Carnegie Mellon University
Software Engineering Institute

RESEARCH REVIEW 2022

FORGING THE PATH

A MESSAGE FROM THE CTO

In the past decade, a number of emerging disruptions have forced those of us concerned with software engineering to adopt new approaches for bringing engineering rigor to software and systems development. These disruptions are of particular concern to the Department of Defense (DoD), which now depends on software to deliver the majority of new capabilities necessary to maintain strategic advantage.

As the only federally funded research and development center (FFRDC) focused on software, we share this concern. This is why, in 2019, we adopted a new technical strategy: *Software Transforming the Mission.* In executing this strategy, we're forging a path along which software engineering can help create stability in a world of software disruption, and software architects can design whole, stable systems that present fewer opportunities for adversaries. In other words, our goal is to help the DoD to realize advantage through software.

To achieve this goal, we established four targeted, cross-cutting objectives:

- Automate the software development and DoD acquisition lifecycle.
- Create operational resilience for missions.
- Realize artificial intelligence (AI) and future computing.
- Integrate the preceding objectives into mission-capable systems.

At the 2022 Research Review, our researchers will detail how they are forging a new path for software engineering by executing this strategy and creating tangible results. They will highlight methods, prototypes, and tools aimed at the most important problems facing the DoD, industry, and academia, including AI engineering, computing at the tactical edge, threat hunting, continuous integration/continuous delivery, and machine learning trustworthiness. You will learn how our researchers' work in areas such as model-based systems engineering, DevSecOps, automated design conformance, software/cyber/AI integration, and AI network defense—to name a few—has produced value for the U.S. DoD and advanced the state of the practice.

Each session will present tools, techniques, and processes arising from our integrated technical objectives, which are designed to transform the mission through software and address four big software challenges for national defense and security: capability, timeliness, affordability, and trustworthiness. You will also hear about the positive impact of those technologies on the DoD's mission.

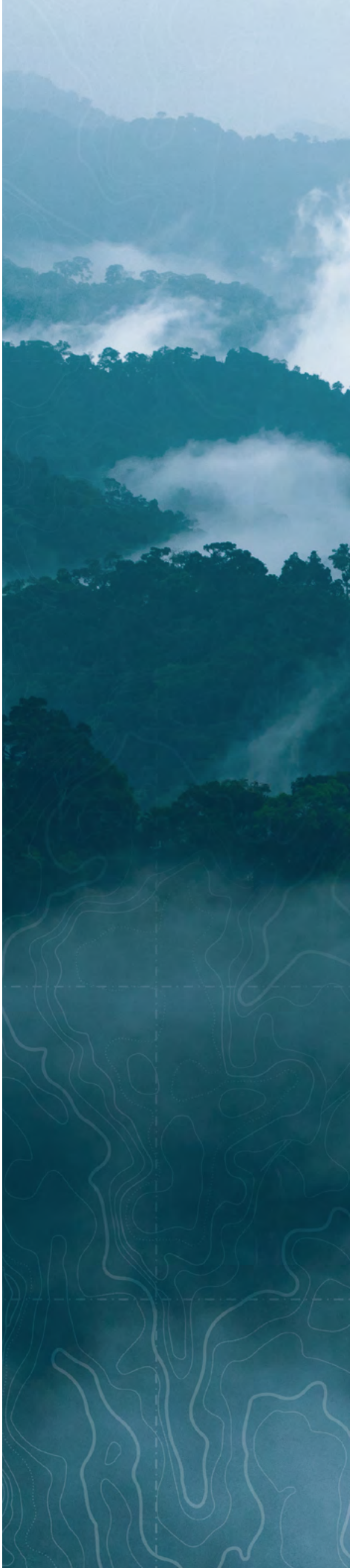
This book highlights selected projects from our portfolio of DoD-sponsored public research for fiscal 2022 presented at the 2022 CMU SEI Research Review. These projects include recently concluded work and work that continues in our research pipeline to study, make, and transition results to the benefit of DoD, the USG, academia, and the private sector.

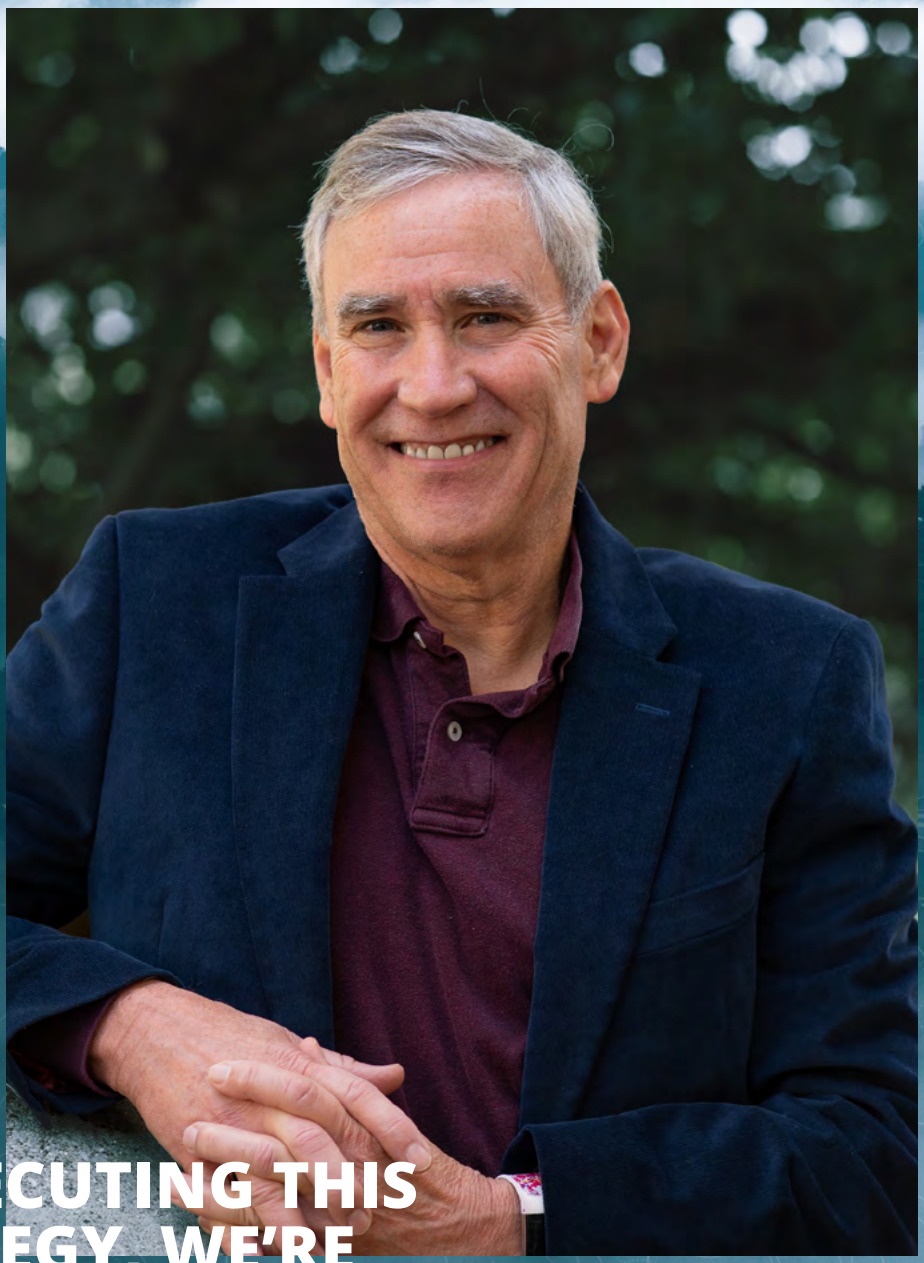
I hope you enjoy reading about CMU SEI's fiscal 2021 research efforts and that the following pages demonstrate the pride we take in this work. I invite you to read more about the SEI's technical strategy in my SEI Blog post, *The Drive Toward Stability,* which you can find at **https://insights.sei.cmu.edu/blog/the-drive-toward-stability/**. I also invite you to view videos of the 2022 Research Review, which you can find at the SEI YouTube channel, **https://www.youtube.com/c/TheSEICMU**.

We stand by to work with you to help you make a difference, and we encourage you to contact us at **info@sei.cmu.edu**.

TOM LONGSTAFF

Chief Technology Officer, Carnegie Mellon University Software Engineering Institute





“IN EXECUTING THIS STRATEGY, WE’RE FORGING A PATH ALONG WHICH SOFTWARE ENGINEERING CAN HELP CREATE STABILITY…”

CONTENTS


DAY 1 DA


Y 1 DAY 1

AI ENGINEERING IN AN UNCERTAIN WORLD

PRINCIPAL INVESTIGATOR

DR. ERIC HEIM
Senior Research Scientist—Machine Learning

1

2

3

4

5

SEI COLLABORATORS

1. JOHN KIRCHENBAUER
Machine Learning Engineer,
AI Division

2. JON HELLAND
Machine Learning Researcher,
AI Division

3. JACOB OAKS
Associate Developer,
AI Division

EXTERNAL COLLABORATORS

4. DR. AARTI SINGH
Associate Professor,
Machine Learning Department

5. DR. ZACHARY LIPTON
Assistant Professor,
Machine Learning Department

The DoD is increasingly seeking to deploy AI systems for mission-critical tasks. Modern AI systems most commonly employ machine learning (ML) models to make important, domain-relevant inferences. However, due in part to uncertainty, state-of-the-art ML models can produce inaccurate inferences in scenarios where humans would reasonably expect high accuracy. Furthermore, many commonly used models do not provide accurate estimates about when they are uncertain about their predictions. Consequently, AI system components downstream from an ML model, or humans using the model's output to complete a task, must reason with incorrect inferences that they expect to be correct. Motivated by this gap, this project aims to accomplish the following objectives:

- Develop new techniques, and utilize existing ones, to give ML models the ability to express *when* they are likely to be wrong without drastically increasing the computational burden, requiring significantly more training data, or sacrificing accuracy.
- Develop techniques to detect the cause of uncertainty, learning algorithms that allow ML models to be improved after the cause of uncertainty is determined, and methods for reasoning in the presence of uncertainty without explicit retraining.
- Incorporate uncertainty modeling and methods to increase certainty in the ML models of government organizations.

This SEI research team has made **real progress** on its goal of detecting model uncertainty and **mitigating its effects** on the quality of model inference.

Our work seeks to realize three overarching benefits. First, ML models in DoD AI systems will be made more transparent, resulting in safer, more reliable use of AI in mission-critical applications. Second, ML models will be more quickly and efficiently updated to adapt to dynamic changes in operational deployment environments. Third, we will make adoption of AI possible for missions where AI is currently deemed too unreliable or opaque to be used.

