ICS Anomaly Detection Lab

CPS and IoT Security 2022/23 - University of Padua

In this lab, we will look at a possible way to detect attacks in an ICS using artificial intelligence. In particular, we will build an autoencoder that is able to discriminate between normal and abnormal behavior. We will test it on a dataset described in the following section.

HAI (HIL-based Augmented ICS) Security Dataset

The HAI dataset was collected from a realistic industrial control system (ICS) testbed augmented with a Hardware-In-the-Loop (HIL) simulator that emulates steam-turbine power generation and pumped-storage hydropower generation.

As per documentation, the testbed is composed of four different processes: boiler process, turbine process, water treatment process and HIL simulation:

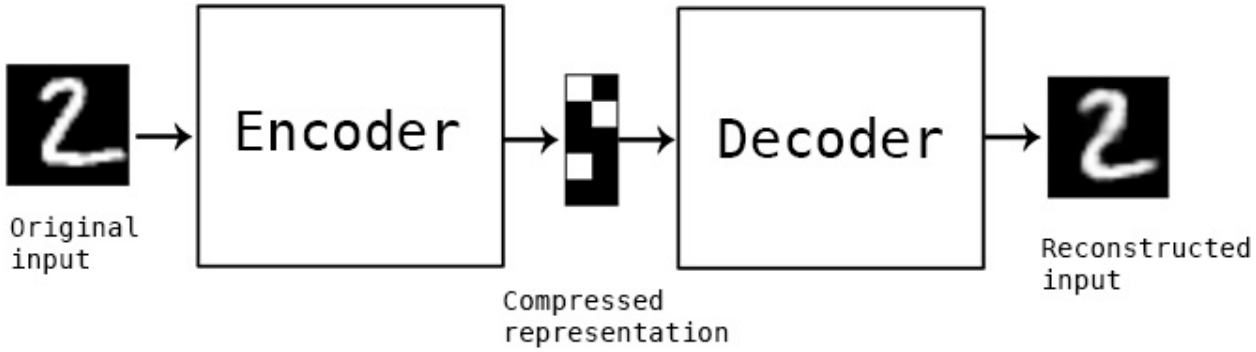
- **Boiler Process (P1)**: This includes water-to-water heat transfer at low pressure and a moderate temperature.
- **Turbine Process (P2)**: A rotor kit process that closely simulates the behavior of an actual rotating machine.
- **Water treatment Process (P3)**: This process includes pumping water to the upper reservoir and releasing it back into the lower reservoir.
- **HIL Simulation (P4)**: Both the boiler and turbine processes are interconnected to synchronize with the rotating speed of the virtual steam-turbine power generation model. The pump and value in the water-treatment process are controlled by the pumped-storage hydropower generation model.

Three major versions of HAI datasets have been released thus far. Each dataset consists of several CSV files, and each file satisfies time continuity.

The time-series data in each CSV file satisfies time continuity. The first column represents the observed time in the “yyyy-MM-dd hh:mm:ss” format, while the rest of the columns provide the recorded SCADA data points. The last four columns provide data labels for whether an attack occurred or not. Out of these four columns, it is

applicable to all the processes, and the other three columns are applicable to the corresponding control processes.

Autoencoder



An autoencoder is a type of artificial neural network used to learn efficient codings of unlabeled data (unsupervised learning).
An autoencoder learns two functions:

- an encoding function that transforms the input data
- a decoding function that recreates the input data from the encoded representation

The autoencoder learns an efficient representation (encoding) for a set of data, typically for dimensionality reduction, but can also be used for anomaly detection.

Autoencoder for anomaly detection

In this case, we use an autoencoder for anomaly detection. The intuition: if we train our autoencoder on normal data, the reconstruction error on malicious data would be higher with respect to normal data. Therefore, in the attached code, we will train our autoencoder in one of the training datasets, which is composed of only benign samples. Then, we extract the standard deviation of the loss as a threshold to discriminate between attacks and normal samples. Since we will use only benign data during training, this process is usually called One Class Classification. This approach employs machine learning, in particular, a simple neural network. No big optimization has been done in the code.

You can try different modification to the code:

- Play with the model by adding/removing layers or by fine-tuning some parameters to improve the final results
- Employ other versions of the dataset
- Merge different datasets (of the same version)
- Implement another model