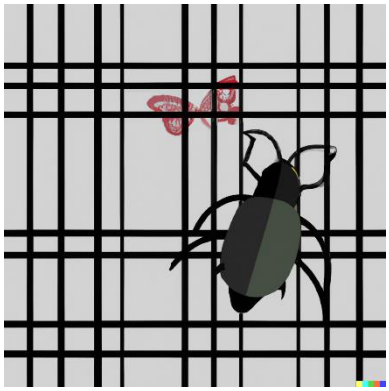
Hardware vulnerability detection using Graph Neural Networks (GNN)



Background: Security verification, in particular for hardware, is a challenging task, since it requires to cover a wide scenario space, usually much wider than for the functional verification. To assist the hardware security verification of software, the Common Weaknesses Enumeration database was recently introduced. The classical verification approach requires writing tests to cover each of the scenarios. This approach is known as dynamic verification. This task is both labor-intensive and incomplete due to the huge space. The community is seeking alternatives to the dynamic verification approach, exploring various types of static verification. In static verification, rather than writing tests to cover scenarios, we define rules and try to prove these rules on a circuit using analytical methods. In this project, we are interested to check the machine learning ability to detect security vulnerabilities. In particular, we will apply the Graph Neural Network techniques to that task.

Project Description: In this project, the students will apply various Graph Neural Network techniques and architectures in an attempt to detect security vulnerabilities in logic circuits. The students will build a set of benchmarks containing one of the vulnerabilities from the CWE database. The benchmarks will be written in RTL (SystemVerilog) and synthesized to obtain gate-level netlist. The netlist will be further converted to a graph structure. Some of the obtained graphs will be used to train the GNN, while the others will be used as an evaluation set.
In the course of the project, the students will acquire knowledge in logic synthesis, circuit analysis as well as machine learning using GNN.

Prerequisites: Logic Design, Algorithms, Machine Learning, Lab 1.

Supervisor: Leonid Azriel (*leonida@technion.ac.il*)