REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
01-07-2022	Final Report	7-Jan-2019 - 19-Mar-2022

4. TITLE AND SUBTITLE
Final Report: Predictive Failure Avoidance

5a. CONTRACT NUMBER
W911NF-19-1-0054

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611102

6. AUTHORS

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
University of Virginia
The Rector and Visitors of the University of Virginia
1001 North Emmet Street
Charlottesville, VA 22901 -4195

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
74823-NS.17

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Matthew Dwyer
UU	UU	UU	UU		19b. TELEPHONE NUMBER 403-243-5206

RPPR Final Report
as of 09-Aug-2022

Agency Code: 21XD

Proposal Number: 74823NS **Agreement Number: W911NF-19-1-0054**
INVESTIGATOR(S):

Name: Ph.D. Matthew Dwyer
Email: matthewbdwyer@virginia.edu
Phone Number: 4032435206
Principal: Y

Name: Ph.D. ThanhVu Nguyen
Email: tnguyen@unl.edu
Phone Number: 4024725086
Principal: N

Organization: **University of Virginia**
Address: The Rector and Visitors of the University of Virginia, Charlottesville, VA 229014195
Country: USA

DUNS Number: 065391526 EIN: 546001796
Report Date: 19-Jun-2022 Date Received: 01-Jul-2022
Final Report for Period Beginning 07-Jan-2019 and Ending 19-Mar-2022
Title: Predictive Failure Avoidance
Begin Performance Period: 07-Jan-2019 **End Performance Period:** 19-Mar-2022
Report Term: 0-Other

Submitted By: Ph.D. Matthew Dwyer Email: matthewbdwyer@virginia.edu
Phone: (403) 243-5206

STEM Degrees: 1 **STEM Participants:** 3

Major Goals: The goals of the PFA project are to explore automated techniques for failure avoidance at runtime. The novel approach is to use sophisticated program analyses to: (1) predict when and where failures may happen, (2) synthesize failure avoidance logic that modifies the state of the program to avoid potential failures, and (3) inject monitoring to detect when failures happen at runtime and when they do execute the failure avoidance logic.

The project involves a close interplay between static analysis and verification, program synthesis methods, and runtime monitoring. Advances in each of these areas is required to realize the PFA vision and major goals of the project in these areas are:

Goal 1: Develop scalable and accurate modular analysis techniques that can detect and characterize when failures may occur at runtime.

Goal 2: Develop techniques for characterizing the space of failure avoiding program executions.

Goal 3: Develop synthesis techniques that can generate program fragments that are able to perturb program execution onto failure avoiding executions.

Goal 4: Incorporating optimizations techniques into synthesis to minimize the state modifications that are generated by synthesized actions.

Goal 5: Develop monitoring and instrumentation to enact synthesized actions.

Goal 6: Evaluate the cost and effectiveness of PFA in the context of C programs.

Accomplishments: Over the course of the project the team made substantial progress towards these goals. A variety of solutions for the goals outlined above were developed, implemented, evaluated and were reported on through published works.

RPPR Final Report
as of 09-Aug-2022

The original conception of the PFA project required that a software system have, either, system-level or component-level correctness specifications that would guide the static analysis and dynamic repair approaches. Unfortunately it is rare to find such specifications even for well-developed safety critical systems.

Consequently, the project added an additional technical goal to develop improved state-of-the-art specification inference techniques. The development of the SymInfer approach, as reported in the published TSE paper and disseminated through the DIG web-site, is the realization of this goal. The use of these approaches will be essential in automating the PFA vision.

The original conception of the PFA project sought to combine static analysis with dynamic repair techniques, but it did not anticipate the potential benefits of using multiple analysis algorithms in a synergistic fashion to improve the accuracy and performance of the approach.

The most recent work on this project has developed a novel approach to predicting which, from a family of, static analysis techniques are most cost-effective for a given instance of an analysis problem. While this advance came late in the project it is a completely novel approach that offers significant potential and the initial research findings are under submission at present (they give appropriate credit to this project).

