Neural Network Verification for Robustness of Malware Classifiers

IFIP Working Group 10.4 Workshop
June 30, 2024

Taylor T. Johnson, PhD, PE, Associate Professor & A. James and Alice B. Clark Foundation Chancellor Faculty Fellow

VeriVITAL - the **Veri**fication and **V**alidation for **I**ntelligent & **T**rustworthy **A**utonomy **L**aboratory

Institute for Software Integrated Systems

Departments of Computer Science & Electrical and Computer Engineering

Vanderbilt University

http://www.taylortjohnson.com/
taylor.johnson@vanderbilt.edu

VANDERBILT UNIVERSITY

VeriVITAL Members & Alumni

Current PhD Students & Postdocs


Judy Nguyen


Anne Tumlin
2024 DOE CSGF


Samuel Sasaki


Preston Robinette
2021 NDSEG


Serena Serbinowska

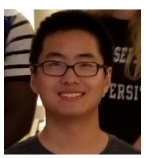

Dr. Diego Manzanas Lopez


Dr. Tianshu Bao

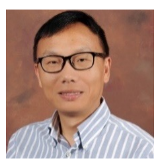

Dr. Neelanjana Pal
MathWorks

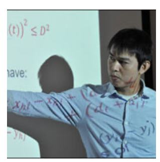
Postdoc / PhD / Research Scientist Alumni


Dr. Nate Hamilton
2019 NDSEG
Parallax Research


Dr. Xiaodong Yang
Visa Research

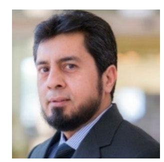

Dr. Patrick Musau
Google


Prof. Weiming Xiang
Augusta University
2022 NSF CAREER


Prof. Hoang-Dung Tran
U Nebraska Lincoln
2021 IEEE TCCPS Outstanding Dissertation


Prof. Joel Rosenfeld
USF
2021 AFOSR YIP


Prof. Luan Nguyen
U Dayton
2023 NSF CRII


Prof. Omar Beg
U Texas PB
2019 UT System Rising STARs


Dr. Shafiul Chowdhury
Meta, ML Senior Research Scientist


Prof. Khaza Hoque
U Missouri


Dr. Andrew Sogokon
Lancaster

MSc Thesis / Undergrad Researcher Alumni: at Google, Meta, Microsoft, Amazon, Qualcomm, Rivian, …

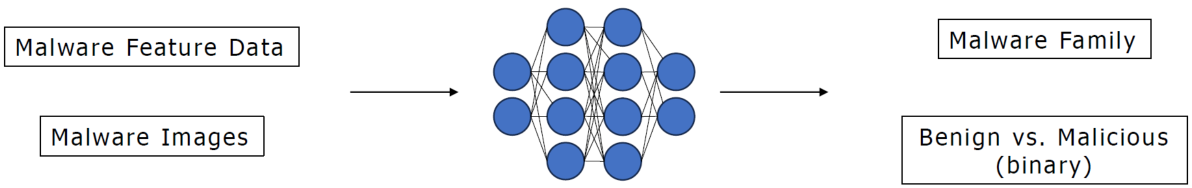
NSA SoS: Improving Malware Classifiers with Plausible Novel Samples


Kevin Leach

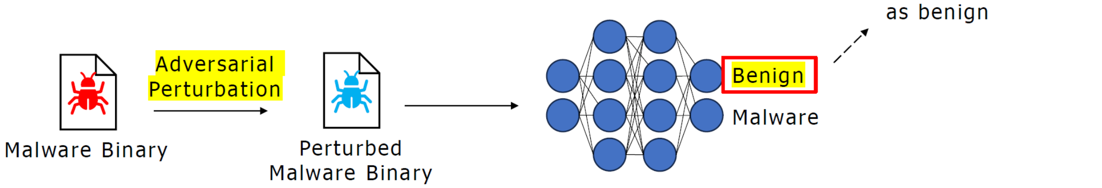

Preston Robinette

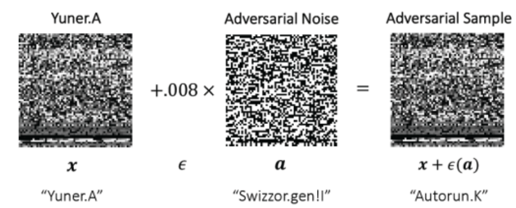
https://github.com/pkrobinette/verify_malware

- Neural Networks are a popular means of classification:
  - Benign vs. malicious
  - Malware family



- Adversary can *perturb* input sample to cause **incorrect classification**





Metric	Model	Tool	Epsilon (ϵ) 1/255	2/255	3/255
CRA (%)	linear-25	NNV	85	83	79
		nnenum	90	86	82
	4-25	NNV	89	76	62
		nnenum	94	80	66
	16-25	NNV	88	82	67
		nnenum	90	86	64
Avg. Time (s)	linear-25	NNV	0.84	0.85	0.85
		nnenum	3.60	3.63	3.69
	4-25	NNV	17.75	41.66	82.18
		nnenum	11.59	10.80	11.13
	16-25	NNV	85.00	210.00	710.25
		nnenum	38.66	44.16	43.43

[Robinette et al, "Case Study: Neural Network Malware Detection Verification for Feature and Image Datasets," Formalise'24]
[Robinette et al, "Benchmark: Neural Network Malware Classification," AISoLA'23]

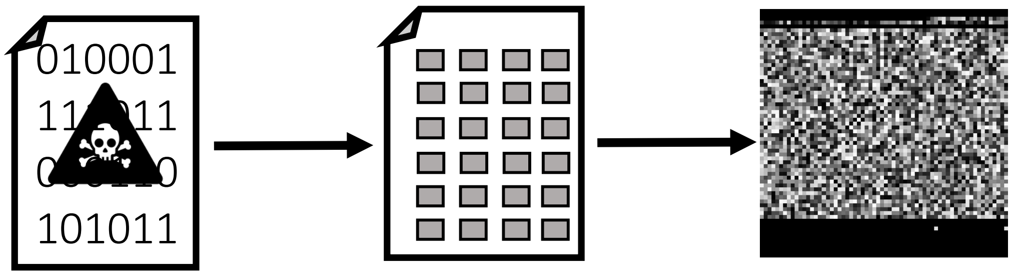
Feature Datasets

BODMAS Dataset

- Composed of “features” extracted from collected samples
- Static and dynamic features
  - **Static:** file properties, binary content, API calls, and embedded resources
  - **Dynamic:** runtime behavior (changes made to files, registries, and the system memory), system interactions, and state changes over time
- Features consist of **different data types** and ranges within each datatype

