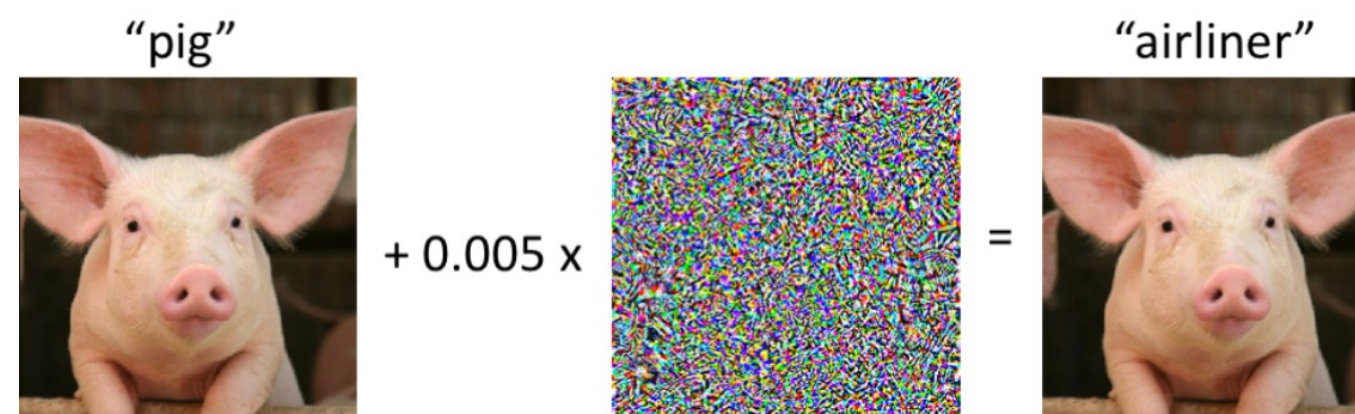# Speeding Up Neural Network Robustness Verification via Algorithm Configuration and an Optimised Mixed Integer Linear Programming Solver Portfolio

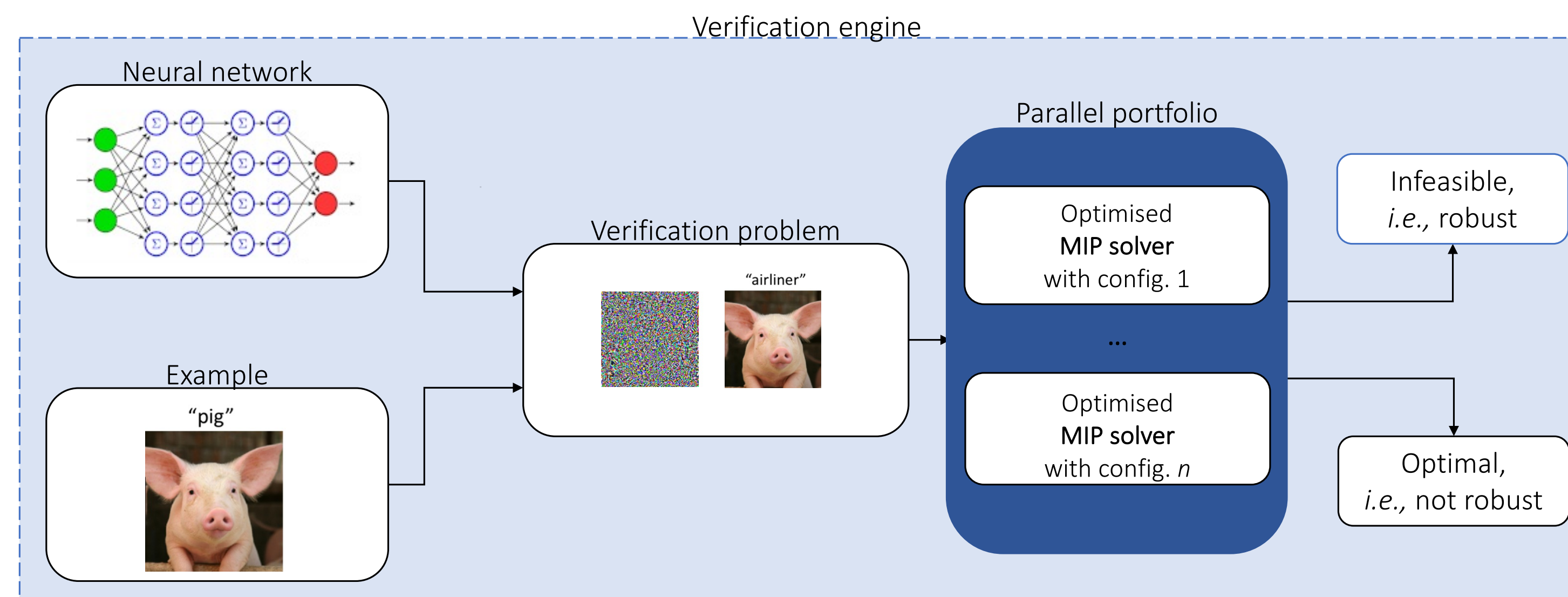Matthias König, Holger H. Hoos, Jan N. van Rijn ▪ LIACS, Leiden University, Leiden, The Netherlands

Universiteit Leiden

Background

- Neural networks are vulnerable to adversarial examples

"pig"   + 0.005 x   =   "airliner"

- Several neural network verification methods are based on mixed integer linear programming (MIP)

- Problem: High computational costs and many timeouts

Method

- Parallel portfolio of optimised solver configurations
- 1 CPU core per configuration

Verification engine

Neural network

Example — "pig"

Verification problem — "airliner"

Parallel portfolio

Optimised MIP solver with config. 1

...

Optimised MIP solver with config. *n*

Infeasible, *i.e.,* robust

Optimal, *i.e.,* not robust

Conclusions

- Improved performance of state-of-the-art MIP-based verification engine **MIPVerify**:
  - Up to 4.7-fold reduction in CPU time
  - Up to 1.4 times fewer timeouts

- Improved performance of state-of-the-art MIP-based verification engine **Venus**:
  - Up to 10.3-fold reduction in CPU time
  - Up to 2.8 times fewer timeouts

Results

(i) MIPVerify − SDP_d MLP_A

(ii) MIPVerify − mnistnet

(iii) Venus − SDP_d MLP_A

(iv) Venus − mnistnet