Training Opportunities: The project has supported three graduate students over the past year. Will Leeson at the University of Virginia, Guolong Zhang at the University of Nebraska, and Didier Ishimwe at the University of Nebraska. Will is a full-time PhD student and Guolong and Didier are Phd and MS students working part time on this project.

Didier Ishimwe completed his Masters degree and has gone on to a PhD program. Will Leeson is continuing his PhD studies. Guolong Zhang completed his PhD degree shortly after the conclusion of this project (in April 2022). The training of these students is credited to the project.

Results Dissemination: The project disseminated its findings through two open-source software repositories with associated web-sites and through 15 fully-refereed publications (with another in submission).

Honors and Awards: During the course of the project the PI was: elevated to ACM Fellow, received two ACM SIGSOFT Impact Paper Awards, won the ACM SIGSOFT Distinguished Service Award, and was named the IEEE Computer Society's Harlan Mills award recipient.

Protocol Activity Status:

Technology Transfer: Nothing to Report

PARTICIPANTS:

Participant Type: PD/PI
Participant: Matthew Dwyer
Person Months Worked: 1.00 **Funding Support:**
Project Contribution:
National Academy Member: N

Participant Type: Co PD/PI
Participant: ThanhVu Nguyen
Person Months Worked: 1.00 **Funding Support:**
Project Contribution:
National Academy Member: N

Participant Type: Graduate Student (research assistant)
Participant: William Leeson

RPPR Final Report
as of 09-Aug-2022

Person Months Worked: 12.00 **Funding Support:**
Project Contribution:
National Academy Member: N

Participant Type: Graduate Student (research assistant)
Participant: Didier Ishimwe
Person Months Worked: 6.00 **Funding Support:**
Project Contribution:
National Academy Member: N

Participant Type: Graduate Student (research assistant)
Participant: Guolong Zhang
Person Months Worked: 12.00 **Funding Support:**
Project Contribution:
National Academy Member: N

ARTICLES:

Publication Type: Journal Article Peer Reviewed: Y **Publication Status:** 2-Awaiting Publicat
Journal: IEEE Transactions on Software Engineering
Publication Identifier Type: DOI Publication Identifier: 10.1109/TSE.2020.3016778
Volume: Issue: First Page #: 1
Date Submitted: 8/26/20 12:00AM Date Published:
Publication Location:
Article Title: Conditional Quantitative Program Analysis
Authors: Mitchell Gerrard, Mateus Borges, Matthew Dwyer, Antonio Fillieri
Keywords: program analysis, model counting, symbolic execution, conditional analysis, software reliability, software certification
Abstract: Standards for certifying safety-critical systems have evolved to permit the inclusion of evidence generated by program analysis and verification techniques. The past decade has witnessed the development of several program analyses that are capable of computing guarantees on bounds for the probability of failure. This paper develops a novel program analysis framework, CQA, that combines evidence from different underlying analyses to compute bounds on failure probability. It reports on an evaluation of different CQA-enabled analyses and implementations of state-of-the-art quantitative analyses to evaluate their relative strengths and weaknesses. To conduct this evaluation, we filter an existing verification benchmark to reflect certification evidence generation challenges. Our evaluation across the resulting set of 136 C programs, totaling more than 385k SLOC, each with a probability of failure below 10e-4, demonstrates how CQA extends the state-of-the-art.

