Verifying Systems with Neural Networks

Project Final Report

Abhishri Ajit Medewar (MSCS), Fall 2021- Semester 1st

ABSTRACT

Category: Verification

The project involves comparing the verification performance of two different tools on the Airborne Collision Avoidance System for Unmanned Aircraft ACAS Xu. It addresses the following questions:

1. How fast the tool can verify the same problem?
2. If some random errors are introduced to ACAS Xu, what do the tools report?

The use of machine learning to perform complex tasks is growing enormously. The problem with using machine learning to accomplish such tasks is that the machine learning-based model's execution is mostly opaque. There is an urgent need to create approaches that can explain network behavior while also providing assurance regarding the network's output.

Two such tools which are used for neural network verification are Marabou and ReluVal. The report explains the idea behind each tool and presents the experimental result of Marabou and ReluVal for ACAS Xu.

Work Performed

Tools Used

Two tools will be used for comparing the verification performance on the Airborne Collision Avoidance System for Unmanned Aircraft ACAS Xu.
Below are the list of tools:

- Marabou
- ReluVal

Neural Network Verification- Basic Idea

The verification of the neural network involves two things:
1. Proving that a property ϕ holds true for the given neural network system (S).
2. Provide a counter example if the property ϕ does not hold true.

Marabou

1. Marabou uses SMT solver to verify the network's properties by transforming them into constraint satisfaction problem.
2. Marabou main components:
a. Simplex Core- determines a satisfactory assignment of linear constraints..
b. SMT - handles piece-wise linear constraint splits.
c. Bound tightening - propagates bounds through the network.

ReluVal

1. To compute rigorous bounds on the outputs of a DNN, ReluVal replaces SMT Solver with interval arithmetic.
2. ReluVal uses two optimization method to tighten the bounds:
a. Symbolic intervals: Tracks the symbolic lower and upper bound of each neuron. ReluVal correctly handles the non-linear functions such as ReLUs when transfering symbolic bound constraints across a DNN.
b. Iterative interval refinement: ReluVal bisects the input range recursively and repeats the range propagation on the smaller input ranges, it does this when the output range is too large. [1]

ACAS Xu Details

ACAS Xu is an Airborne Collision avoidance system for drones. It is being developed by the US Federal Aviation Administration(FAA). It is a system that is mounted on the airborne drone (Ownship) and it reads sensor information of another drone (intruder) that is nearby. The ownship tries to read information like speed, distances, angles. The system then is responsible for producing an advisory (suggestion) of what it should do when an intruder is present nearby.

The ACAS Xu system consists of 45 deep neural networks (DNNs). Each network is composed of an input layer taking 5 inputs, an output layer generating 5 outputs, and 6 hidden layers with each containing 50 neurons.

Five inputs include:
ρ: the distance between ownship and intruder
θ: the heading direction angle of ownship relative to the intruder
ψ: the heading direction angle of the intruder relative to ownship
vown: the speed of ownship
vint: the speed of intruder. [2]

Output of the DNN includes:
1. COC: clear of conflict
2. Weak left: heading left with angle 1.5 degree/s
3. Weak right: heading right with angle 1.5 degree/s
4. Strong left: heading left with angle 3.0 degree/s
5. Strong right: heading right with angle 3.0 degree/s. [3]