Image Datasets

Malimg Dataset



Malware Binary 8-bit Vector Malware Image

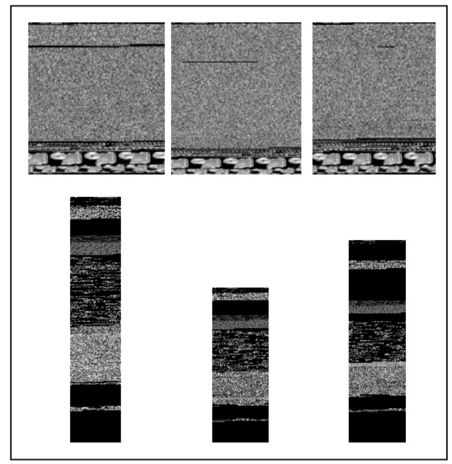
VANDERBILT UNIVERSITY

Image Datasets

Malimg Dataset

Tab. 1: Image Width for Various File Sizes

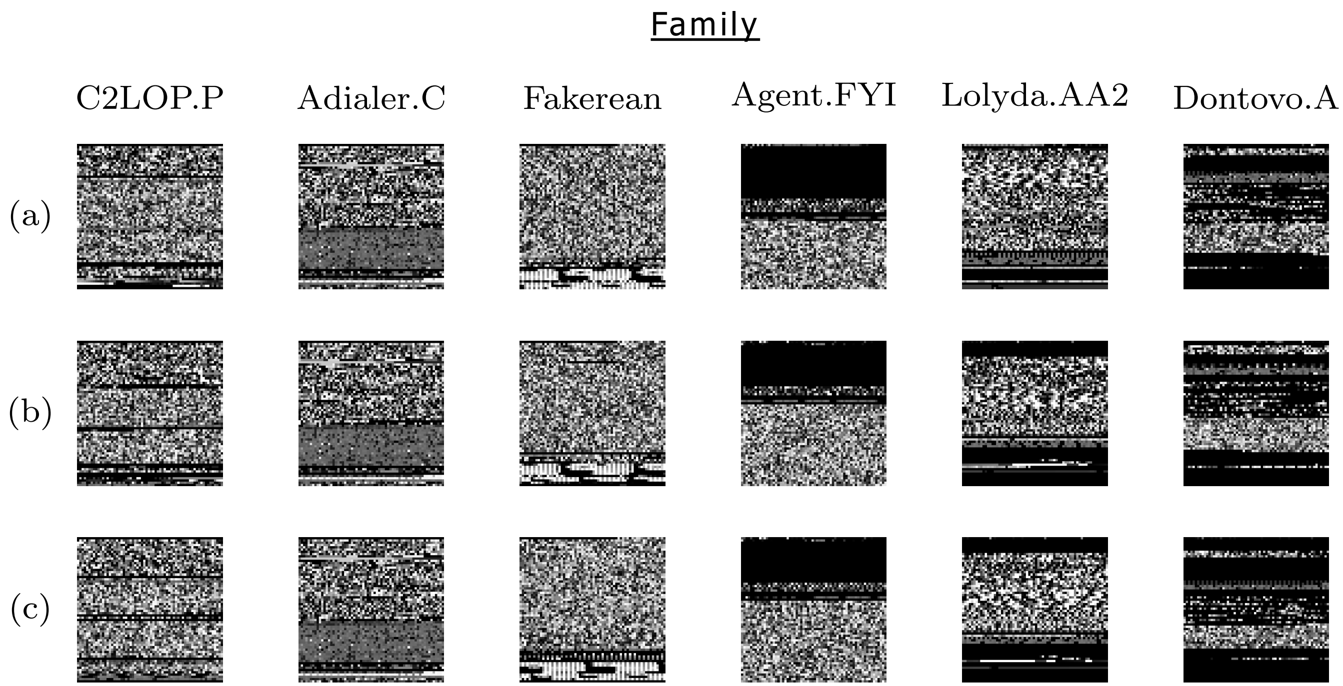
File Size Range	Image Width
<10 kB	32
10 kB – 30 kB	64
30 kB – 60 kB	128
60 kB – 100 kB	256
100 kB – 200 kB	384
200 kB – 500 kB	512
500 kB – 1000 kB	768
>1000 kB	1024



Fakerean

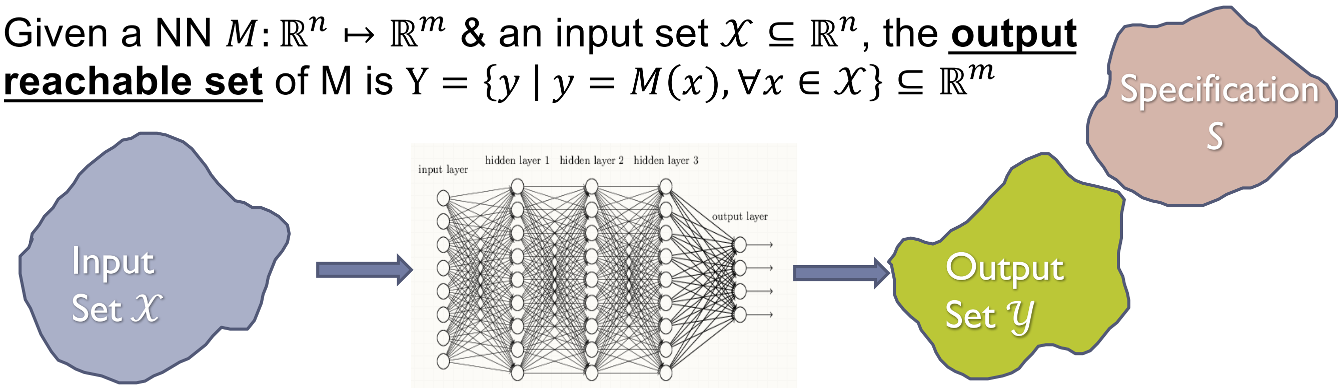
Dontovo.A

Image Datasets



Neural Network Verification with Reachability

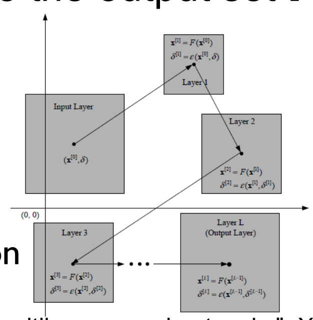
- Given a NN $M: \mathbb{R}^n \mapsto \mathbb{R}^m$ & an input set $\mathcal{X} \subseteq \mathbb{R}^n$, the **output reachable set** of M is $\mathrm{Y} = \{y \mid y = M(x), \forall x \in \mathcal{X}\} \subseteq \mathbb{R}^m$