Acknowledged Federal Support: **Y**

Publication Type: Journal Article Peer Reviewed: Y **Publication Status:** 1-Published
Journal: IEEE Transactions on Software Engineering
Publication Identifier Type: DOI Publication Identifier: 10.1109/TSE.2021.3106964
Volume: Issue: First Page #: 1
Date Submitted: 7/1/22 12:00AM Date Published: 8/24/21 4:00AM
Publication Location:
Article Title: Using Symbolic States to Infer Numerical Invariants
Authors: Thanhvu Nguyen, Kim Hao Nguyen, Matthew Dwyer
Keywords: Program Invariants , Numerical Invariants , Dynamic Analysis , Symbolic Execution , CounterExample Guided Refinement , Program Testing and Verification
Abstract: Automatically inferring invariant specifications has proven valuable in enabling a wide range of software verification and validation approaches over the past two decades. Recent approaches have shifted from using observation of concrete program states to exploiting symbolic encodings of sets of concrete program states in order to improve the quality of inferred invariants. In this paper, we demonstrate that working directly with symbolic states generated by symbolic execution approaches can improve invariant inference further. Our technique uses a counterexample-based algorithm that iteratively creates concrete states from symbolic states, infers candidate invariants from both concrete and symbolic states, and then validates or refutes candidate invariants using symbolic states. The refutation process serves both to eliminate spurious invariants and to drive the inference process to produce more precise invariants. This framework can be employed to infer complex invariants that captur
Distribution Statement: 1-Approved for public release; distribution is unlimited.
Acknowledged Federal Support: **Y**

Publication Type: Journal Article Peer Reviewed: Y **Publication Status:** 4-Under Review
Journal: ACM Transactions on Software Engineering and Methodology
Publication Identifier Type: Publication Identifier:
Volume: Issue: First Page #:
Date Submitted: 7/1/22 12:00AM Date Published:
Publication Location:
Article Title: Algorithm Selection for Software Verification using Graph Attention Networks
Authors: Will Leeson, Matthew Dwyer
Keywords: algorithm selection, graph attention networks, graph neural networks
Abstract: The field of software verification has produced a wide array of algorithmic techniques that can prove a variety of properties of a given program. It has been demonstrated that the performance of these techniques can vary up to 4 orders of magnitude on the same verification problem. Even for verification experts, it is difficult to decide which tool will perform best on a given problem. For general users, deciding the best tool for their verification problem is effectively impossible. In this work, we present \textsc{Graves}, a selection strategy based on graph neural networks (GNNs). \textsc{Graves} generates a graph representation of a program from which a GNN predicts a score for a verifier that indicates its performance on the program. We evaluate \textsc{Graves} on a set of 10 verification tools and over 8000 verification problems and find that it improves the state-of-the-art in verification algorithm selection by 11\%. We conjecture this is in part due to \textsc{Graves}' use o
Distribution Statement: 2-Distribution Limited to U.S. Government agencies only; report contains proprietary info
Acknowledged Federal Support: **Y**

CONFERENCE PAPERS:

Publication Type: Conference Paper or Presentation **Publication Status:** 1-Published
Conference Name: OOPSLA
Date Received: 01-Jul-2022 Conference Date: 15-Nov-2020 Date Published: 15-Nov-2020
Conference Location: Chicago, Illinois
Paper Title: DynamiTe: Dynamic Termination and Non-termination Proofs
Authors: Ton-Chanh Le, Timos Antonopoulos, Parisa Fathololumi, Eric Koskinen, ThanhVu Nguyen
Acknowledged Federal Support: **Y**

RPPR Final Report
as of 09-Aug-2022

Publication Type: Conference Paper or Presentation **Publication Status:** 2-Awaiting Publicat
Conference Name: ICSME
Date Received: 26-Aug-2020 Conference Date: 27-Sep-2020 Date Published: 27-Sep-2020
Conference Location: Adelaid, Australia
Paper Title: Using Symbolic Execution to Analyze Linux KBuild Makefiles
Authors: ThanhVu Nguyen and KimHao Nguyen
Acknowledged Federal Support: **Y**

Publication Type: Conference Paper or Presentation **Publication Status:** 0-Other
Conference Name: ICSME
Date Received: 01-Jul-2022 Conference Date: 27-Sep-2020 Date Published: 27-Sep-2020
Conference Location: Adelaide, Australia (virtual in 2020)
Paper Title: Debugging Declarative Models in Alloy
Authors: Guolong Zheng, Hamid Bagheri and Thanhvu Nguyen
Acknowledged Federal Support: **N**