Workflow Diagram

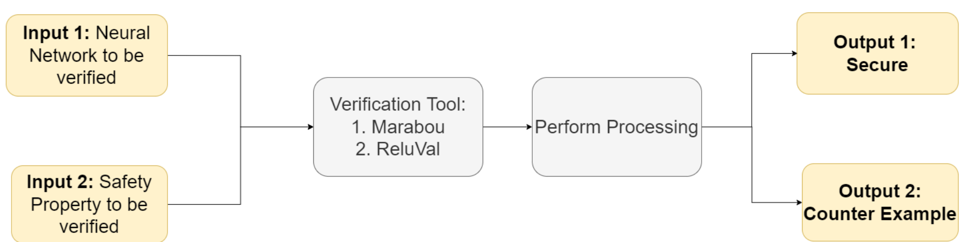


Figure 1. Workflow diagram of the system

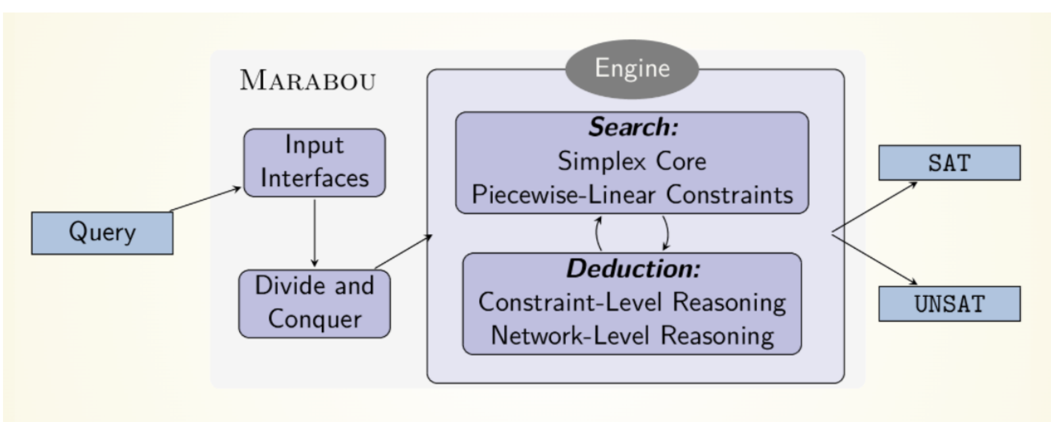


Figure 2. Workflow diagram of the Marabou Tool. .[4]

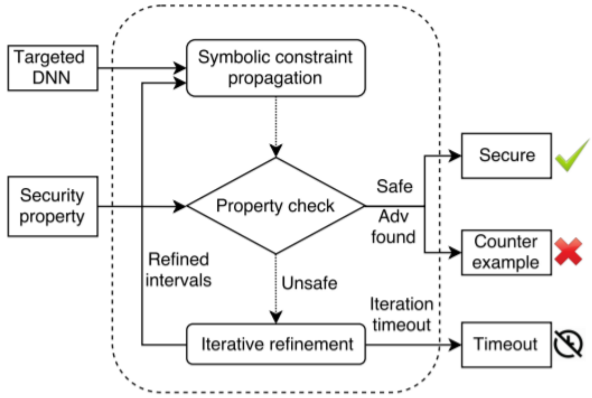


Figure 3. Workflow diagram of the ReluVal Tool.[3]

Tool Input Parameters
The tool requires two inputs.
1. Neural Network file (.nnet)
2. Property file which has the lower and the upper bound values.

Tool Outputs
Marabou Output:
1. UNSAT- when the property is not satisfied.
2. SAT- when the property is satisfied.

ReluVal Output:
1. No Adv- when the property specified is satisfied by the DNN.
2. Adv- when the property is not satisfied by the DNN. It also provides the bounded adversarial input set(counter example) in case of failure.

Property Description
The below list provides the property description of 4 properties that are used for the comparison of the two tools, Marabou and ReluVal.

Property 1: The intruder is distant and is significantly slower than the ownship, the score of a COC advisory will always be below a certain fixed threshold.
Input ranges: $\rho\ (\geq) 55947.691, vown \geq 1145, vint \leq 60$.

Property 2: If the intruder is distant and is significantly slower than the ownship, the score of a COC advisory will never be maximal.
Input ranges: $\rho \geq 55947.691, vown \geq 1145, vint \leq 60$.

Property 3: If the intruder is directly ahead and is moving towards the ownship, the score for COC will not be minimal.
Input ranges: $1500 \geq \rho \geq 1800, 0.06 \leq \theta \leq 0.06, \psi \geq 3.10, vown \geq 980, vint \geq 960$.

Property 4: If the intruder is directly ahead and is moving away from the ownship but at a lower speed than that of the ownship, the score for COC will not be minimal.
Input ranges: $1500 \leq \rho \leq 1800, 0.06 \leq \theta \leq 0.06, \psi = 0$, vown $\geq 1000, 700 \leq vint \leq 800$.

Experiment Results

Marabou and ReluVal tool were installed on the system and was ran on one of the ACAS Xu neural network (ACASXU_experimental_v2a_2_7.nnet) along with the property 3 as specified above.