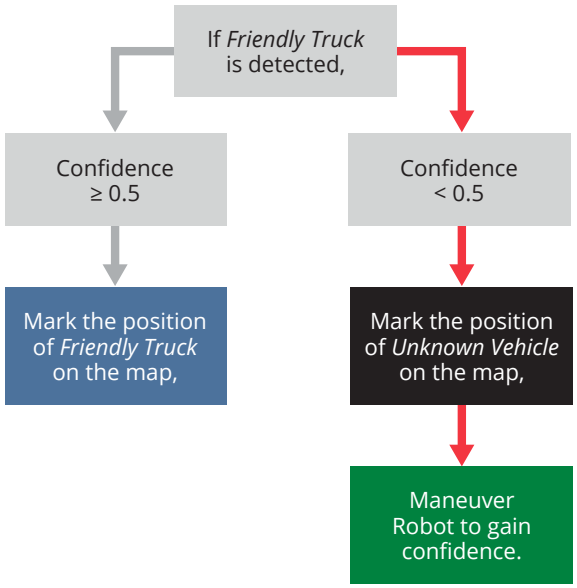
Our SEI team of Eric Heim, John Kirchenbauer, Jon Helland, and Jake Oaks brings expertise in the science and engineering of AI systems, human-computer interaction, enterprise-level infrastructures, and perspectives informed by a collective 50 years of experience leading and conducting projects for both government and industry. Our CMU collaborators Dr. Zachary Lipton and Dr. Aarti Singh bring expertise in monitoring and improving ML models in the presence of uncertainty.

Now in the second year of this three-year project, the team has worked together to make progress on their goal of detecting model uncertainty and mitigating its effects on the quality of model inference by using uncertainty to characterize types of errors and applying uncertainty quantification to object detection.

IN CONTEXT

This FY2021–2023 project

- builds on DoD line-funded research, including graph algorithms and future architectures; big learning benchmarks; automated code generation for future-compatible high-performance graph libraries; data validation for large-scale analytics; and events, relationships, and script learning for situational awareness
- aligns with the CMU SEI technical objective to be timely so that the cadence of acquisition, delivery, and fielding is responsive to and anticipatory of the operational tempo of DoD warfighters and that the DoD is able to field these new software-enabled systems and their upgrades faster than our adversaries
- aligns with the DoD software strategy to accelerate the delivery and adoption of AI



By allowing high-level reasoning to be informed by predictive uncertainty, AI systems can be **more robust** to failures caused by unconfident predictions.

Quantifying Uncertainty: A Key Component for Informative and Robust AI Systems

PORTABLE HIGH-PERFORMANCE INFERENCE ON THE TACTICAL EDGE (PHITE)

PRINCIPAL INVESTIGATOR

DR. SCOTT MCMILLAN
Member of the Technical Staff—
Principal Engineer

1

2

3

4

5

SEI COLLABORATORS

1. JAY PALAT
Senior Engineer

2. OREN WRIGHT
Member of the Technical Staff—
Senior Researcher

EXTERNAL COLLABORATORS

3. DR. TZE MENG LOW
Co-Principal Investigator, Assistant Research Professor, Electrical and Computer Engineering
Carnegie Mellon University

5. DR. NICHOLAI TUKANOV
PhD, Electrical and Computer Engineering
Carnegie Mellon University

4. DR. UPASANA SRIDHAR
PhD, Electrical and Computer Engineering
Carnegie Mellon University

STUDENTS

PANKTI RAJESH SHAH
Master's Student, Electrical and Computer Engineering
Carnegie Mellon University

NAVYA CHANDRA
Master's Student, Electrical and Computer Engineering; Independent Study: "Fused Convolution on Pi Pico"
Carnegie Mellon University

DoD applications at the tactical edge will increasingly involve the use of sensors and other devices across a range of edge-based capabilities. To support this capability, an emerging branch of ultra-low-power machine learning (ML) technology has arisen, in part, to meet the needs of this technology. To make it work, the DoD needs access to high-performing software support for the wide variety of hardware architectures encountered in the embedded space. However, the software ecosystems for low-power embedded devices remain rudimentary, highly varied or non-existent, and dependent on users to develop their own software stack.

In year two, we will **continue to optimize** the portable library of ML algorithms (targeting 10x–100x performance gains over baseline)...

To address these challenges, our project, Portable High-Performance Inference at the Tactical Edge (PHITE), a collaboration with experts from the Department of Electrical and Computer Engineering at Carnegie Mellon University, applies performance engineering processes to the analysis of existing open source ML frameworks for embedded systems, to inform the development and optimization of a portable software library that can achieve significantly higher performance (10–100x power efficiency) for a set of ML applications across a range of targeted embedded devices.

Our approach proceeds in three phases:

1. selection and characterization of baseline applications and hardware platforms
2. software design and development
3. integration, evaluation, benchmarking, and demonstration of open source benchmark applications that have DoD relevance

In the first year of the project, we analyzed application workloads and performance modelling of the hardware, specified the first version of the API specification for the portable inference interface, and completed the initial implementations of a library of ML algorithms for each ultra-low power hardware platform. In year two, we will continue to optimize the portable library of ML algorithms (targeting 10x—100x performance gains over the baseline) for embedded devices and address any additional needs to support a live data stream.

IN CONTEXT

This FY2022–23 project

- aligns with the SEI technical objectives to bring capabilities that make new missions possible or improve the likelihood of success of existing ones
- aligns with the SEI technical objective to be timely so that the cadence of acquisition, delivery, and fielding is responsive to and anticipatory of the operational tempo of DoD warfighters and that the DoD is able to field these new software-enabled systems and their upgrades faster than our adversaries
- aligns with the DoD software strategy to realize computational and algorithmic advantage

PHITE provides an open source library of ML algorithms optimized for low-power embedded devices to enable a wide range of applications at the tactical edge through portable and more capable software foundations. These applications can range from small drone-mounted sensors (left) to sensors that support the ML-enabled ATLAS targeting system being developed by DEVCOM's C5ISR and Armaments Centers.
Photos U.S. Army

AUTOMATING MISMATCH DETECTION AND TESTING IN ML SYSTEMS

PRINCIPAL INVESTIGATOR

DR. GRACE LEWIS
Principal Researcher and TAS Initiative Lead

1

2

3

4

SEI COLLABORATORS

1. DR. RACHEL BROWER-SINNING
Machine Learning Research Scientist

2. ALEX DERR
Associate Software Engineer

EXTERNAL COLLABORATORS

3. DR. CHRISTIAN KÄSTNER
Associate Professor
Carnegie Mellon University
School of Computer Science

4. CHENYANG YANG
PhD Student
Carnegie Mellon University
School of Computer Science

As the DoD adopts machine learning (ML) to solve mission-critical problems, it is faced with an inability to detect and avoid inconsistences among assumptions and decisions made by data science/ML engineering, software engineering, and operations stakeholders. Examples include (1) poor model accuracy because model training data is different from production data, (2) system failure due to inadequate testing because developers were not able to produce appropriate test cases or lacked access to test data, and (3) monitoring tools are not set up to detect diminishing model-related problems such as diminishing accuracy. We therefore define *ML mismatch* as a problem that occurs in the development, deployment, and operation of an ML-enabled system due to *incorrect assumptions* made about system elements by different stakeholders that results in a negative consequence. This ML mismatch can lead to delays, rework, and failure in the development, deployment, and evolution of ML-enabled systems.

According to NIST, it is **15 times more expensive** to fix a bug detected during systems testing and **30 times more expensive** if detected during production [NIST 2022].

It is common knowledge in software engineering that the later a fault is detected, the more expensive it is to fix—this also applies to late discovery of ML mismatches. As an example, in ML-enabled systems, the differences between training data and production data are often not discovered until field testing or in production. According to NIST, it is 15 times more expensive to fix a bug detected during systems testing and 30 times more expensive if detected during production [NIST 2022]. Without automated and extensible tools for ML mismatch detection, incorrect assumptions made about system elements by different stakeholders are discovered too late in the development of ML-enabled systems.

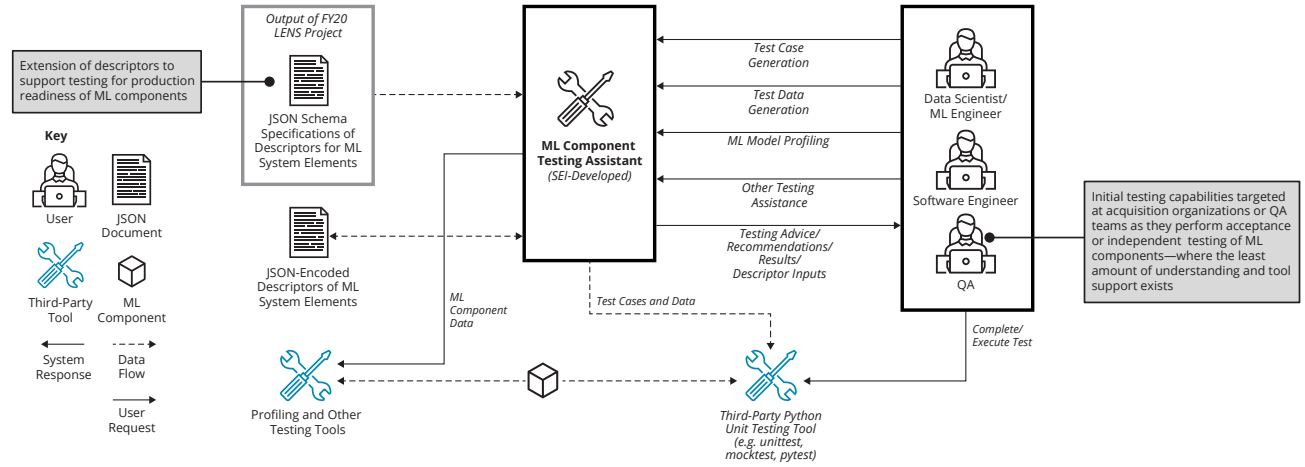
This project builds on a set of SEI-developed descriptors for elements of ML-enabled systems [Lewis 2021]. We are developing a suite of tools to (1) automate ML mismatch detection and (2) demonstrate how to extend descriptors to support testing of ML-enabled systems. The tools will also support descriptor validation on open source and DoD ML systems and components. For testing, we are explicitly focusing on production readiness of ML components that we define based on several attributes: ease of integration, testability, monitorability, maintainability, and quality, which we define as meeting both model requirements and system requirements.

This project's end goal is for DoD organizations to adopt descriptors and tools for early mismatch detection and production-readiness test and evaluation as part of their ML-enabled systems development process. To this end, this project contributes to and advances the SEI's modernizing software development and acquisition objective, both by improving formalization of detection of ML mismatch, improving testing practices for ML-enabled systems, and providing tool support that in the long run can be integrated into ML-enabled system-development toolchains.

IN CONTEXT

This FY2021–23 project

- builds on the results of our *FY20 Characterizing and Detecting Mismatch in ML Systems* project that empirically defined and validated the information that explicitly needs to be shared between stakeholders to avoid ML mismatch
- aligns with the CMU SEI technical objective to reduce the cost of acquisition and operations, despite increased capability, and makes future costs more predictable by reducing delays, rework, and failure
- aligns with the CMU SEI technical objective to be trustworthy in construction and implementation and resilient in the face of operational uncertainties
- aligns with the DoD software strategy to accelerate the delivery and adoption of AI



This SEI suite of tools aims to (1) automate ML mismatch detection and (2) demonstrate how to extend descriptors to support testing of ML-enabled systems.