Publication Type: Conference Paper or Presentation **Publication Status:** 1-Published
Conference Name: International Workshop on Software Security from Design to Deployment
Date Received: 01-Jul-2022 Conference Date: 11-Nov-2020 Date Published: 11-Nov-2020
Conference Location: Melbourne, Australia (virtual in 2020)
Paper Title: Using Dynamically Inferred Invariants to Analyze Program Runtime Complexity
Authors: Thanhvu Nguyen, Didier Ishimwe, Alexey Malyshev, Timos Antonopoulos, Quoc-Sang Phan
Acknowledged Federal Support: **Y**

Publication Type: Conference Paper or Presentation **Publication Status:** 1-Published
Conference Name: International Conference on Software Engineering
Date Received: 01-Jul-2022 Conference Date: 25-May-2021 Date Published: 25-May-2021
Conference Location: Madrid, Spain
Paper Title: GenTree Using Decision Trees to Learn Interactions for Configurable Software
Authors: KimHao Nguyen, ThanhVu Nguyen
Acknowledged Federal Support: **Y**

Publication Type: Conference Paper or Presentation **Publication Status:** 0-Other
Conference Name: International Conference on Software Engineering
Date Received: 01-Jul-2022 Conference Date: 25-May-2021 Date Published: 25-May-2021
Conference Location: Madrid, Spain
Paper Title: FLACK: Counterexample-Guided Fault Localization for Alloy Models
Authors: Guolong Zheng, ThanhVu Nguyen, Simón Gutiérrez, Germán Regis, Marcelo F. Frias, Nazareno Aguirr
Acknowledged Federal Support: **Y**

Publication Type: Conference Paper or Presentation **Publication Status:** 0-Other
Conference Name: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)
Date Received: 01-Jul-2022 Conference Date: 22-May-2021 Date Published: 22-May-2021
Conference Location: Madrid, ES
Paper Title: Bounded Exhaustive Search of Alloy Specification Repairs
Authors: Simón Gutiérrez Brida, Germán Regis, Guolong Zheng, Hamid Bagheri, ThanhVu Nguyen, Nazareno Ag
Acknowledged Federal Support: **Y**

RPPR Final Report
as of 09-Aug-2022

Publication Type: Conference Paper or Presentation **Publication Status:** 0-Other
Conference Name: OOPSLA
Date Received: 01-Jul-2022 Conference Date: 15-Oct-2021 Date Published: 15-Oct-2021
Conference Location: Chicago
Paper Title: Dynaplex: analyzing program complexity using dynamically inferred recurrence relations
Authors: Didier Ishimwe, KimHay Nguyen, ThanhVu Nguyen
Acknowledged Federal Support: **Y**

Publication Type: Conference Paper or Presentation **Publication Status:** 1-Published
Conference Name: TACAS
Date Received: 01-Jul-2022 Conference Date: 30-Mar-2022 Date Published: 30-Mar-2022
Conference Location: Munich
Paper Title: Graves-CPA: A Graph-Attention Verifier Selector
Authors: Will Leeson, Matthew Dwyer
Acknowledged Federal Support: **Y**

WEBSITES:

URL: https://bitbucket.org/mgerrard/alpaca
Date Received:
Title: ALPACA : A Large Portfolio-based Alternating Conditional Analysis
Description:
URL: https://github.com/dynaroars/dig
Date Received: 01-Jul-2022
Title: DIG: Dynamic Invariant Generation
Description: Implementation of state-of-the-art dynamic invariant inference techniques

Partners

,

I certify that the information in the report is complete and accurate:
Signature: Matthew Dwyer
Signature Date: 7/1/22 2:28PM

Final Report for ARO Project “Predictive Failure Avoidance”

Participants

The ExtraNet web-site does not permit entry of the total number of months supported on the project. We include that information here. PI Dwyer and co-PI Nguyen were each supported for 3 months of effort across the project. Graduate Student Leeson was supported for 36 months. Graduate Student Zhang was supported for 18 months. Graduate Student Ishimwe was supported for 6 months.