Operating System: Ubuntu 18
RAM: 8GB

Result of Marabou tool

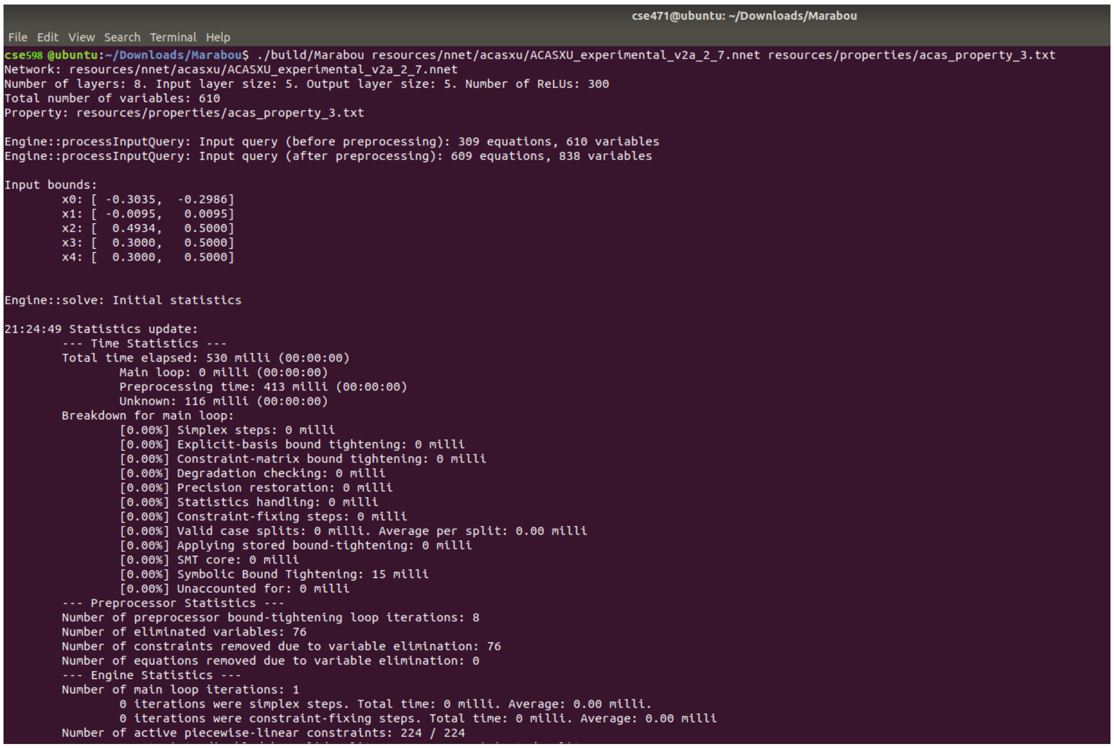
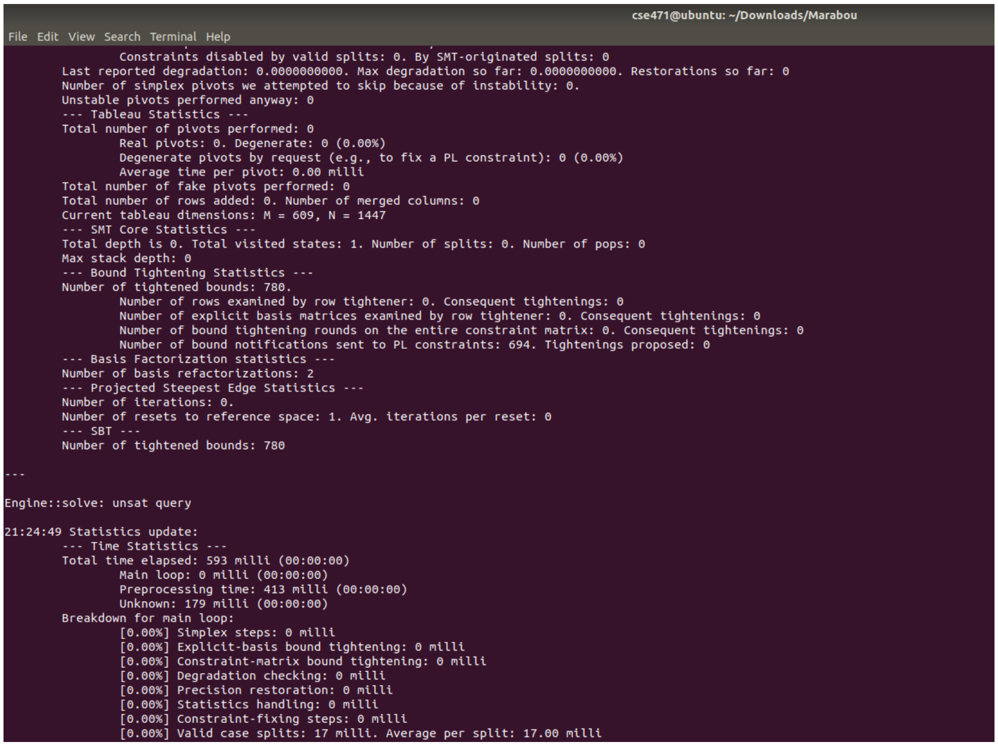