A MACHINE LEARNING PIPELINE FOR DEEPFAKE DETECTION

PRINCIPAL INVESTIGATOR

DR. SHANNON GALLAGHER
Machine Learning Research Scientist

1

2

3

SEI COLLABORATORS

1. DOMINIC ROSS
Multi-Media Design and Communications Lead

2. JEFFREY MELLON
Machine Learning Research Scientist

3. DR. CATHERINE BERNACIAK
Senior Machine Learning Research Scientist

The use of AI to create image or video forgeries (deepfakes) represents a pervasive threat to the DoD. Indeed, both the National Defense Authorization Act (NDAA) of 2021 and the Identifying Outputs of Generative Adversarial Networks (IOGAN) Act of 2020 address the risk posed by deepfakes to national security. However, the barriers to creating photorealistic deepfakes are low: They can be created with a single, affordable gaming-type graphics processing unit with easily accessible and frequently maintained open source software.

The aim of this project is to develop a **deepfake detection** prototype framework… with at least **85% accuracy**…

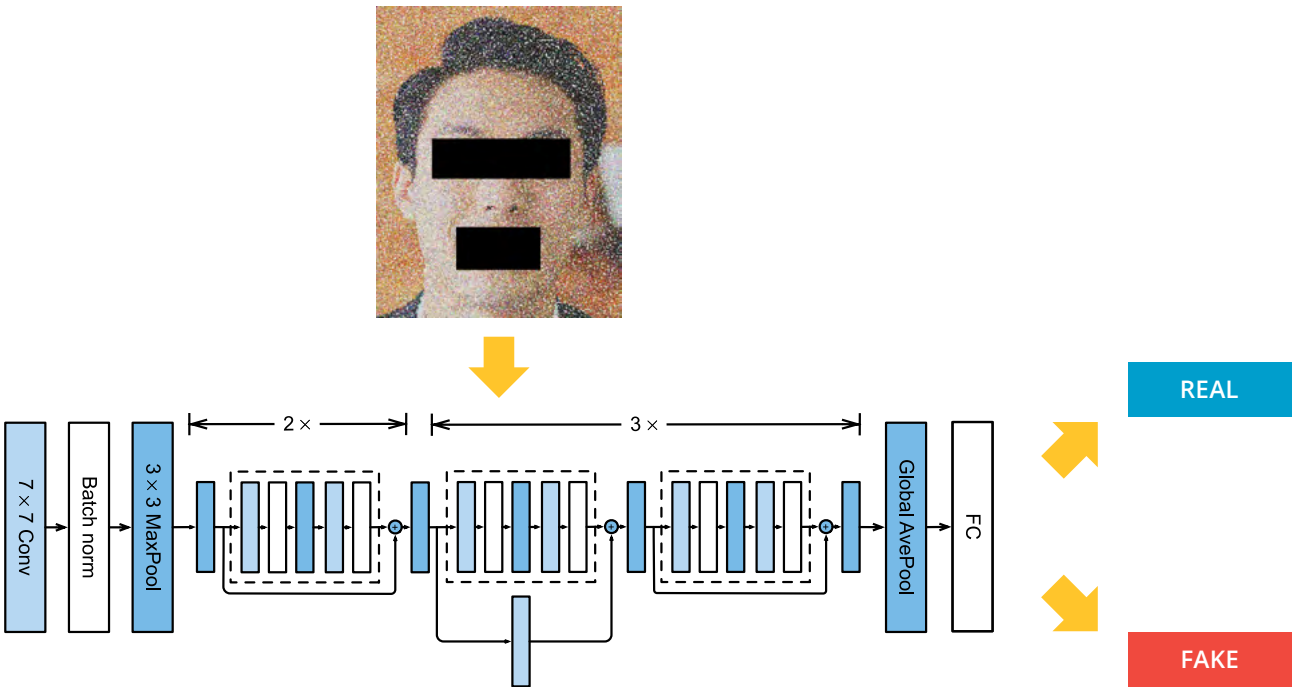
Fortunately, the neural network architectures used to create deepfakes can also be engineered to detect them. AI detection algorithms have been developed that can identify manipulation not readily visible to the human eye, but no detector tool exists that can ingest multiple modes of media (image, video) and process them at scales required in DoD or other medium-to-large scale environments. The aim of this project is to develop a deepfake detection prototype framework that incorporates open source and novel detector algorithms capable of detecting at least three types of AI artifacts per mode with at least 85% accuracy, for both image and video modes.

The initial phase of our work consisted of identifying and testing state-of-the-art, open source deepfake detector algorithms and selecting those that perform the best on open source data. We plan to incorporate these algorithms into a prototype tool that can ingest each mode of data (image and video) at scale, perform filtering and triaging relevant to a specific type of media (e.g., images of human faces, text, videos containing human subjects), and assign a probability of that media having been manipulated with AI. We will work to determine the best way to combine individual model scores into a robust, high-integrity performance metric. The goal of these efforts is a working prototype capable of ingesting all modes of data and able to detect at least three types of AI artifacts for each mode with at least 85% accuracy.

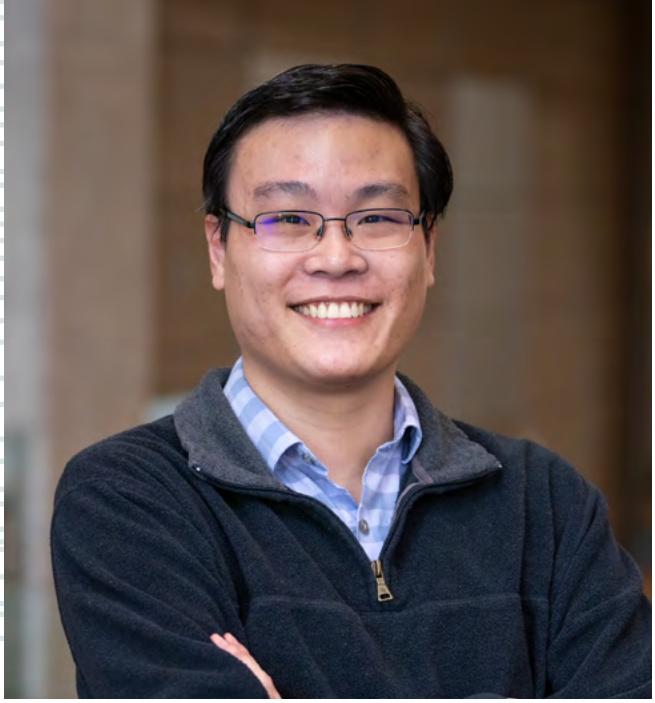
IN CONTEXT

This FY2022–23 project

- aligns with the SEI technical objective to bring capabilities that make new missions possible or improve the likelihood of success of existing ones
- aligns with the DoD software strategy to realize computational and algorithmic advantage
- aligns with the DoD software strategy to accelerate the delivery and adoption of AI



Deepfake detection methods generally involve finding the face, extracting facial landmarks and normalizing them, applying masking and/or noise, and sending it to a pre-trained image detector.



AI EVALUATION METHODOLOGY FOR DEFENSIVE CYBER OPERATOR TOOLS

PRINCIPAL INVESTIGATOR

DR. SHING-HON LAU
Senior Cybersecurity Engineer

1

2

3

4

5

6

SEI COLLABORATORS

1. GRANT DEFFENBAUGH
Senior Security Researcher

2. LYNDSI HUGHES
Systems Engineer

3. JARRETT BOOZ
Associate Cybersecurity Engineer

4. KEN BROWN
Assistant Cybersecurity Engineer

5. BRANDON MARZIK
Associate Cybersecurity Engineer

6. DERRICK SPOONER
Member of the Technical Staff—Senior Engineer

As AI-powered network defense cyber operations (DCO) tools, or AI-based defenses, become more prevalent and critical to our security, our adversaries are stepping up their efforts to evade detection by these technologies by leveraging adversarial AI techniques, such as data obfuscation and data poisoning [Subedar 2019]. At this time, to our knowledge, no publicly available method exists for thoroughly and methodologically evaluating the capabilities of an AI-enabled DCO tool nor to quantify the degree to which those capabilities remain effective despite adversarial manipulation. The DoD requires an evaluation methodology to practically test the defensive capabilities of an AI defense that is protecting a network.

The objective of this project is to develop a methodology for evaluating the capabilities of an AI defense using publicly available information of defensive network capabilities. Without such a methodology, DoD organizations must either (1) refuse to use AI defenses since they cannot properly test them, (2) apply traditional cybersecurity testing techniques that do not test for AI-specific properties, such as susceptibility to data poisoning, or (3) perform unprincipled ad hoc testing. None of these three options are satisfactory; they can result in the DoD having an unjustified confidence in the quality of their cyber defenses.

The goal of this project is to create a two-part methodology that will

- enable the evaluation capabilities of an AI-enabled network DCO tools
- enumerate the principles by which the efficacy of an AI-based DCO tool might be reduced when subjected to adversarial evasions

The completed extensible evaluation methodology will

- produce a new capability for the DoD—to test and evaluate the defensive capabilities of an AI defense under realistic conditions
- represent an increase in the state-of-the-art in broader cybersecurity, as there is not yet a principled methodology for evaluating the defensive capabilities for an AI defense for enterprise networks
- allow the DoD to repeatably test AI defenses to examine whether they have the desired defensive benefits, representing an increase in capability
- lead to a deeper understanding about detecting and mitigating obfuscations and data poisoning with next-generation DCO tools

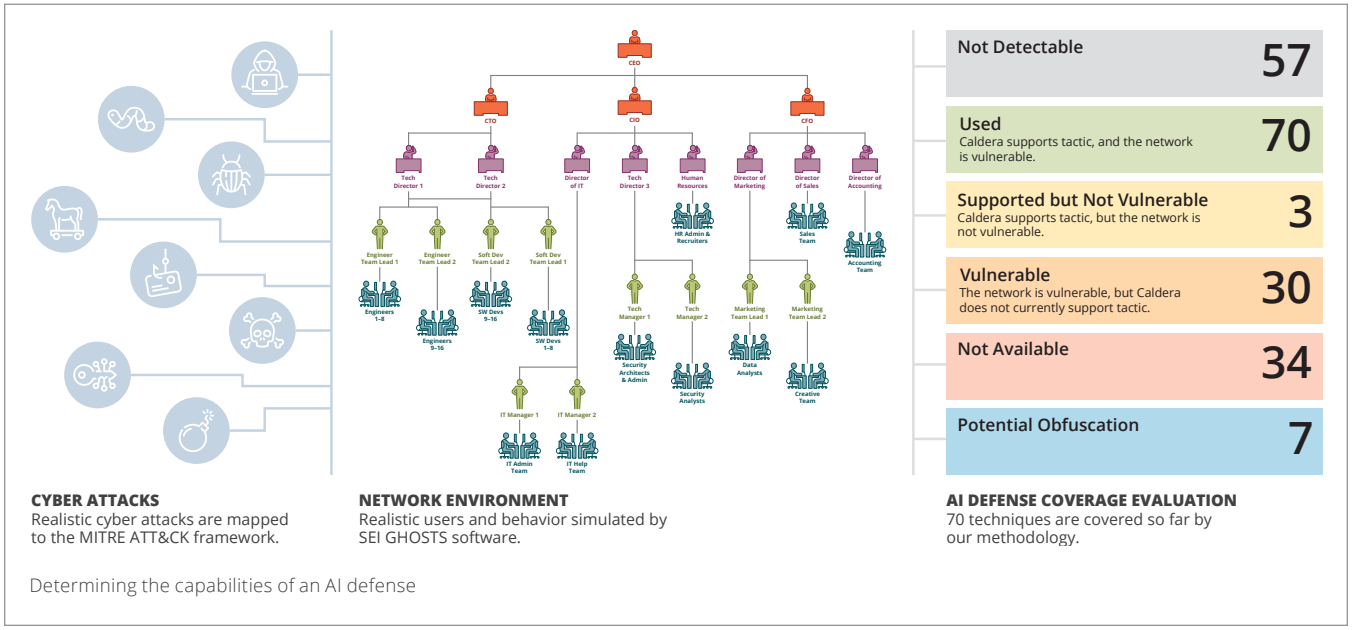
The objective of this project is to develop a methodology for **evaluating the capabilities of an AI defense.**

Currently, the project has developed an initial methodology that will permit the DoD to test and evaluate the defensive capabilities of an AI defense under realistic conditions. This methodology is currently undergoing additional refinement and expansion to better increase the information revealed by its application.