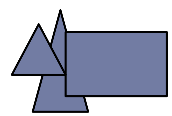
- Computationally: Given a NN M, a convex initial set of inputs I represented as a polytope poly($\mathcal{X}$), compute the output set $Y = M(I)$ of the network

$\boxed{\mathrm{I=poly}(\mathcal{X})}$

$I = \{x \in \mathbb{R}^n | Ax \leq B\}$



Layer-by-Layer Propagation
of Polytopes



Compute: $Y = M(I)$
Verify: $M(I) \cap S = \emptyset$?

["Output reachable set estimation and verification for multilayer neural networks", Xiang, Tran, Johnson, TNNLS'18]

Neural Network Reachability Illustrative Example

Weiming Xiang

Given a NN $M: \mathbb{R}^n \mapsto \mathbb{R}^m$ & an input set $\mathcal{X} \subseteq \mathbb{R}^n$, the **output reachable set** of M is $\mathrm{Y} = \{y \mid y = M(x), \forall x \in \mathcal{X}\} \subseteq \mathbb{R}^m$

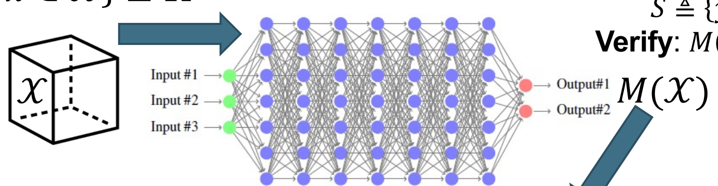
M: simple feedforward NN with 3 inputs, 2 outputs, 7 hidden layers of 7 neurons each, ReLU activations; $M: \mathbb{R}^3 \to \mathbb{R}^2$

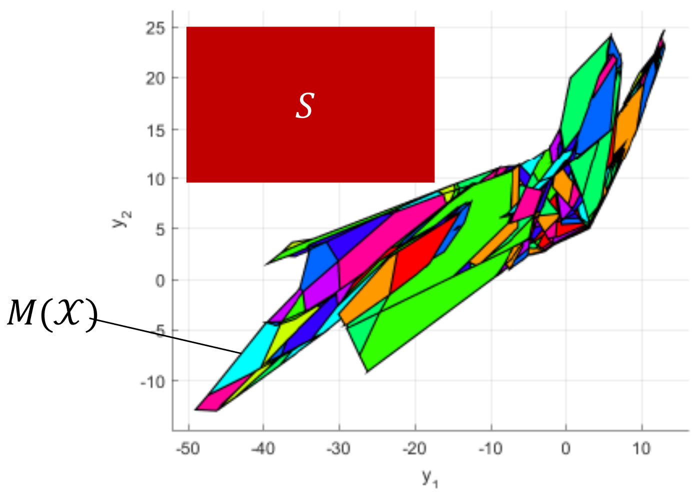
Input set: $\mathcal{X} \triangleq \{x \in \mathbb{R}^3 \mid \|x\|_\infty \leq 1\}$

Specification:

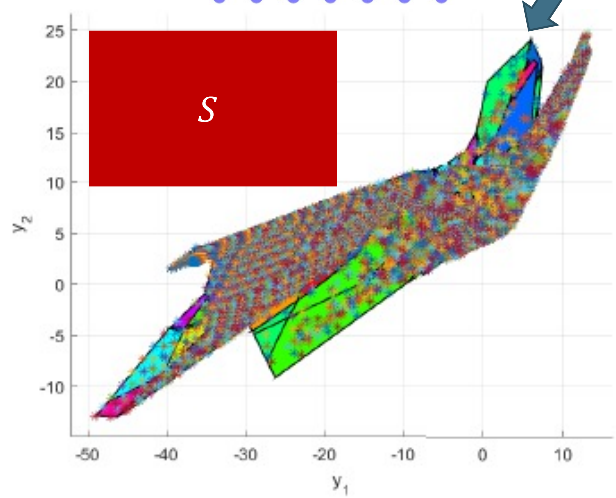
$S \triangleq \{y \in \mathbb{R}^2 \mid -50 \leq y_1 \leq -20 \land 10 \leq y_2 \leq 25\}$

Verify: $M(\mathcal{X}) \cap S = \emptyset$?





Output reachable set $Y = M(\mathcal{X})$: union of 1250 polytopes, shown in different colors



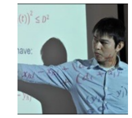
8000 randomly generated outputs (evaluating M on points, e.g., $M(x)$ for 8000 points $x \in \mathcal{X}$)

Scalability Challenge: for ReLU activations, this problem is NP-complete

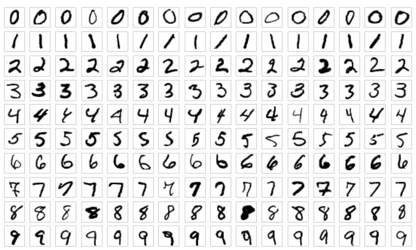
Intuition: number of polytopes may grow exponentially in number of ReLUs due to case splitting

This is exact $M(\mathcal{X})$, often overapproximate $\widehat{M}(\mathcal{X}) \supseteq M(\mathcal{X})$ for scalability, but then need to worry about precision (suppose convex hull for this example)