Figure 4. Output Marabou



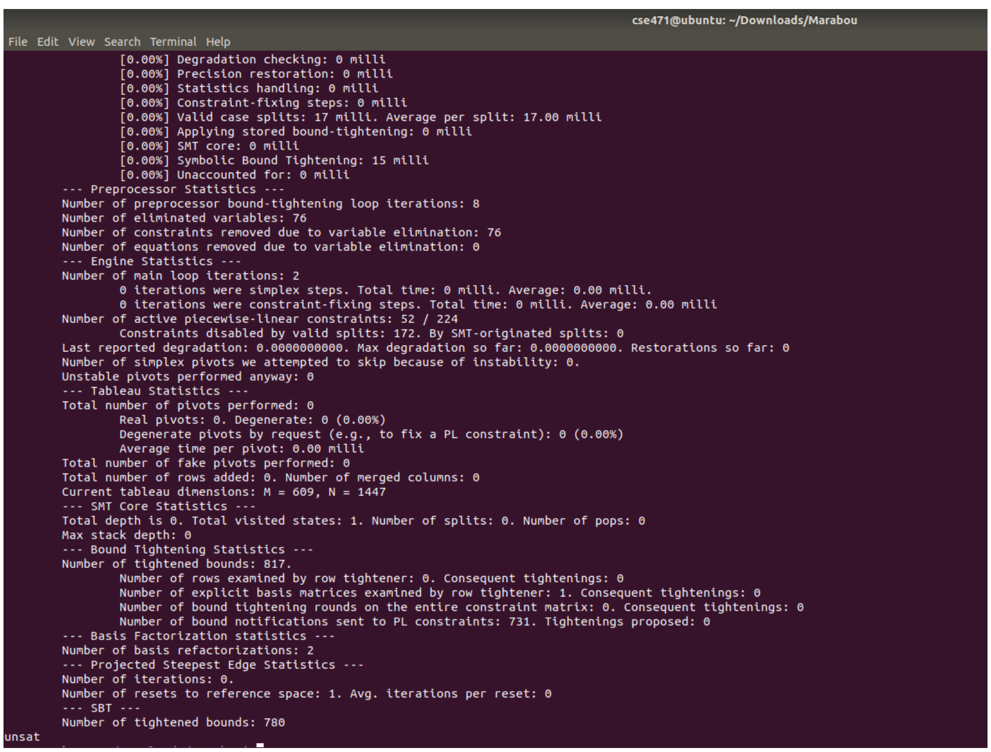
```
cse471@ubuntu: ~/Downloads/Marabou
File Edit View Search Terminal Help
                Constraints disabled by valid splits: 0. By SMT-originated splits: 0
        Last reported degradation: 0.0000000000. Max degradation so far: 0.0000000000. Restorations so far: 0
        Number of simplex pivots we attempted to skip because of instability: 0.
        Unstable pivots performed anyway: 0
        --- Tableau Statistics ---
        Total number of pivots performed: 0
                Real pivots: 0. Degenerate: 0 (0.00%)
                Degenerate pivots by request (e.g., to fix a PL constraint): 0 (0.00%)
                Average time per pivot: 0.00 milli
        Total number of fake pivots performed: 0
        Total number of rows added: 0. Number of merged columns: 0
        Current tableau dimensions: M = 609, N = 1447
        --- SMT Core Statistics ---
        Total depth is 0. Total visited states: 1. Number of splits: 0. Number of pops: 0
        Max stack depth: 0
        --- Bound Tightening Statistics ---
        Number of tightened bounds: 780.
                Number of rows examined by row tightener: 0. Consequent tightenings: 0
                Number of explicit basis matrices examined by row tightener: 0. Consequent tightenings: 0
                Number of bound tightening rounds on the entire constraint matrix: 0. Consequent tightenings: 0
                Number of bound notifications sent to PL constraints: 694. Tightenings proposed: 0
        --- Basis Factorization statistics ---
        Number of basis refactorizations: 2
        --- Projected Steepest Edge Statistics ---
        Number of iterations: 0.
        Number of resets to reference space: 1. Avg. iterations per reset: 0
        --- SBT ---
        Number of tightened bounds: 780

---

Engine::solve: unsat query

21:24:49 Statistics update:
        --- Time Statistics ---
        Total time elapsed: 593 milli (00:00:00)
                Main loop: 0 milli (00:00:00)
                Preprocessing time: 413 milli (00:00:00)
                Unknown: 179 milli (00:00:00)
        Breakdown for main loop:
                [0.00%] Simplex steps: 0 milli
                [0.00%] Explicit-basis bound tightening: 0 milli
                [0.00%] Constraint-matrix bound tightening: 0 milli
                [0.00%] Degradation checking: 0 milli
                [0.00%] Precision restoration: 0 milli
                [0.00%] Statistics handling: 0 milli
                [0.00%] Constraint-fixing steps: 0 milli
                [0.00%] Valid case splits: 17 milli. Average per split: 17.00 milli
```