IN CONTEXT

This FY2022–23 project

- will allow DoD organizations to assess whether the AI defenses of today or tomorrow would enhance the protections of critical networks
- aligns with the CMU SEI technical objective to bring capabilities through software that make new missions possible or improve the likelihood of success of existing ones and to be trustworthy in construction and implementations
- aligns with the CMU SEI technical objective to ensure the cadence of acquisition, delivery, and fielding is responsive to and anticipatory of the operational tempo of DoD warfighters and that the DoD is able to field these new software-enabled systems and their upgrades faster than our adversaries
- aligns with the DoD software strategy to accelerate the delivery and adoption of AI



Determining the capabilities of an AI defense


DAY 2 DA


AY 2
DAY 2

CHAIN GAMES: POWERING AUTONOMOUS THREAT HUNTING

PRINCIPAL INVESTIGATOR

PHIL GROCE
Senior Network Defense Analyst

1

2

3

SEI COLLABORATORS

1. MATT SISK
Member of the Technical Staff

2. DR. JOSHUA FALLON
Senior Network Defense Analyst

EXTERNAL COLLABORATOR

3. DR. CHRISTOPHER KIEKINTVELD
Associate Professor, Computer Science
University of Texas at El Paso (UTEP)

Assuring information system security requires not just preventing system compromises but finding adversaries already present in the network before they can take action on their objectives. Defensive computer operations (DCO) personnel find the technique of cyber threat hunting the most effective approach for identifying such threats. While the Department of Defense (DoD) does conduct human-driven threat hunting, it often does so only when resources are not devoted to other demands. The time, expense, and expert resources required for cyber threat hunting typically preclude comprehensive investigation. However, an autonomous threat-hunting tool could run more pervasively, achieve standards of coverage currently considered impractical, and significantly reduce competition for limited analyst resources.

In this project, we take on this challenge by developing algorithms to enable fully autonomous threat hunting by modeling threat hunting as a Cyber Camouflage Game (CCG), a type of mathematical game played between a "probing" player (analogous to a threat hunter) and a potentially deceptive "target" (analogous to an attacker). We will test these algorithms in a simulation environment, and evaluate success using metrics derived from CCG analysis and the threat-hunting domain. Cloud telemetry data will be used to develop and verify the hunt algorithms, assessing the sufficiency of this data for threat hunting, and identifying potential gaps that can be fed back into vendor requirements and open standards to make threat hunting more effective in cloud-native environments.

The **time, expense, and expert resources** required for cyber threat hunting typically **preclude comprehensive investigation.**

Our initial investigation is bounded to

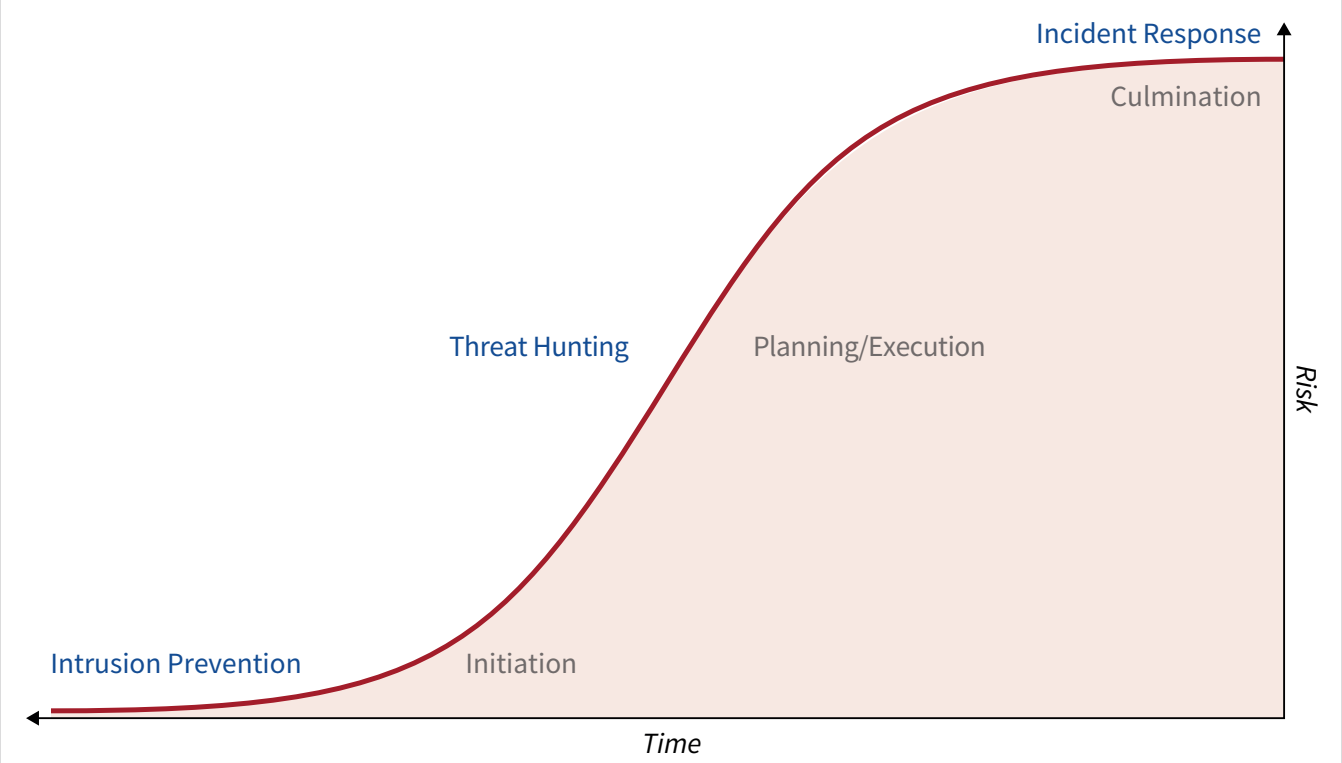
- Microsoft Azure cloud hosting and services (and evaluation of Azure cloud telemetry data)
- a containerized, infrastructure-as-a-service (IaaS) application architecture intended to be representative of DoD enterprise applications migrated to the cloud
- an attacker already resident in the environment and seeking to stage application data for later exfiltration

If our work is successful, threat-hunt operators will confirm that algorithms developed from game-theoretic analysis successfully identify attacker-controlled infrastructure as well as or better than the traditional state of the practice within the investigatory constraints.

IN CONTEXT

This FY2022–23 project

- aligns with the SEI technical objective to bring capabilities that make new missions possible or improve the likelihood of success of existing ones
- aligns with the DoD software strategy to attain autonomous cyber operations and resilience in DoD missions



Threat hunting takes time and skill. Inexpensive, faster hunting could investigate more data sources, coordinate for coverage, and help triage human threat hunts. The key to faster, less expensive threat hunting is autonomy.

MATURING ASSURANCE CONTRACTS IN MODEL-BASED ENGINEERING

PRINCIPAL INVESTIGATOR

DR. DIONISIO DE NIZ
Technical Director

1

2

3

4

5

6

7

8

9

10

SEI COLLABORATORS

1. DR. BJORN ANDERSSON
Principal Researcher

2. DR. JEROME HUGUES
Senior Researcher

3. DR. SAM PROCTER
Senior Researcher

4. JOHN HUDAK
Principal Engineer

5. SHOLOM COHEN
Principal Engineer

6. LUTZ WRAGE
Senior Researcher

7. DR. AARON GREENHOUSE
Senior Researcher

8. DR. RUBEN MARTINS
Assistant Research Professor

9. DR. GABRIEL MORENO
Principal Researcher

10. JOSEPH SEIBEL
Software Engineer

Architectural models and analyses using the Architecture Analysis and Design Language (AADL) have proven very useful in discovering and correcting problems with early design decisions that can be very costly if discovered late. For example, the Army Software Engineering Directorate (SED) Software Airworthiness and Safety Lab (SASL) utilized a model-based requirement verification technique with AADL on the Apache Flight Management Computer Obsolescence (FMCO) software and discovered 16 issues linked to 103 individual requirements [Feiler 2012]. The report of this work estimated that the corrections of each issue could cost $10k during architectural design as opposed to $3M if discovered during flight test.

Highlighting the SEI's **expertise in model-based systems** engineering, Maturing Assurance Contracts in Model-Based Engineering **uses mathematically sound formalisms** internally to ensure users make their models analyzable.

As these models continue to be used and refined with additional analyses applied, their benefits multiply, leading to implementations that avoid catastrophic cost and schedule increases. However, while AADL architectural models raise the level of abstraction to simplify early design decisions, the consequences of these decisions can be complex. Furthermore, the algorithms used to analyze these consequences to ensure the system exhibits the necessary characteristics make complex assumptions about the model that can invalidate the analysis results if not properly validated. This creates a fundamental transition barrier that inhibits the DoD's digital engineering strategy, emphasizing the use of models and analyses in the design and engineering of new capabilities and systems.

This project proposes to make analysis assumption verification technology transition ready by developing the infrastructure to support analysis assumption verification and correction throughout the modeling process. That is, it will be ready (1) before we run an analysis to verify that the model conforms with the assumptions necessary to run the model successfully, (2) when we use early models with incomplete information to allow it to defer the verification of some assumptions, and (3) when we verify that the implementation matches the model and its assumptions.

