Verification of ACAS-Xu benchmark using Alpha Beta Crown and Nnenum

Amalia Duma Diana Răzvan Maciovan Alexandru Luca Mitroi
Vlad Petcu Vlad Țîrcomnicu

West University of Timișoara
Faculty of Mathematics and Informatics
Master Study Program: Software Engineering

Scientific Coordinator: Mădălina Erașcu

Friday 26th January, 2024

Motivation and Problem Specification

The ACAS-Xu system is designed to enhance airspace safety by providing resolution advisories. We aimed to verify the reliability of ACAS-Xu neural networks using verification tools, namely Alpha-Beta-CROWN and Nnenum.

Dataset Description

The ACAS-Xu dataset contains inputs representing sensor readings and advisories for unmanned aircraft. We then mix the time until loss of vertical separation (τ) and the previous advisory (a_{prev}) to get five outputs: Clear-of-Conflict (COC), weak/strong right, weak/strong left.

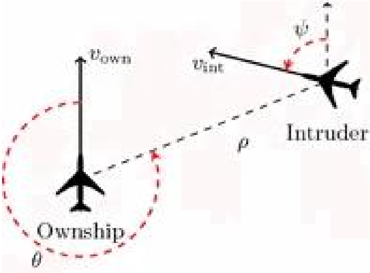

Figure: Geometry of ACAS Xu horizontal logic table [1]

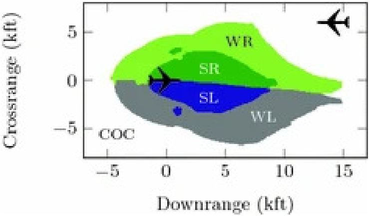

Figure: Advisories for a head-on encounter with a_{prev} = Clear of Conflict and τ = os [1]

Tools Used and Challenges

Alpha-Beta-CROWN

A neural network verifier based on an efficient linear bound propagation framework.

- Prerequisites: Conda

Setting up the tool involved a two-step process: installing Miniconda, which was straightforward, and then the tool itself, which presented complications.

The tool's installation required resolving errors related to the conda environment configuration and manually retrieving essential files due to permission issues.

Tools Used and Challenges

Nnenum

A neural network verifier that relies on multiple levels of abstraction to achieve high performance verification of ReLU networks without sacrificing completeness.

- Prerequisites: Docker

Following the steps outlined in the GitHub repository's "Getting started" section, including cloning the repository and executing two commands in the docker file, ensured a straightforward installation without any errors or unexpected behavior.

Experimental Results

The verification process involved evaluating 10 properties on 45 DNNs. The first four properties were assessed across all networks, while the rest were checked on a single network, following the original authors' approach. A consistent timeout of 116 seconds was applied to each property evaluation.

The final verified accuracy stands at approximately 74.73%.

Tool	Verified	Falsified	Penalty	Score	Avg. time	Max time
alpha-beta-CROWN	139	47	0	1860	2.58	69.08
nnenum	139	47	0	1860	2.3	61.58

Table: Benchmark 2023-ACAS-Xu

Experimental Results

ONNX	VNN-LIB	Result	Time (seconds)
onnx/ACASXU_run2a_1_1_batch_2000.onnx	vnnlib/prop_1.vnnlib	unsat	5.5214
onnx/ACASXU_run2a_1_9_batch_2000.onnx	vnnlib/prop_7.vnnlib	sat	0.355

Table: Example of results for Alpha-Beta-CROWN

ONNX	VNN-LIB	Result	Time (seconds)
onnx/ACASXU_run2a_3_4_batch_2000.onnx	vnnlib/prop_1.vnnlib	holds	1.46
onnx/ACASXU_run2a_1_9_batch_2000.onnx	vnnlib/prop_3.vnnlib	violated	0.93

Table: Example of results for Nnenum

Discussion and Conclusions

Our investigation shows the importance of formal verification in ensuring the safety of systems affecting real life scenarios like ACAS-Xu.

This study aimed to replicate results from VNN-COMP 2023 using alpha-beta-CROWN and nnenum as the neural network verifiers. While the tools produced identical results, alpha-beta-CROWN stood out by the absence of the timeout penalty that was applied in the competition

The verification process maintained consistency with competition parameters and utilized files sourced directly from the official GitHub repository.

References

Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer.
Reluplex: An efficient SMT solver for verifying deep neural networks.
In *Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I 30*, pages 97–117. Springer, 2017.