MNIST Robustness Verification: Comparison of Set Representations



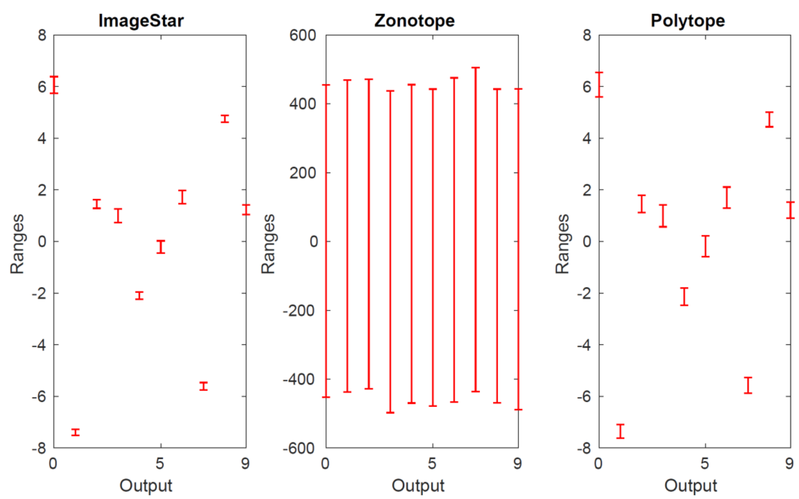
Hoang-Dung Tran



http://yann.lecun.com/exdb/mnist/

If $M(x) = M(x')$ for all $x' \in \{x' \in \mathbb{R}^n : \|x - x'\|_p \leq \epsilon\}$, then M is **locally adversarially robust up-to ϵ** about x

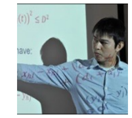
Here x is an image  from MNIST, and this says all nearby x' have same class as x



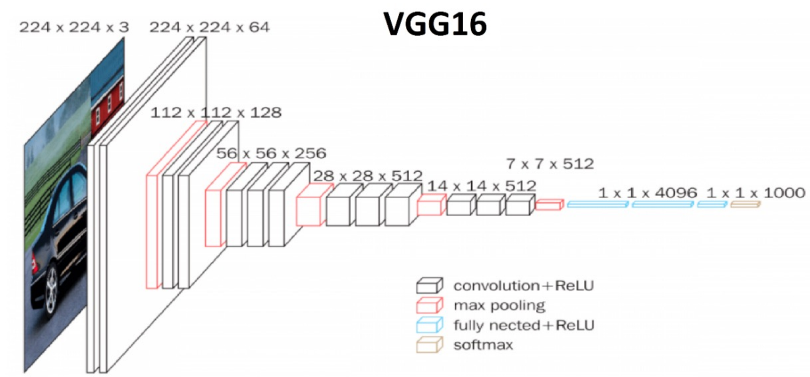
- MNIST classifier is a function from images to classes, $M: \mathbb{R}^{28\times 28} \mapsto \{0, \ldots, 9\}$
- Input: $\mathbb{R}^{28\times 28}$; input set: a convex subset $\mathcal{X} \subseteq \mathbb{R}^{28\times 28}$
- Output prior to softmax/argmax: $\mathbb{R}^{10}$; output set: shape in $\mathbb{R}^{10}$
- Final output: take argmax over these 10 dimensions, this is the identified class
  - If min of ground truth class in $M(\mathcal{X})$ (say a 0 for this example) > max of all other classes, then locally adversarially robust up-to perturbation ϵ about data sample x (input image); can write this in VNN-LIB and compatible with intersection checking approach
- ImageStar: efficient and accurate set representation developed for NNV, extension of star sets for images
- Do this analysis across data set to get **certified robust accuracy (CRA)**, which is <= accuracy

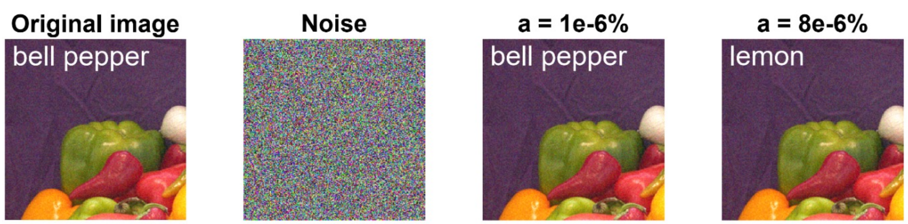
[Tran et al, “Verification of Deep Convolutional Neural Networks Using ImageStars,” **CAV’20**]

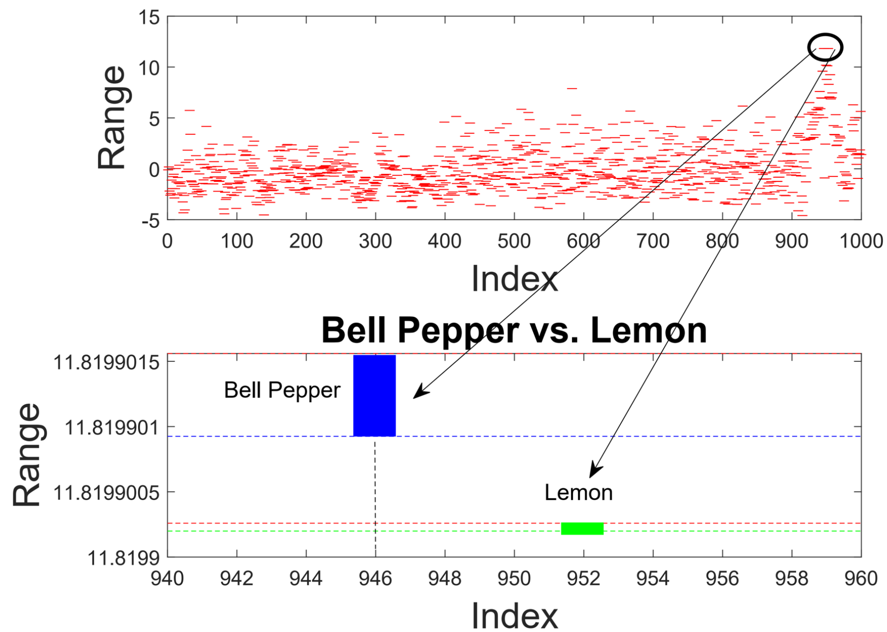
VGG16 Robustness Verification Example



Hoang-Dung Tran







Disturbed images = Original image + *a* * Noise; note a is a set. Essentially upper/lower bound of noise

Is VGG16 robust to an FGSM attack for $a \leq 2\times10^{-8}$?