More concretely, we will develop a contacts framework that can

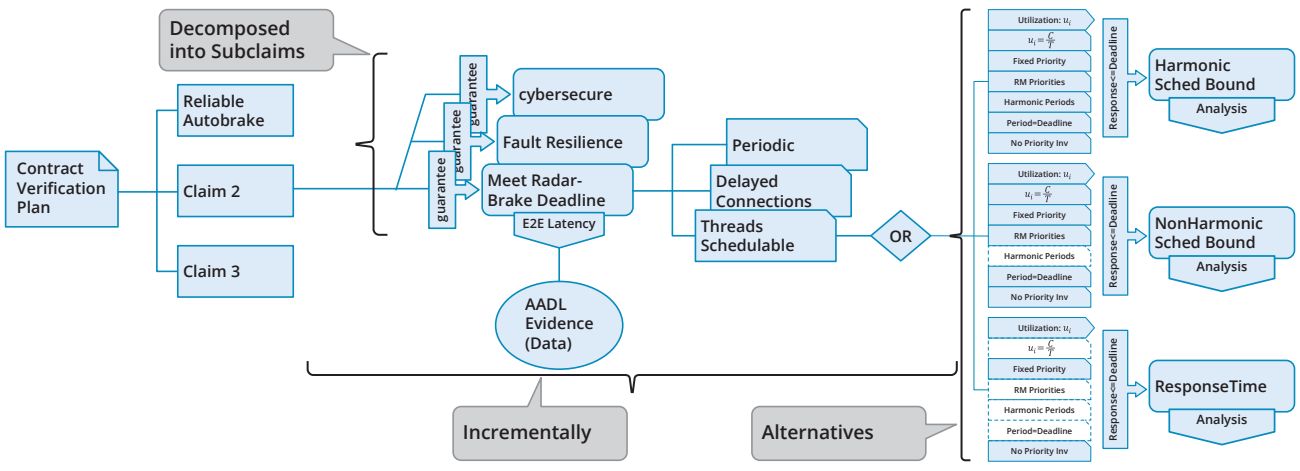
- describe and enforce assumptions of 75% more analyses than known state-of-the-art
- can incrementally refine and enforce assumptions of 75% more analyses than known state-of the art
- validate the conformance of 70% more assumptions in a system implementation than the known state-of-the art

Highlighting the SEI's expertise in model-based systems engineering, Maturing Assurance Contracts in Model-Based Engineering uses mathematically sound formalisms internally to ensure users make their models analyzable. At the end of the first year, this project has created the core infrastructure to validate assumptions, providing assistance to the designer to correct the assumptions and look for analysis alternatives.

IN CONTEXT

This FY2022–24 project

- builds on SEI expertise in model-based systems engineering and complements additional efforts currently underway focused on tool usability, training, and diffusion
- aligns with the CMU SEI technical objective to reduce and make predictable the cost of acquisition and operations, despite increased capability, and provides a cost advantage over our adversaries
- aligns with the CMU SEI technical objective to make the cadence of acquisition, delivery, and fielding responsive to and anticipatory of the operational tempo of DoD warfighters so that the DoD is able to field these new software-enabled systems and their upgrades faster than our adversaries



A formal argumentation structure was developed and refined to prove how the system designer's verification results guarantee assurance claims.

SAFETY ANALYSIS AND FAULT DETECTION ISOLATION AND RECOVERY SYNTHESIS FOR TIME-SENSITIVE CYBER-PHYSICAL SYSTEMS

PRINCIPAL INVESTIGATOR

DR. JEROME HUGUES
Senior Architecture Researcher

1

2

3

4

5

SEI COLLABORATORS

1. DR. AARON GREENHOUSE
Senior Architecture Researcher

2. KEATON HANNA
Assistant Software Engineer

3. DR. SAM PROCTER
Senior Architecture Researcher

4. JOSEPH SEIBEL
Member of the Technical Staff

5. LUTZ WRAGE
Senior Member of the Technical Staff

The operational complexity of cyber-physical systems (CPS) forces new autonomous features into day-to-day systems, such as vehicles and factories, a phenomenon termed *increasingly autonomous CPS systems* (IA-CPS) [Alves 2018]. IA-CPS have a complex architecture that weaves hardware, AI-enabled functions or decision-making processes, human operators, and software. They are time sensitive and substitute human actions with high-frequency real-time algorithms. In such systems, the conjunctions of faults and their timed propagation can cause fatal incidents, such as those involving autonomous cars. In these particular cases, the safety mechanisms were either too inefficient to prevent a fault or actually *caused* the incident.

This situation creates concerns for future DoD programs: These systems not only need to be able to detect failures and recover once, but they also need to be able to reconfigure multiple times—autonomously—as they adapt to different situations without human intervention.

SAFIR addresses **safety analysis** of time-sensitive CPS **in both** its theoretical and practical **dimensions**.

Implementing the DoD's AI vision requires advances in safety analysis, and fault detection isolation and recovery synthesis (or SAFIR) to (1) model and analyze dynamic reconfiguration and fault propagation due to fault sequences, and (2) enforce safe reconfiguration. In the first two years, SAFIR has investigated the properties a CPS architecture must demonstrate to integrate autonomy functions and fulfill safety objectives and how to integrate them into a model-based systems engineering (MBSE) practice:

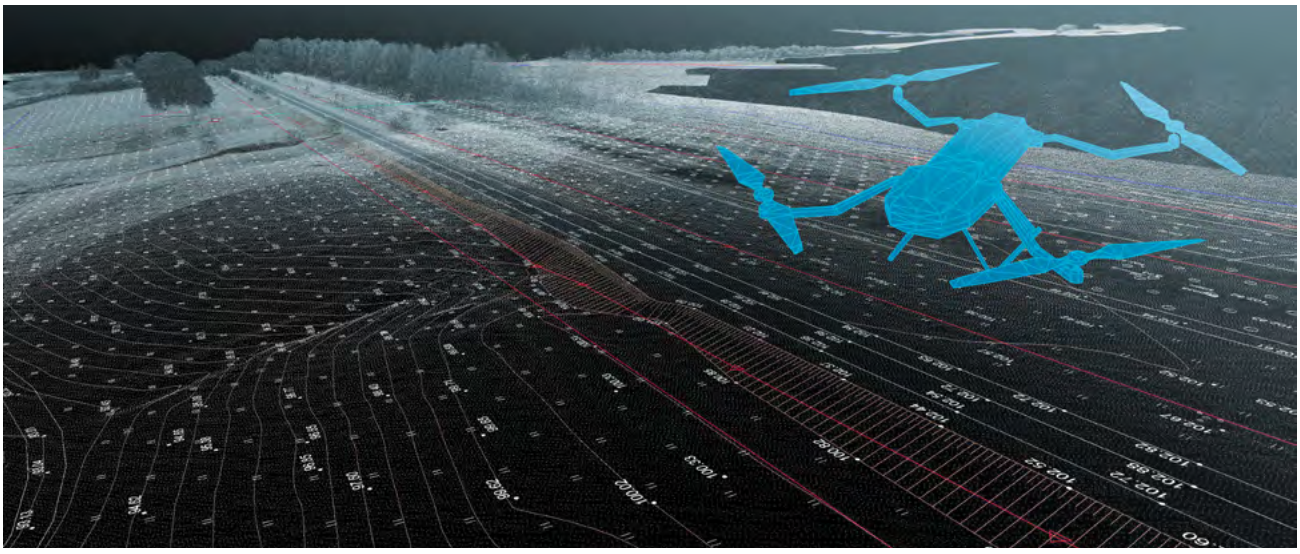
- SAFIR improved the IA-CPS systems engineering body of knowledge, focusing on safety mechanisms, from design to verification and validation (V&V), using model-based engineering (MBE) techniques.
- SAFIR delivered an updated taxonomy to express fault models of IA-CPS and derive efficient detection mechanisms. Together with Georgia Tech, we explored the techniques to detect tampering with sensors data either in case of faults or cyber attacks, conditions for detectability of these attacks, and the possibility to derive a controller for an IA-CPS in the case of timing errors.
- SAFIR delivered formally backed reasoning and simulation capabilities for IA-CPS architectures by mechanizing the SAE AADL language using the Coq theorem prover, expanding V&V capabilities for MBE tooling.
- SAFIR defined and implemented the Architecture-Supported Audit Processor (ASAP): a tool that generates a number of safety-specific system views that deeply integrate a system's architecture and arguments.

SAFIR addresses safety analysis of time-sensitive CPS in both its theoretical and practical dimensions, and it contributes to the SEI's line of research on artificial intelligence and autonomy. At the end of the second year, SAFIR has established the theoretical foundation to perform safety evaluations in the context of time-dependent failure conditions.

IN CONTEXT

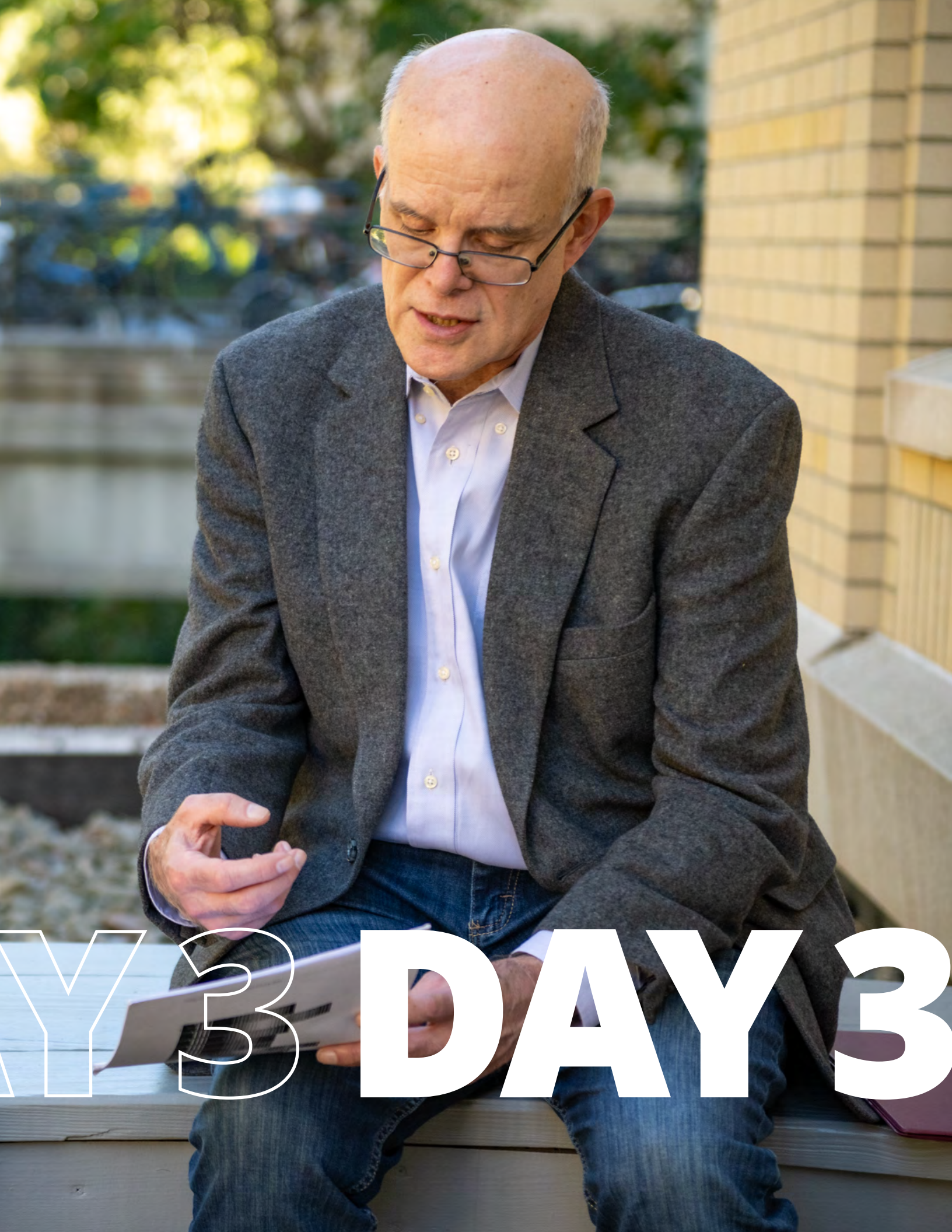
This FY2021–23 project

- builds on SEI expertise in MBSE, safety analysis, and the AADL language and extends past contributions from Integrated Safety and Security Engineering (ISSE) and TwinOps
- aligns with the CMU SEI technical objective to bring capabilities through software that make new missions possible or improve the likelihood of success of existing ones and to be trustworthy in construction and implementations
- aligns with the CMU SEI technical objective to be resilient in the face of operational uncertainties, including known and yet-unseen adversary capabilities



SAFIR expands MBSE with mathematically grounded techniques to analyze the architecture of AI-based cyber-physical systems—such as drones—and derive an argument on their safety.


