynman-Kac formalism to improve natural and robust accuracies," in *NeurIPS*, pp. 1657–1667, 2019.

[6] B. Chamberlain, J. Rowbottom, M. I. Gorinova, M. Bronstein, S. Webb, and E. Rossi, "Grand: Graph neural diffusion," in *ICML*, pp. 1407–1418, 2021.

Integrated Workflow Management for Scientific Research on Complex Systems

Dexin Wang and Zhenyu (Henry) Huang

Background

Many problems in the energy system domain require investigation of complex systems, composed of many components which may interact with each other. For example, the power system has evolved into a stage that its resilience, reliability, and efficiency must consider emerging complex factors both within and without [1]. Such complex factors include:

1) Within the grid, more uncertainties and scenarios need to be analyzed in a timely manner for both operation and planning purposes.
2) Within the grid, the mutual impact between transmission and distribution can no longer be ignored.
3) External to the grid, the communication system which the grid command and control heavily rely on has significantly increasing its coupling with the grid. Such dependency must be fully analyzed and understood for optimizing grid operation and cyber security.
4) External to the grid, many other energy systems increase their impact on the grid, as an input function or as an output function or as an operating constraint.

Workflow Management

The energy systems research community needs tools to facilitate the management and collaboration along the typical workflow depicted in Figure 1, which is inspired by typical workflows in the data science community [2, 3].

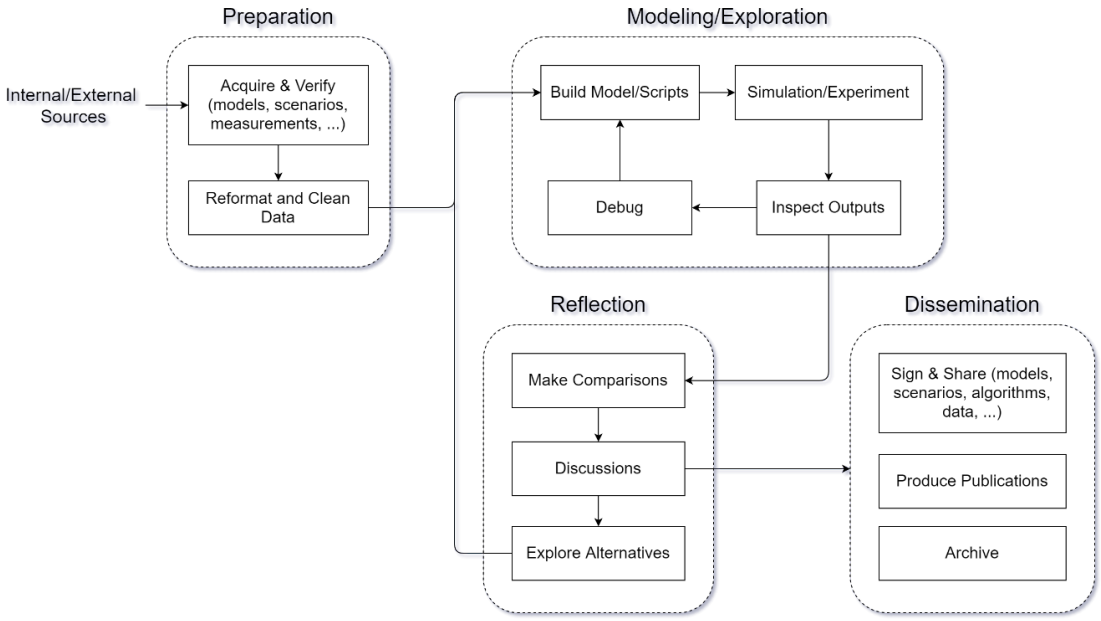


Figure 1 A Typical Research Workflow for Complex Systems

There are four major phases in the workflow:

Preparation

The acquisition of high-quality data, including system/component models, use cases, measurements, etc., is crucial for, and often the first step of, impactful research on energy systems. Verifiable provenance and integrity of the data is also indispensable for a vibrant research community that encourages open, trustworthy, and efficient exchange of ideas and results. In addition, the acquired data sometimes require reformat and/or cleaning to accommodate the specific needs of the research at hand.

Modeling/Exploration

When dealing with complex systems, the capability to model subsystems and their interactions in paramount to the success of research projects. In view of the increasing complexities in the power systems and their interdependencies with other systems, grid architecture ought to be, and has been as part of the DOE-sponsored efforts, reexamined to define the functionality and services across the grid layers and its surrounding systems. For example, GMLC supported a co-simulation platform HELICS that facilitates studies of the complexity in the emerging interactions in the power grid and other systems.