[Tran et al, “Verification of Deep Convolutional Neural Networks Using ImageStars,” **CAV’20**]

Status of Neural Network Verification

- Significant progress in scalability (~1 order of magnitude improvement in size of network [# neurons] annually since ~2017): **up to hundreds of millions of neurons, see VNN-COMP reports**
- Ongoing challenges
  - Specifications, Benchmarks, VNN-LIB/ONNX, …
  - Scalability: size of network, but also complexity of specification (“volume” of input set), …
  - Balancing precision and scalability: CEGAR, CEGIS, abstraction (INN), …
  - Architectural support (layer types, …)
  - Learning/Design-for-verification: have seen newcomers to area try to apply tools blindly, often won’t work, need to collaborate with teams developing verification approaches
    - Representative design guidance: try to mostly use ReLUs, minimize sequence of ReLU layers; many tools can’t work with other activations and scalability much worse (for max pooling, tanh/sigmoid, etc.)

https://doi.org/10.1007/s10009-023-00703-4
https://sites.google.com/view/vnn2024 and https://www.vnnlib.org/

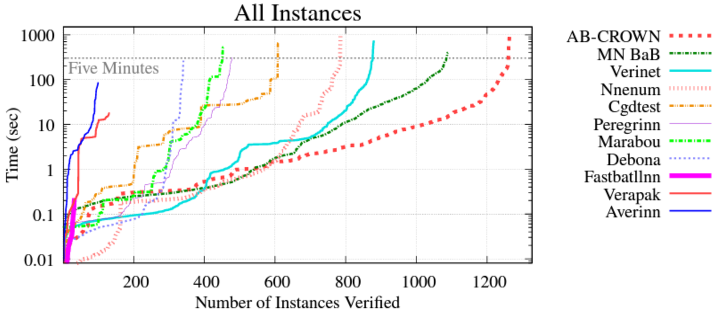
Table 5 Comparison across years

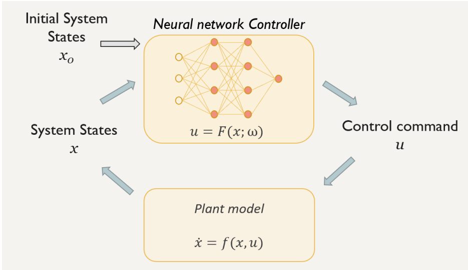
	2020	2021	2022
Tools registered	N/A	15	18
Tools submitted	8	13	11
Benchmarks submitted	5	8 (+1 unscored)	12 (+1 unscored)
Max. network depth	8	18	27
Max. network parameters	855,600	42,059,431 (sparse)	138,356,520
Activation functions	ReLU, tanh, sigmoid	ReLU, sigmoid, MaxPool, AveragePool	ReLU, sigmoid, MaxPool
Layer types	Fully Connected, Conv	Fully Connected, Conv, Residual	Fully Connected, Conv, Residual, BatchNorm
Applications	Image Recognition, Control	Image Recognition, Control, Database Indexing	Image Recognition, Control, Database Indexing, Cardinality Estimation
Mean #benchmarks/tool	3.0 (min 2, max 5)	5.5 (min 1, max 9)	7.3 (min 1, max 13)

NN and NNCS Verification Related Work

- NN verification
  - Approaches
    - SMT, MILP, Reachability, Abstract interpretation, …
  - Tools
    - α,β-CROWN, MN BaB, Verinet, NNV, nnenum, cdgtest, Peregrinnm Marabou, Debona, Fastballnn, Reluplex, DLV, ReluVal, ERAN, Venus, OVAL, DNNF, RPM, NV.jl, MIPVerify, Verapak, Averinn, Veritex, …
  - Competition
    - VNN-COMP (NNV participant 2020, 2021, 2023, 2024)
    - https://sites.google.com/view/vnn2024
  - Tech transfer: several startups, Matlab toolbox, …
    - https://safeintelligence.ai/, https://latticeflow.ai/, https://www.mathworks.com/products/deep-learning-verification-library.html
- Neural Network Control System (NNCS) verification
  - ARCH-COMP Friendly Competition
    - ARCH-COMP AINNCS (NNV participant 2019, 2020, 2021, 2022, 2023, 2024): https://cps-vo.org/group/ARCH/FriendlyCompetition
  - Tools
    - CORA, JuliaReach, Verisig, ReachNN*, NNV, POLAR, OVERT, VenMAS, Sherlock, RINO, NFL_veripy, DeepNNC, SMC, AutomatedReach, GoTube, immrax, …

[MN Müller et al, VNN-COMP 2022]





NNCS

(P.S. Apologies if we missed your tool, please come talk to us after the talk and we'll fix it for the next one)

5th International Competition on Verification of Neural Networks (**VNN-COMP’24**), co-located with International Conference on Computer-Aided Verification (CAV’24) in new **Symposium on AI Verification (SAIV’24)**

Stanley Bak | Christopher Brix | Taylor Johnson | Changliu Liu | Mark Müller

https://sites.google.com/view/vnn2024

https://www.aiverification.org/

FLoC 2022

FoMLAS

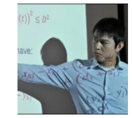
2023 report: https://arxiv.org/abs/2312.16760

2020-2022 comparative report: https://arxiv.org/abs/2301.05815

2022 report: https://arxiv.org/abs/2212.10376

2021 report: https://arxiv.org/abs/2109.00498

Neural Network Verification (NNV) Software Tool

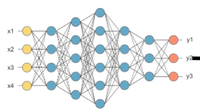


Hoang-Dung Tran

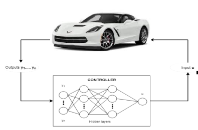
*The Neural Network Verification (**NNV**) Tool*

https://github.com/verivital/nnv

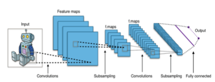
$M \triangleq$



Feedforward Neural Networks (FFNN)



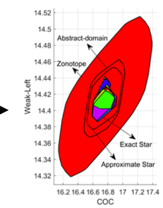
Neural Network Control Systems (NNCS)



Convolutional Neural Networks (CNN)

Reachability Solvers

Visualizer

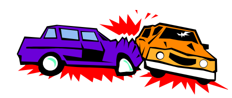


Verifier
$M \vDash S$?