DAY 3 DA


DAY 3
DAY 3

REFACTORING FOR SOFTWARE ISOLATION

PRINCIPAL INVESTIGATOR

JAMES IVERS
Principal Engineer

1

2

3

4

5

6

7

SEI COLLABORATORS

1. CHRIS SEIFRIED
Associate Engineer

2. JONNY LOUNGANI
Assistant Software Engineer

3. DR. IPEK OZKAYA
Technical Director, Engineering Intelligent Software Systems

4. MARIO BENITEZ
Software Architect

5. TAMARA MARSHALL-KEIM
Team Lead of Technical Communications

6. GREG SUCH
Program Development Manager

7. DR. ANDREW KOTOV
Software Architect

EXTERNAL COLLABORATORS

DR. MAROUANE KESSENTINI
Oakland University

DR. KHOULOUD GAALOUL
University of Michigan

ESTHER BAE
Carnegie Mellon University

OWEN DONOVAN
Pennsylvania State University

Software-reliant systems need to evolve over time to meet new requirements and take advantage of new technology. However, all too often, the structure of software becomes too complicated to allow rapid and cost-effective improvements. This challenge is common in long-lived DoD systems and not uncommon even in newer systems, and it can result in a need to perform large-scale refactoring. As noted in a recent SEI survey [Ivers 2022a], industry practitioners reported that large-scale refactoring exercises lack robust tool support and consume considerable effort (an average of 1,500 staff days).

This project extends our **refactoring assistant** to refactor Java or C# code, matures its automated software isolation capability, and enhances its usability, solution completeness, and solution quality.

In the FY19–21 Line project, Untangling the Knot [Ivers 2022b], we created a refactoring assistant that automates the majority of the work required for software isolation, a key element of large-scale refactoring projects that modularizes existing code for use in new contexts. The refactoring assistant combines static code analysis, formalized refactorings, and a multi-objective genetic algorithm to search hundreds of thousands of combinations of primitive refactorings applied to different code elements to find solutions that solve the software isolation problem in balance with other criteria (e.g., code size, maintainability, or understandability). Results from applying the current prototype to multiple open source and DoD projects demonstrated an ability to scale to at least 1.2M source lines of code and generate solutions that, on average, solve 88% of the structural couplings that hinder software isolation.

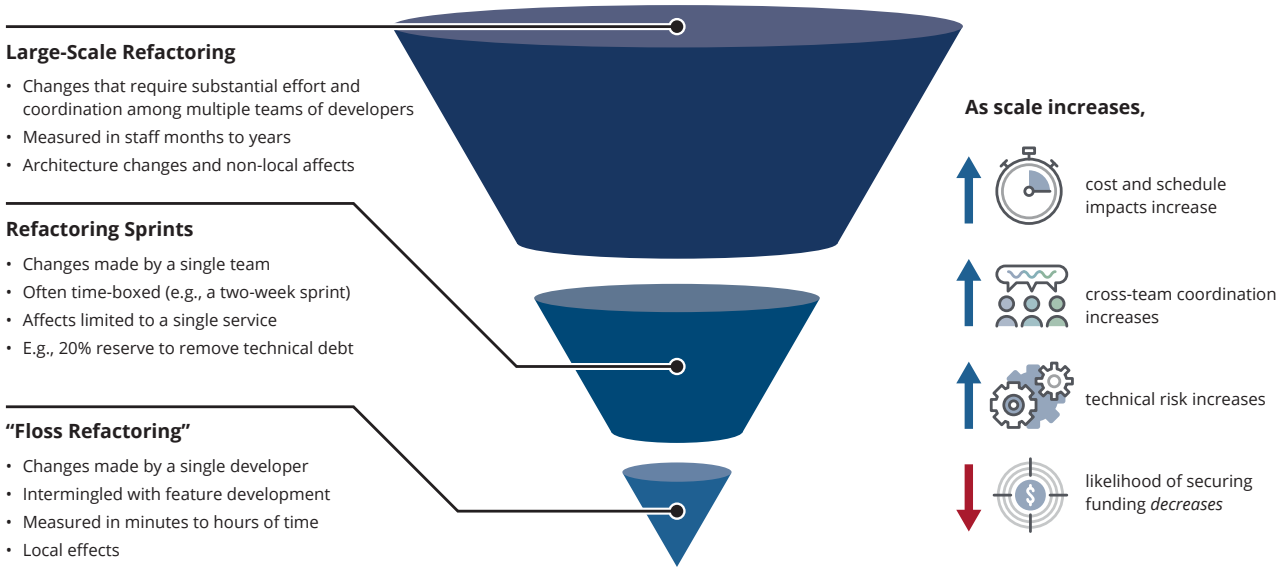
In this project, we have extended the refactoring assistant to address a second programming language and are now able to refactor Java or C# code. We are further maturing the automated software isolation capability of this refactoring assistant and validating this research by applying it to open source projects, and assessing success against the following objectives:

- Usability: Developers using the refactoring assistant will have to review 70% fewer generated refactorings to select and understand their preferred solution.
- Solution completeness: The refactoring assistant generates solutions that resolve at least 90% of the structural couplings that hinder software isolation.
- Solution quality: Experienced developers who use the refactoring assistant agree with and use at least 80% of the recommended refactorings in solutions without modification.

IN CONTEXT

This FY2022–24 project

- builds on prior DoD line-funded research in software architecture analysis, static code analysis, and identifying technical debt
- aligns with the CMU SEI technical objective to make software delivery timely so that the cadence of acquisition, delivery, and fielding is responsive to and anticipatory of the operational tempo of DoD warfighters
- aligns with the DoD software strategy to mitigate technical debt by directly recommending code changes to resolve some forms of technical debt
- addresses a widespread, recurring need in software organizations (As requirements and technology are never frozen in time, the need to adapt working software to new contexts is likely to remain a common need across many software systems.)



By applying our prototype refactoring assistant to multiple open source and DoD projects, we demonstrated an ability to scale to at least 1.2M source lines of code and generate solutions that, on average, solve 88% of the structural couplings that hinder software isolation.

AUTOMATED DESIGN CONFORMANCE DURING CONTINUOUS INTEGRATION

PRINCIPAL INVESTIGATOR

DR. ROBERT NORD
Principal Member of the Technical Staff

1

2

3

4

5

SEI COLLABORATORS

1. JAMES IVERS
Principal Engineer

2. DR. JOHN KLEIN
Principal Member of the Technical Staff

3. LENA PONS
Software Architecture and AI Researcher

4. CHRIS SEIFRIED
Associate Engineer

5. DR. JOSH FALLON
Senior Network Defense Analyst

Software architecture enables our ability to innovate through extensible design and to deliver future growth in capability that is affordable and timely. To reduce the time needed to field capabilities and to lower lifecycle costs, the DoD has instructed program managers to consider a modular open systems approach (MOSA). MOSA promotes extensibility through technical standards such as the Future Airborne Capability Environment (FACE). Achieving these qualities depends on how the design allocates responsibilities to components and on what relationships are allowed among these components, which means that it depends on the software architecture. Co-evolving architecture and code is good practice for both new development and modernization, and it is increasingly important as programs adopt a software acquisition pathway using Agile and DevSecOps approaches.

The end goal—to build and evolve systems that provide timely and cost-effective capability to users—is achieved only if the implemented code conforms to the architecture. However, a gap exists in verifying that implemented capabilities satisfy the design constraints of the intended architecture.

This project developed an automated conformance checker prototype that can be used in a continuous integration workflow to detect and report nonconformances within minutes, instead of the months or years it takes to discover these problems today.

The central research of this project is automatic recognition of abstractions commonly used in software architecture from source code. This includes extracting relevant facts from the source code and related artifacts, inferring architecture abstractions from those facts, and synthesizing the abstractions into a design. Inferring design from code is hard because there are few indications of intent in the code and because implementations of an abstraction show significant variations both within a project and across projects. Many software projects reuse one or more off-the-shelf frameworks, and we use information implicit in these frameworks to advance automation in architecture analysis to extract design as implemented in C++ source code. We are focusing on detecting nonconformance in systems using architecture communication styles, such as publish-subscribe, that are essential to achieving the extensibility goals of MOSA.

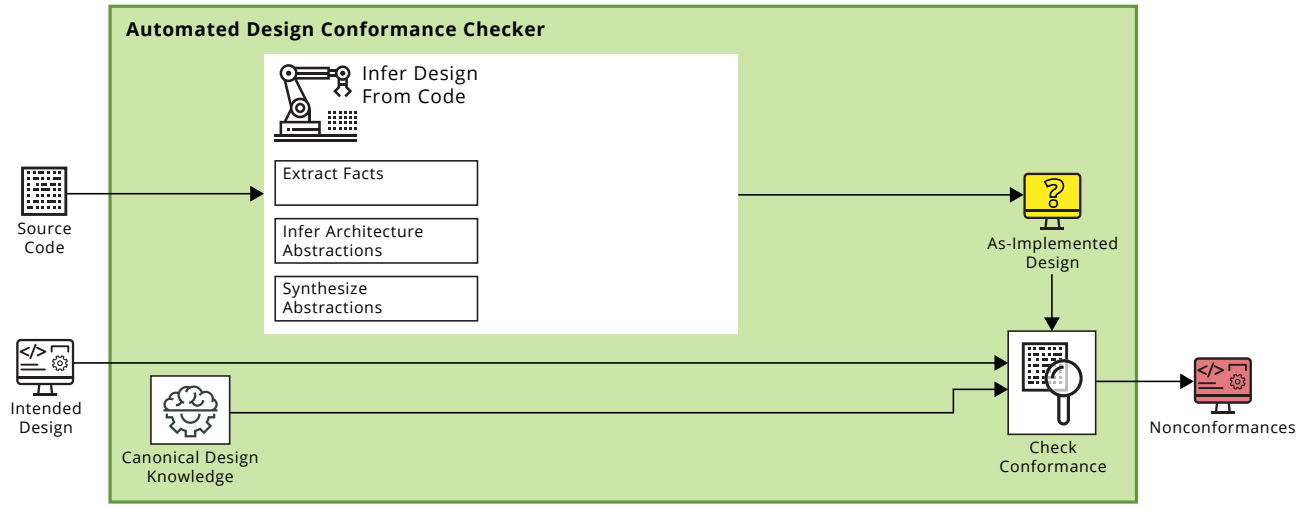
This project developed an **automated conformance checker** prototype that can be used in a continuous integration workflow to detect and report nonconformances **within minutes,** instead of the months or years it takes to discover these problems today.