Figure 5. Output Marabou



```
cse471@ubuntu: ~/Downloads/Marabou
File Edit View Search Terminal Help
                [0.00%] Degradation checking: 0 milli
                [0.00%] Precision restoration: 0 milli
                [0.00%] Statistics handling: 0 milli
                [0.00%] Constraint-fixing steps: 0 milli
                [0.00%] Valid case splits: 17 milli. Average per split: 17.00 milli
                [0.00%] Applying stored bound-tightening: 0 milli
                [0.00%] SMT core: 0 milli
                [0.00%] Symbolic Bound Tightening: 15 milli
                [0.00%] Unaccounted for: 0 milli
        --- Preprocessor Statistics ---
        Number of preprocessor bound-tightening loop iterations: 8
        Number of eliminated variables: 76
        Number of constraints removed due to variable elimination: 76
        Number of equations removed due to variable elimination: 0
        --- Engine Statistics ---
        Number of main loop iterations: 2
                0 iterations were simplex steps. Total time: 0 milli. Average: 0.00 milli.
                0 iterations were constraint-fixing steps. Total time: 0 milli. Average: 0.00 milli
        Number of active piecewise-linear constraints: 52 / 224
                Constraints disabled by valid splits: 172. By SMT-originated splits: 0
        Last reported degradation: 0.0000000000. Max degradation so far: 0.0000000000. Restorations so far: 0
        Number of simplex pivots we attempted to skip because of instability: 0.
        Unstable pivots performed anyway: 0
        --- Tableau Statistics ---
        Total number of pivots performed: 0
                Real pivots: 0. Degenerate: 0 (0.00%)
                Degenerate pivots by request (e.g., to fix a PL constraint): 0 (0.00%)
                Average time per pivot: 0.00 milli
        Total number of fake pivots performed: 0
        Total number of rows added: 0. Number of merged columns: 0
        Current tableau dimensions: M = 609, N = 1447
        --- SMT Core Statistics ---
        Total depth is 0. Total visited states: 1. Number of splits: 0. Number of pops: 0
        Max stack depth: 0
        --- Bound Tightening Statistics ---
        Number of tightened bounds: 817.
                Number of rows examined by row tightener: 0. Consequent tightenings: 0
                Number of explicit basis matrices examined by row tightener: 1. Consequent tightenings: 0
                Number of bound tightening rounds on the entire constraint matrix: 0. Consequent tightenings: 0
                Number of bound notifications sent to PL constraints: 731. Tightenings proposed: 0
        --- Basis Factorization statistics ---
        Number of basis refactorizations: 2
        --- Projected Steepest Edge Statistics ---
        Number of iterations: 0.
        Number of resets to reference space: 1. Avg. iterations per reset: 0
        --- SBT ---
        Number of tightened bounds: 780
unsat
```