No: bug

Yes: proof

$S \triangleq \neg$

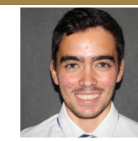


$S \triangleq \neg$

[Eykholt et al, CVPR 2018]

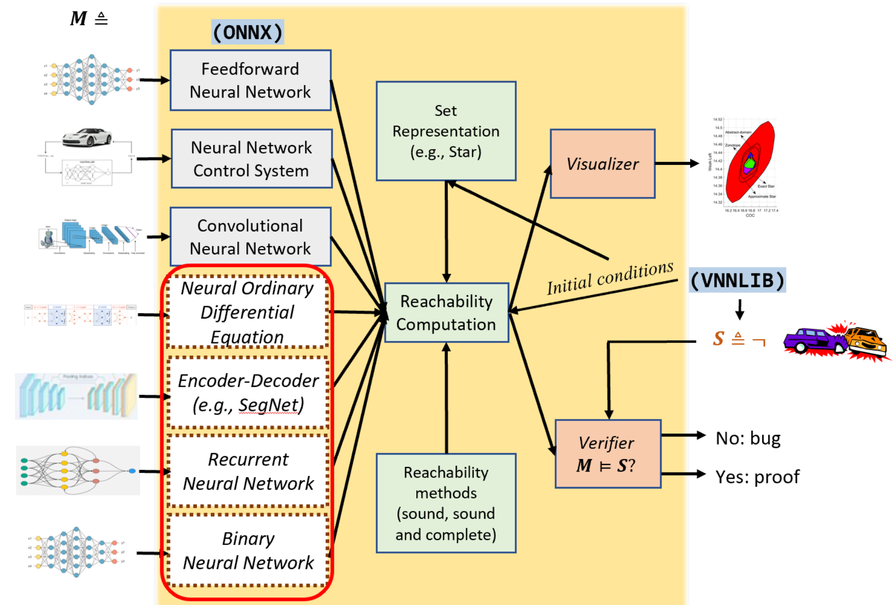
[Xiang et al, "Output Reachable Set Estimation and Verification for Multi-Layer Neural Networks", **TNNLS'18**]
[Tran et al, "Star-Based Reachability Analysis for Deep Neural Networks", **FM'19**]
[Tran et al, "Safety Verification of Cyber-Physical Systems with Reinforcement Learning Control", **EMSOFT'19**]
[Tran et al, "NNV: The Neural Network Verification Tool for Deep Neural Networks and Learning-Enabled Cyber-Physical Systems", **CAV'20**]
[Tran et al, "Verification of Deep Convolutional Neural Network using ImageStars", **CAV'20**]
[Bak et al, "Improved Geometric Path Enumeration for Verifying ReLU Neural Networks", **CAV'20**]
[Xiang et al, "Reachable Set Estimation for Neural Network Control Systems: A Simulation-Guided Approach", **TNNLS'20**]
[Tran et al, "Robustness Verification of Semantic Segmentation Neural Networks using Relaxed Reachability", **CAV'21**]
[Lopez et al, "Evaluation of Neural Network Verification Methods for Air to Air Collision Avoidance", **AIAA JAT'22**]
[Lopez et al, "Reachability Analysis of a General Class of Neural Ordinary Differential Equations", **FORMATS'22**]
[Lopez et al, "NNV 2.0: The Neural Network Verification Tool", **CAV'23**]

Neural Network Verification (NNV) Software Tool: New Version 2.0



Diego Manzanas Lopez

- Significant updates to NNV: version 2.0 presented at CAV’23
- Upcoming tutorial at DSN’24, recent tutorials at EMSOFT’23 and IAVVC’23
  - https://github.com/verivital/nnv/tree/master/code/nnv/examples/Tutorial
- Participation in VNN-COMP’24 and ARCH-COMP’24 AINNCS category
  - https://sites.google.com/view/vnn2024
  - https://github.com/verivital/ARCH-COMP2024
- Organization of AISoLA’24 Verification for Neuro-Symbolic Artificial Intelligence (VNSAI) track
  - https://2024-isola.isola-conference.org/aisola-tracks/