The conformance checker will benefit developers and program managers. Developers can detect problems continuously and near the time they are introduced, allowing faster and more economical realignment of implementation and architecture. Program managers can hold developers (contractor or organic) accountable for delivering sustainable systems.

IN CONTEXT

This FY2020–22 project

- aligns with SEI strategic focus areas of timely and trustworthy software by introducing automation into the development and acquisition lifecycle
- aligns with the DoD software strategy to mitigate technical debt by identifying instances of technical debt as it is introduced
- advances our understanding of how architecture inference differs from other code inference problems and the research challenges that follow from these differences



The key to this work is new research inferring design information from source code. Detecting design constructs is challenging due to imprecise definitions of abstractions, variation in implementation, and limits of fact-gathering analyses.

AUTOMATED CONTINUOUS ESTIMATION FOR PIPELINES OF PIPELINES

PRINCIPAL INVESTIGATOR

DR. WILLIAM NICHOLS
Principal Member of the Technical Staff

1

2

3

4

5

6

SEI COLLABORATORS

1. LUIZ ANTUNES
DevOps Engineer

2. ROB MCCARTHY
DevOps Engineer

3. DR. CHRIS MILLER
Member of the Technical Staff—Senior Researcher

4. JULIE COHEN
Senior Member of the Technical Staff

5. MELISSA LUDWICK
Member of the Technical Staff

6. AKIA WILLIAMS
Senior Administrative Assistant

The DoD wants to transform defense acquisitions and enable rapid and iterative delivery of software capability through new acquisition pathways and by exploiting DevSecOps. The emphasis on DevSecOps creates opportunities for increased automated data collection and analysis to significantly shorten feedback cycles and provide program information at the speed of relevance. Currently, lack of automation for collection, archiving (for ML), and analysis fails to match the release-feedback cadence.

… this project **envisions instrumenting** the complex software production environments typical of the DoD to **continuously monitor** the DevSecOps pipeline and use that data to **continuously update** estimates for cost, schedule, and quality.

What's more, the DevOps Research and Assessment (DORA) metrics typical of DevSecOps (lead time for change, deployment frequency, time to restore service, and change failure rate) may not be sufficient for complex DoD environments, particularly under the new Software Acquisition Pathway. At a minimum, current metrics are not causal, and they are connected to Ops rather than to program management decisions.

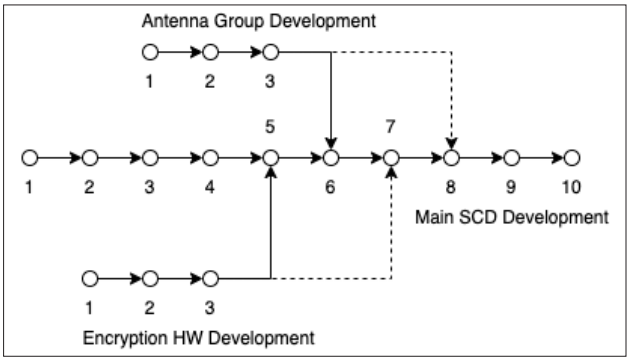
To address this situation, this project envisions instrumenting the complex software production environments typical of the DoD to continuously monitor the DevSecOps pipeline and use that data to continuously update estimates for cost, schedule, and quality. The data will also be available for AI and ML. The ultimate vision is to provide program managers and other stakeholders with a smart dashboard that displays timely, relevant, and correlated data. They will use this dashboard to obtain status; evaluate risks; and predict cost, schedule, and quality without the latency and disruption of data calls.

We're working to address basic but crucial questions, such as, "If x changes, how will that affect the overall program?" Presently, answers to these questions elude program managers of large complex cyber-physical development programs in a fast-paced Agile environment.

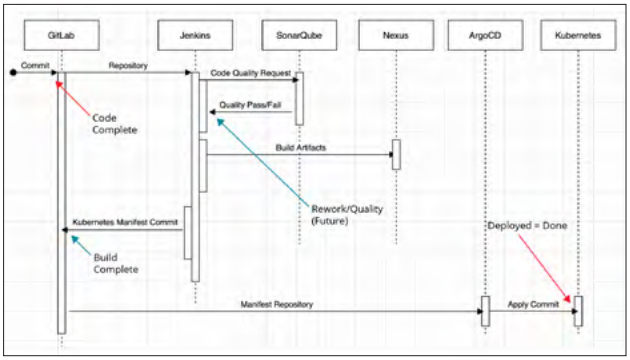
IN CONTEXT

This FY2022–23 project

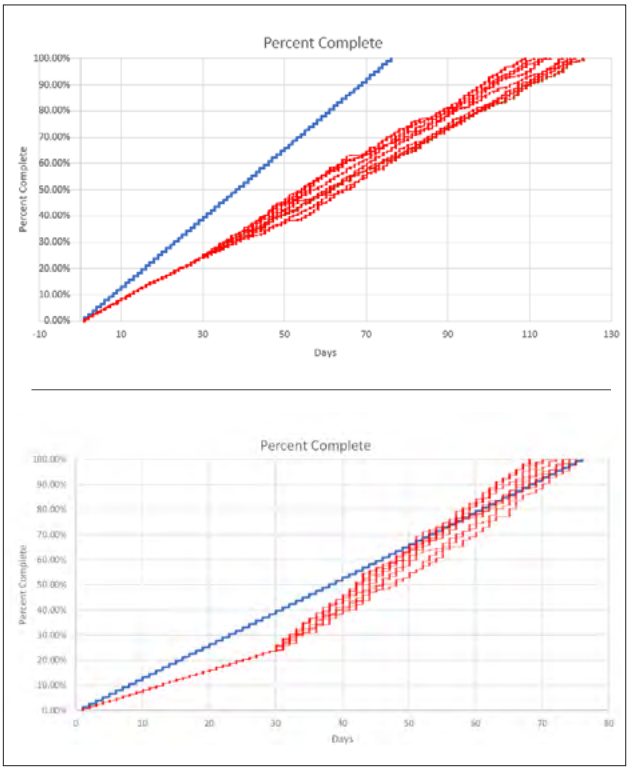
- codifies advanced continuous integration and continuous delivery (CI/CD) acquisition practices
- formalizes the development, integration, and use of models to advance software acquisition
- improves the efficacy of the software acquisition process to realize better performance and affordability
- aligns with the DoD software strategy to advance DevOps



Integration of Multiple Pipelines



Metric Data Collection Automation



Timely Analysis and Forecasting

The ultimate vision of this work is to provide program managers and other stakeholders a smart dashboard that displays timely, relevant, and correlated data on status, risks, cost, schedule, and quality without the latency and disruption of data calls.

ADVANCING ALGORITHMS FOR FILE DEDUPLICATION ACROSS CONTAINERS

PRINCIPAL INVESTIGATOR

KEVIN PITSTICK
Senior Software Engineer

1

2

3

4

5

6

7

SEI COLLABORATORS

1. SEBASTIÁN ECHEVERRÍA
Senior Software Engineer

2. BRANDON BORN
Associate Software Engineer

3. BRENT CLAUSNER
DevOps Engineer

4. CARL GRUHN
Assistant Software Engineer

5. GARY ZHANG
Software Developer Intern

6. LIHAN ZHAN
Assistant Software Engineer

7. JOSEPH BELL
Associate Software Engineer

Container software virtually packages and isolates applications for deployment. It can operate over multiple network resources so applications can run in isolated user spaces (containers) in any cloud (or non-cloud) environment. The Department of Defense (DoD) wants to use containers to support its vision of a cloud-to-edge continuum in which capabilities packaged as containers are pushed from the cloud to edge devices to support localized data processing. However, devices deployed at the tactical edge are resource limited and commonly operate over disconnected, intermittently connected, low-bandwidth (DIL) networks or hostile environments in which there is a high likelihood of bad actors trying to tamper with them.

The Department of Defense wants to use **containers** to support its vision of a **cloud-to-edge continuum** in which capabilities packaged as containers are pushed from the cloud to edge devices to support localized data processing.

To address these limitations, we developed an automated container image minimization technology. This technology combined and improved on two minimization approaches: pruning (removing unnecessary files from single images) and deduplication (combining shared files across images into common layers). We focused on advancing the state-of-the-art in deduplication across container images.

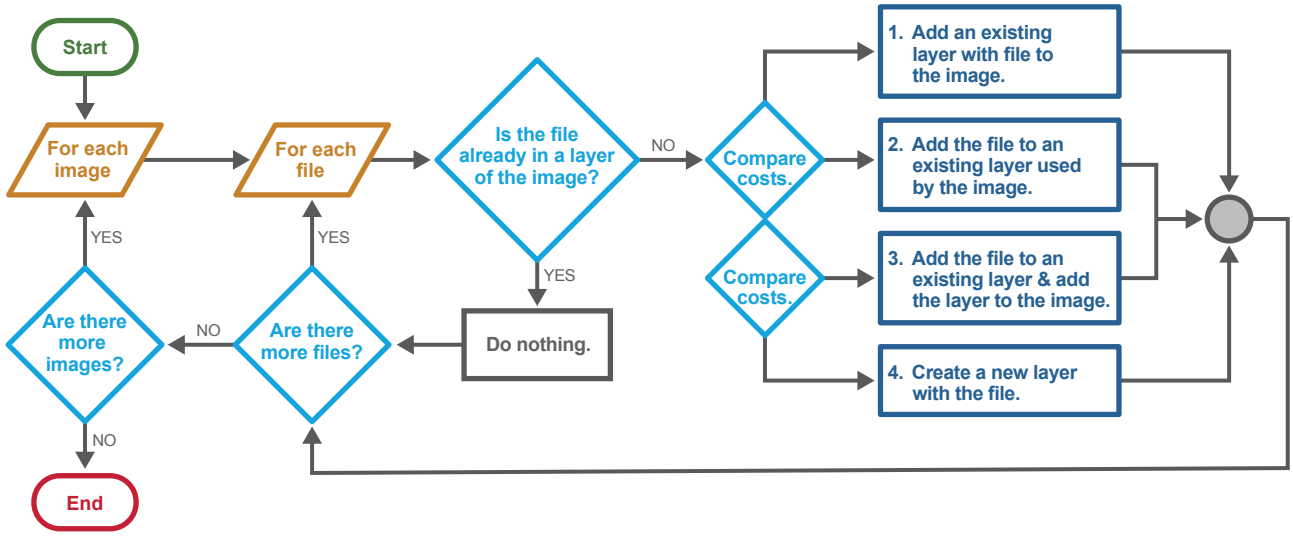
To create this new technology, we developed an algorithm for file deduplication across a collection of container images that can reduce container image storage usage and update bandwidth by up to 5–15% for multi-container deployments and by up to 10–30% for pruned container deployments. In our tests with real multi-container image systems, our algorithm deduplicates 100% of shared files and processes 10 images with 225,000 files in approximately 81 minutes.