Figure 6. Output Marabou

Result of ReluVal tool

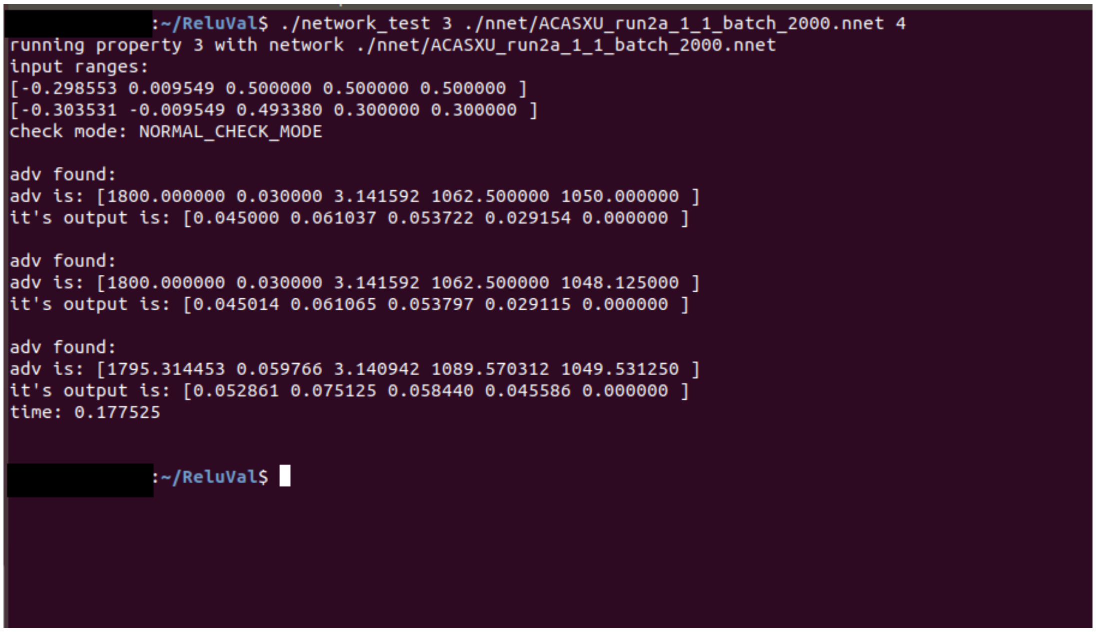


Figure 7. Output ReluVal

Performance Comparison

The above results demonstrate the following observation:
a. ReluVal took **170** milliseconds to run the property 3 file on one of the neural network file.
b. Marabou took **593** milliseconds to run the property 3 file on one of the neural network file.

Observations:
1. It can be seen that ReluVal is efficient and faster for neural network verification in comparison to Marabou framework.
2. Both the tools gave the same output that is property 3 not satisfied.

The Marabou paper provides the same observation. It can be seen from the figure below that ReluVal takes less time as compared to Marabou for verification.

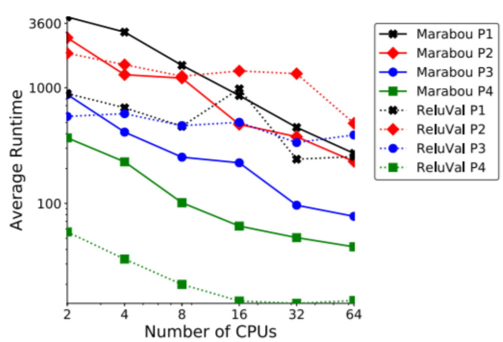


Figure 8. Comparison of ReluVal and Marabou [4]

Output of tool when random errors introduced

The input property file was modified to add some noise to the lower and upper bounds and then given as input to the Marabou and ReluVal tool.
Reluval was stable and did not deviate much from specified targets, whereas Marabou suffered from deviations.

Status and Lessons learned

I was able to successfully perform the comparison of verification performance of two tools and follow the schedule and timeline proposed in the project proposal. I would like to further explore the detailed working of the tools and work towards developing one such verification tool which can verify simple classification based neural network.

Conclusion

The system compares the verification performance of Marabou and ReluVal on the ACAS Xu Airborne Collision Avoidance System for Unmanned Aircraft. The ACAS Xu DNN was successfully executed using Marabou tool and ReluVal. Four properties were used to verify the performance of the ACAS Xu system.

References

1. Guy Katz, D. L. D. K. J. M. J. K., Clark Barrett. Reluplex: An efficient smt solver for verifying deep neural networks. (2017).

2. Michael P. Owen, R. M. L. A., Adam Panken & Leeper, C. Acas xu: Integrated collision avoidance and detect and avoid capability for uas. (2019).

3. Shiqi Wang, J. W. J. Y., Kexin Pei & Jana, S. Formal security analysis of neural networks using symbolic intervals. (2018).

4. Guy Katz, D. I. K. J. C. L. R. L. P. S. S. T. H. W. A. Z. D. L. D. M. J. K. C. B., Derek A. Huang. The marabou framework for verification and analysis of deep neural networks. (2019).