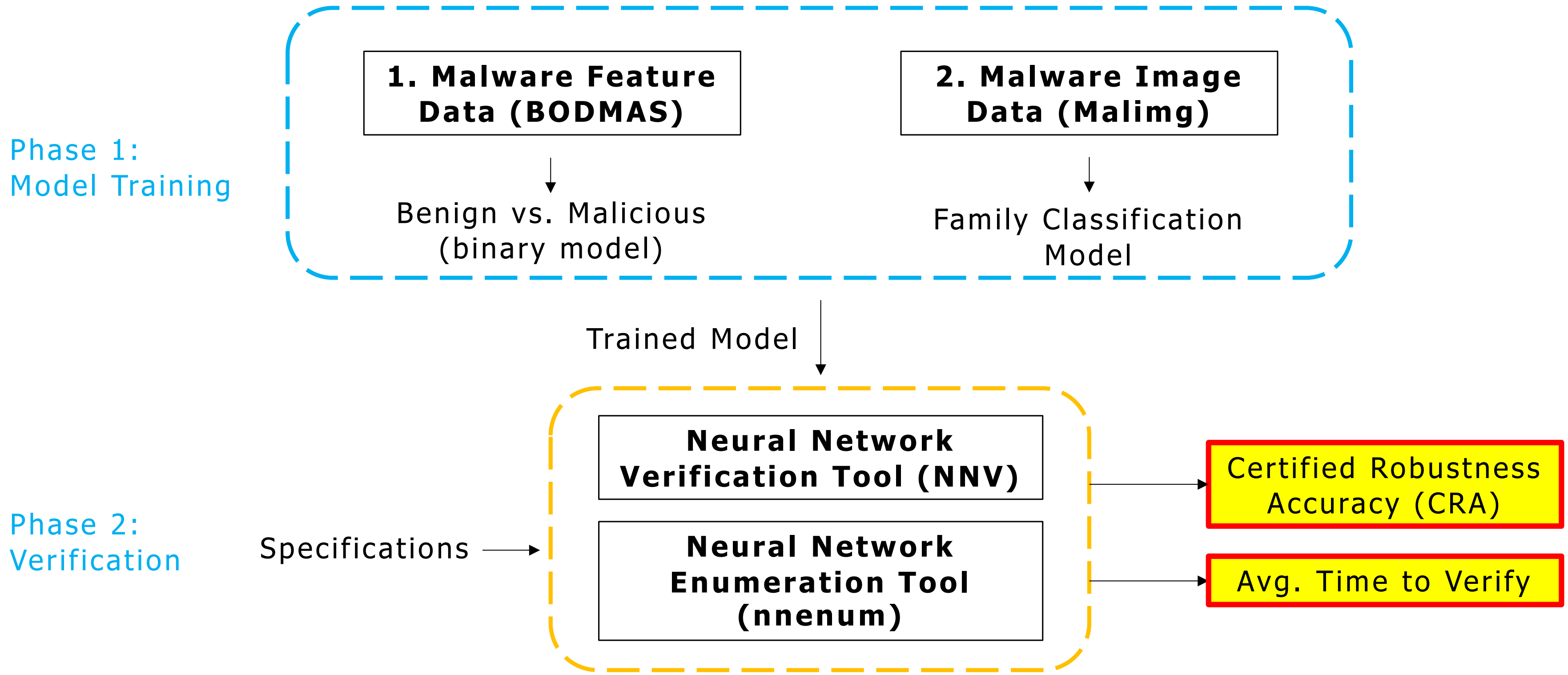


[Manzanas Lopez et al, "Verification of Neural Network Compression of ACAS Xu Lookup Tables with Star Set Reachability", **AIAA'21**]
[Xiang et al, “Reachable Set Estimation for Neural Network Control Systems: A Simulation-Guided Approach”, **TNNLS’21**]
[Tran et al, “Robustness Verification of Semantic Segmentation Neural Networks using Relaxed Reachability”, **CAV’21**]
[Tran et al, "Verification of Piecewise Deep Neural Networks: A Star Set Approach with Zonotope Pre-filter", **FAOC'21**]
[Manzanas Lopez et al, "Reachability Analysis of a General Class of Neural Ordinary Differential Equations", **FORMATS'22**]
[Manzanas Lopez et al, "Evaluation of Neural Network Verification Methods for Air-to-Air Collision Avoidance", **JAT'22**]
[Tran et al, "Verification of Recurrent Neural Networks using Star Reachability", **HSCC'23**]
[Ivashchenko et al, "Verifying Binary Neural Networks on Continuous Input Space using Star Reachability", **FormaliSE’23**]
*[Manzanas Lopez et al, "NNV 2.0: The Neural Network Verification Tool", **CAV’23**]*
[Robinette et al, “Case Study: Neural Network Malware Detection Verification for Feature and Image Datasets”, **FormaliSE’24**]

https://github.com/verivital/nnv

NNV 2.0

Malware Robustness Case Study



VANDERBILT UNIVERSITY

NNV: https://github.com/verivital/nnv
nnenum: https://github.com/stanleybak/nnenum

Case Study: Metrics

$$\text{Certified Robustness Accuracy (CRA)} = \frac{\text{\# of samples certified robust}}{\text{Total \# of samples}}$$

$$\text{Avg. Time to Verify} = \frac{\text{Total wall time to verify for all samples}}{\text{Total \# of samples}}$$

Results: Model Performance

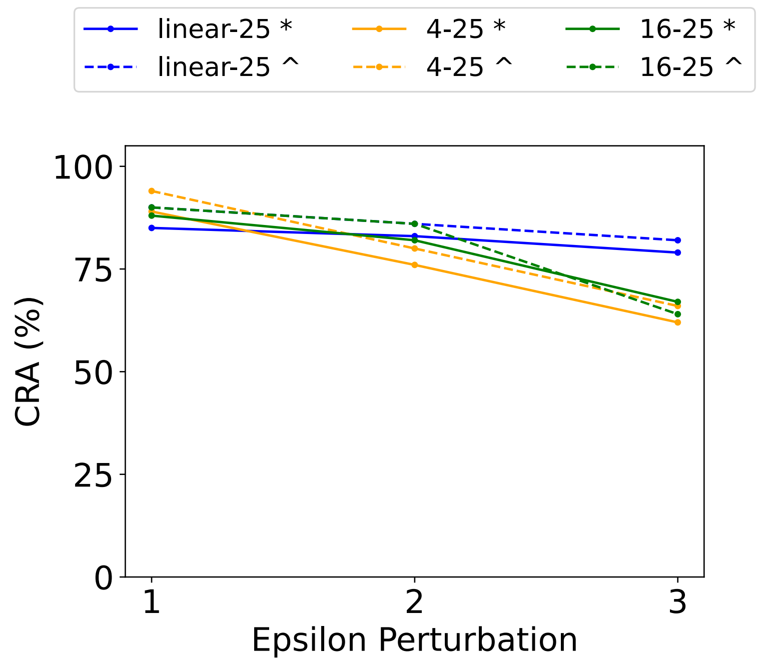
Dataset	Model	Accuracy	Precision	Recall	F1
BODMAS (26,887)	none-2	0.99	0.98	0.99	0.99
	4-2	0.99	0.99	0.99	0.99
	16-2	0.99	0.99	0.99	0.99
Malimg (935)	linear-25	0.99	0.98	0.97	0.97
	4-25	0.98	0.97	0.96	0.97
	16-25	0.99	0.97	0.96	0.97

Models achieve high performance for each dataset type.

Results: Image Dataset, CRA

- As the perturbation size increases, the models decrease in CRA
- Small models outperform larger models with high epsilon values
- NNV and nnenum have similar CRA evaluation performance for each epsilon value

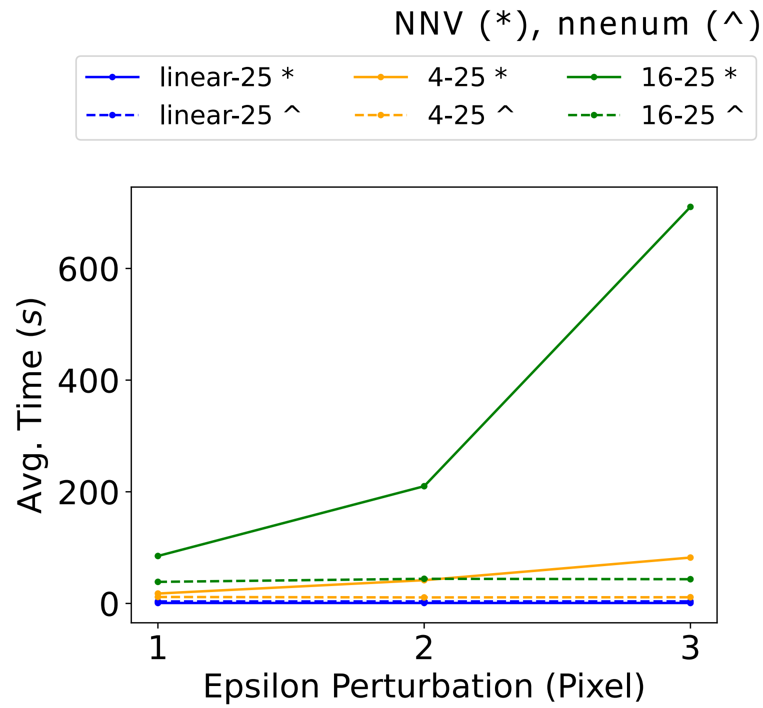
NNV (*), nnenum (^)