This project focused on technology that supports the Open Container Initiative (OCI) standard because the DoD aims to avoid vendor lock-in and leverage OCI-compliant containers. Additionally, this project has the potential to accelerate the SEI's impact by open sourcing minimization algorithms to gain wider interest and adoption from industry and the DoD community.

IN CONTEXT

This FY2022 project

- aligns with the SEI technical objective to be trustworthy in construction and implementation and resilient in the face of operational uncertainties, including known and yet unseen adversary capabilities
- aligns with the SEI technical objective to be affordable such that the cost of acquisition and operations, despite increased capability, is reduced and predictable and provides a cost advantage over our adversaries
- aligns with the DoD software objective to enhance resilience



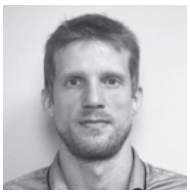
Our solution for pushing containers to the tactical edge employs a "greedy" algorithm that creates a new set of layers for every image with minimal duplicates.

SEMANTIC-EQUIVALENCE CHECKING OF DECOMPILED BINARIES

PRINCIPAL INVESTIGATOR

DR. WILL KLIEBER
Software Security Engineer

1



2

3

4

SEI COLLABORATORS

1. DAVID SVOBODA
Software Security Engineer

2. MIKE MCCALL
Software Security Engineer

3. DR. LORI FLYNN
Software Security Engineer

4. DR. RUBEN MARTINS
Assistant Research Professor

Assuring that fielded software is free of vulnerabilities exploitable by adversaries is critically important in military applications deployed by the Department of Defense (DoD). For systems that include binary-only software components, existing analysis solutions are limited, and any warnings of potential vulnerabilities would require significant manual effort by experts to investigate. Repairing binary code is even more difficult and expensive.

It is typically much easier to work with decompiled code than with raw machine code or assembly code, so using decompilation has the potential to greatly decrease the cost of finding and fixing vulnerabilities in binary code and to enable the DoD to fix potential vulnerabilities that might otherwise be cost prohibitive to investigate or repair. However, existing decompilers often do not produce code that is completely semantically faithful to the original binary.

Our tool can **identify** which decompiled functions are likely to be **semantically equivalent** to the original binary function and which are unlikely to be equivalent.

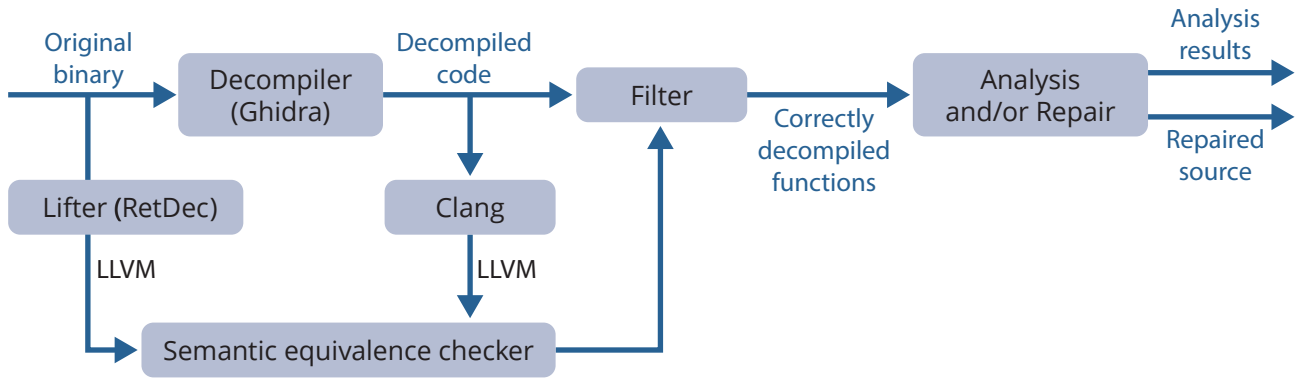
To address this challenge, we launched this project to develop and implement techniques for automated semantic-equivalence checking. Our tool can identify which decompiled functions are likely to be semantically equivalent to the original binary function and which are unlikely to be equivalent. Our ultimate goal is to make it practical for DoD to use existing source-code static analyzers and repair tools on decompiled code, thereby increasing trust in the software. The figure below shows a pipeline for using our tool in practice. The binary code is decompiled with Ghidra, and the resulting decompiled code is compared for semantic equivalence to the LLVM produced by an independent binary lifter such as RetDec. Only those functions that are semantically equivalent are passed along for static analysis and repair.

This collaborative effort was informed by work with personnel who have significant experience in software assurance at the DoD and who are familiar with the types of issues that DoD faces when performing software assurance on binary code.

IN CONTEXT

This FY2022 project

- aligns with the SEI technical objective to be trustworthy in construction and implementation and resilient in the face of operational uncertainties, including known and yet-unseen adversary capabilities
- aligns with the DoD software strategy to improve designed-in trustworthiness
- builds on previous SEI work on automated code repair and inference of memory bounds



Pipeline for Analysis and Repair of Binary Code

REFERENCES

[Alves 2018]
Alves, Erin E.; Bhatt, Devesh; Hall, Brendan; Driscoll, Kevin; Murugesan, Anitha; & Rushby, John. *Considerations in Assuring Safety of Increasingly Autonomous Systems.* Technical Report NASA/CR-2018-220080, NF1676L-30426, National Aeronautics and Space Administration. 2018. **https://ntrs.nasa.gov/api/citations/20180006312/downloads/20180006312.pdf**

[Feiler 2012]
Feiler, Peter H.; Hudak, John; & Meyers, B. Craig. *An Architecture-Centric Analysis of the Apache FMCO: Final Report.* CMU/SEI-2012-SR-012. Software Engineering Institute, Carnegie Mellon University. September 2012. Not publicly available.

[Ivers 2022a]
Ivers, James; Nord, Robert L.; Ozkaya, Ipek; Seifried, Chris; Timperley, Christopher S.; & Kessentini, Marouane. Industry Experiences with Large-Scale Refactoring. *Foundations of Software Engineering: Software Engineering in Practice* (ESEC/FSE). November 2022. **https://doi.org/10.48550/arXiv.2202.00173**

[Ivers 2022b]
Ivers, James; Seifried, Chris; & Ozkaya, Ipek. Untangling the Knot: Enabling Architecture Evolution with Search-Based Refactoring. *19th IEEE International Conference on Software Architecture* (ICSA 2022). March 2022. **https://doi.org/10.1109/ICSA53651.2022.00018**

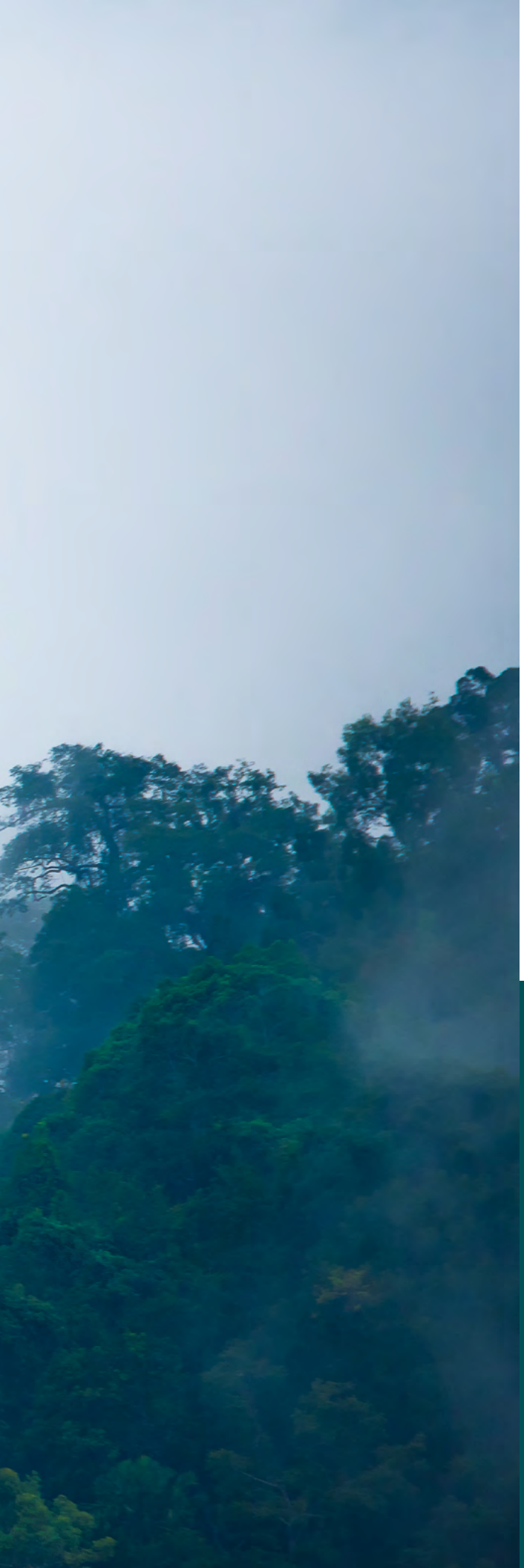
[Lewis 2021]
Lewis, Grace A.; Bellomo, Stephany; & Ozkaya, Ipek. Characterizing and Detecting Mismatch in Machine-Learning-Enabled Systems. Pages 133–140. In 2021 *IEEE/ACM 1st Workshop on AI Engineering—Software Engineering for AI (WAIN).* 2021. **https://doi.org/10.1109/WAIN52551.2021.00028**

[NIST 2022]
NIST. *The Economic Impacts of Inadequate Infrastructure for Software Testing.* May 2002. **https://www.nist.gov/system/files/documents/director/planning/report02-3.pdf**

[Subedar 2019]
Subedar, Mahesh; Ahuja, Nilesh; Krishnan, Ranganath; Ndiour, Ibrahima J.; & Tickoo, Omesh. Deep Probabilistic Models to Detect Data Poisoning Attacks. In *Fourth Workshop on Bayesian Deep Learning (NeurIPS 2019), Vancouver, Canada.* 2019. **http://bayesiandeeplearning.org/2019/papers/112.pdf**

COPYRIGHT

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.



ABOUT US

The Software Engineering Institute (SEI) at Carnegie Mellon University is a Federally Funded Research and Development Center (FFRDC)—a nonprofit, public–private partnership that conducts research for the United States government. One of only 10 FFRDCs sponsored by the U.S. Department of Defense (DoD), the SEI conducts R&D in software engineering, systems engineering, cybersecurity, and many other areas of computing, working to introduce private-sector innovations into government.

As the only FFRDC sponsored by the DoD that is also authorized to work with organizations outside of the DoD, the SEI is unique. We work with partners throughout the U.S. government, the private sector, and academia. These partnerships enable us to take innovations from concept to practice, closing the gap between research and use.

CONTACT US

Carnegie Mellon University
Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213-2612

412.268.5800 | 888.201.4479
sei.cmu.edu | info@sei.cmu.edu