However, it has been an ongoing challenge to leverage these efforts and incorporate them to facilitate the development, evaluation, and demonstration of new designs, algorithms, technologies, and tools to cope with such emerging complexity and scales which are beyond today's industry capabilities. Such complexity also increases the need for high performance computing hardware and software, large-scale data analytics, tool reconfigurability, and fast iterations. The industry and industry tools do not have easy ways to accomplish these.

Reflection

Any projects of practical size and complexity involve reflecting on the results produced in the previous phase. This could be internal discussions within the research team or ones including larger groups of people. The workflow management system should be able to facilitate some reflections at an integral part of the scientific workflow.

Dissemination

The effective dissemination of models, algorithms, data, and discoveries is of paramount importance to form complete loops in the research community to inspire and enable new research projects. The products of one research project are often building blocks of another. Artifacts involved in the workflow should be signed by the author(s) to verifiably ensure their integrity and preserve their provenance.

Conclusions

To facilitate investigations on complex systems, the energy systems research community needs an integrated workflow management system that encompasses all phases of the typical scientific workflow, ensures verifiable provenance and integrity, and encourages open, trustworthy, citable exchanges of research artifacts.

References

[1] Jeffrey D. Taft, "Grid Architecture 2," United States, 2016, [Online] Available: https://www.osti.gov/biblio/1244801
[2] Jeff Saltz, "What is a Data Science Workflow?", Data Science Process Alliance, 2020, [Online] Available: https://www.datascience-pm.com/data-science-workflow/
[3] Philip Guo, "Data Science Workflow: Overview and Challenges," Communications of the ACM, 2013, [Online] Available: https://cacm.acm.org/blogs/blog-cacm/169199-data-science-workflow-overview-and-challenges

Data Integrity and Provenance in Self-Driving Facilities

John Wu

As more scientific research activities are being computerized, many of the decisions used to require human involvement have become automated. The recent DOE report on AI for Science envisioned Self-Driving Labs and Self-Driving User Facilities in the next decade. In this position paper, we propose that such self-driving facilities require extensive work on data integrity and provenance to ensure the proper functioning of such facilities and our confidence of the scientific results obtained from them.

Advancements in both hardware and software technologies have made it possible to scientific user facilities operated by DOE collected detailed information about fast-evolving physical phenomenon at unprecedented scale. Assisted by the emerging artificial intelligence technology, many of the user facilities are exploring dynamic experiment designs, for example, to control the heating of a sample under soft X-ray or vary the experiment parameters in response to the observed conditions. A number of DOE workshop reports anticipate that some of these user facilities could be fully self-driving in another decade. Such self-driving facilities would be able to complete experiments and observations much faster, conduct intricate examinations of experiment parameter space beyond human reaction time, and safely handle conditions that might be toxic to humans. However, in order for the results to be trustworthy, we have to be able to ensure the integrity of the data and be able to trace the history of data from raw data through all its transformations and derivations.

There are significant challenges to data integrity and provenance in a distributed environment involving large user facilities with its own data acquisition and processing tools, as well as the supporting tools, external data sources, edge computers, AI devices/accelerators, cloud computing resources, and high-performance computers. The first challenge is the variety of the devices involved. The second challenge is the new computing, memory, and storage technologies that might not follow the well-understood trust models, such as defined by the Trusted Computing Group. The third challenge in this environment is the dynamic nature of the computing elements involved. For example, it is likely, the cloud computing resources will expand and shrink based on the anticipated computing needs. At the same time, there might be IoT devices that could be brought into the system as needed. Data integrity in such a dynamic and distributed heterogeneous environment require new research.

The existing software framework for data integrity heavily rely on checksums and blockchain. In fully self-driving scenarios, heavy reliance on blockchain technology might not be able to support fast interactions, and work needs to be done to ensure the data integrity framework would not slow down the interactions unnecessarily. Similarly, the existing provenance standards are defined for execution on powerful CPUs, and might not be well-suited for the diversity of computing elements available at the edge. For example, some of the edge computing elements might be too small, while others (e.g., ML chips and quantum accelerators) might not follow Von Neumann Architecture.

Experimental measurements and observations generally have errors/uncertainties associated with them. Traditional data integrity concepts do not capture such uncertainties. We believe it is necessary to expand the scope of data integrity to include uncertainty and provide ways to propagate uncertainties through all forms of transformations, derivations and inferences.