(125 Samples)

Results: Image Dataset, Time to Verify

- The larger the model, the more time typically required for each of the verification steps following falsification
  - Calculation of reachable set
- nnenum takes less time to verify than NNV for larger models



(125 Samples)

Results: Image Dataset

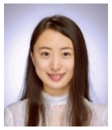
Metric	Model	Tool	Epsilon (ϵ) 1/255	2/255	3/255
CRA (%)	linear-25	NNV	85	83	79
		nnenum	90	86	82
	4-25	NNV	89	76	62
		nnenum	94	80	66
	16-25	NNV	88	82	67
		nnenum	90	86	64
Avg. Time (s)	linear-25	NNV	0.84	0.85	0.85
		nnenum	3.60	3.63	3.69
	4-25	NNV	17.75	41.66	82.18
		nnenum	11.59	10.80	11.13
	16-25	NNV	85.00	210.00	710.25
		nnenum	38.66	44.16	43.43

Summary

- Neural network verification is emerging approach for establishing properties of trained models, with significant scalability progress
- Shown snippets, particularly for evaluation of robustness through certified robust accuracy (CRA) for some malware classifiers
  - Challenges: samples in perturbed set under L-infinity norm may not correspond to valid binaries (but some may, and still an attack vector if adversary knows these types of classifier used), working toward other types of perturbations that preserve executability, semantics, etc.
  - Working toward coverage evaluation of input space
- Overall status, related work, etc.: look at VNN-COMP reports
- Major open challenge in field: specification
  - Domain specific approaches necessary

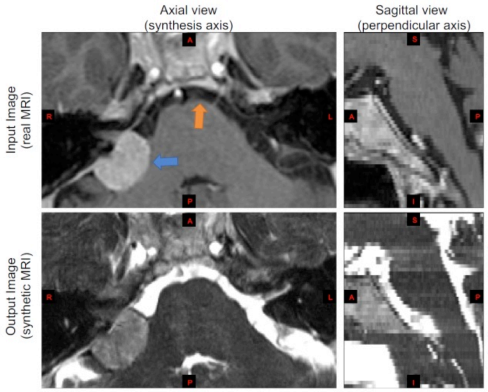
NSF FMitF: Track I: Generative Neural Network Verification in Medical Imaging Analysis

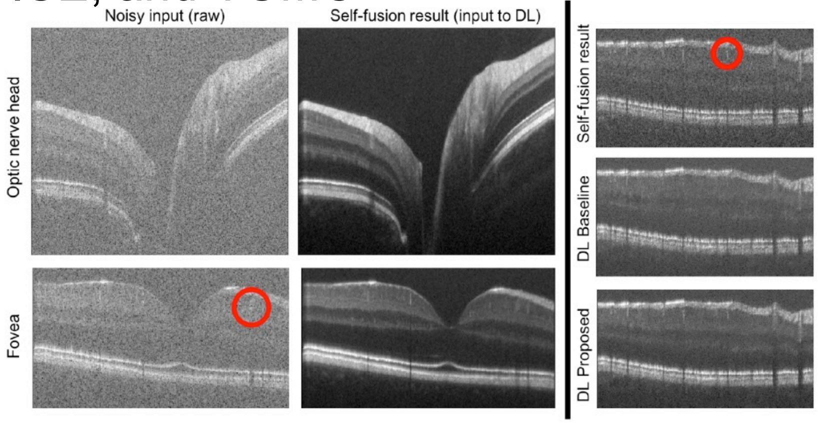

Ipek Oguz


Meiyi Ma

- DNNs, GANs, ... increasingly used to process medical data, including images (segmentation, denoising, synthesis, image reconstruction, …)
  - Major concerns about introduction of artifacts, etc. with generative models; less concerns about adversaries, but also to a degree
  - Project goals: develop ways to write specifications for generative models, define/scale verification for segmentation and image synthesis
- Collaboration between ISIS, VISE, and VUMC


Francesca Bagnato






Kenny Tao

Vanderbilt Institute for Surgery & Engineering (VISE): https://www.vanderbilt.edu/vise/

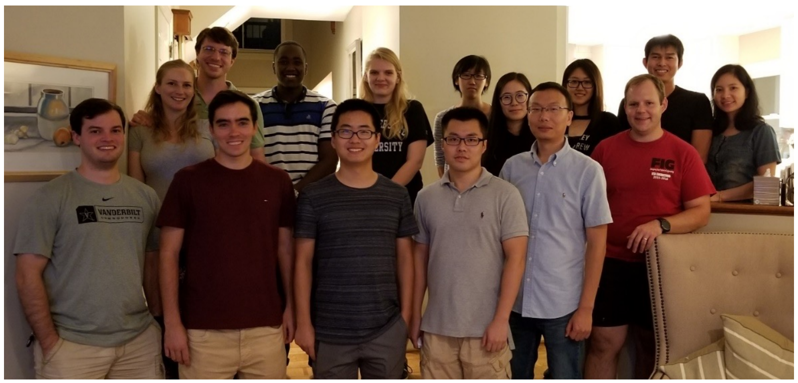
Verification for Neuro-Symbolic Artificial Intelligence (VNSAI) Track at ISoLA/AISoLA'24 in Crete, Greece

- Co-organize VNSAI track with Daniel Neider
- Please talk with me or email if interested to visit Crete ~Oct. 30-Nov. 3, 2024! taylor.johnson@vanderbilt.edu
- On-site LNCS proceedings deadline: June 28, 2024
- Invited talks: can publish in post-proceedings (LNCS / STTT), deadline for abstracts is July 29, post-proceedings paper deadline ~Jan. 2025

https://aisola.org/
https://2024-isola.isola-conference.org/aisola-tracks/
https://equinocs.springernature.com/service/vnsai

Thank You: Questions?

taylor.johnson@vanderbilt.edu
http://www.verivital.com/
Twitter: @taylorjohnson @verivital



VANDERBILT UNIVERSITY®