Benchmark: Neural Networks for Anomaly Detection in Batch Distillation

Simon Lutz[1,2] and Daniel Neider[1,2]

[1] TU Dortmund University, Germany
[2] Center for Trustworthy Data Science and Security, UA Ruhr, Germany

Abstract. In recent years, deep learning has achieved considerable success in the field of anomaly detection. These models have to work safely and reliably when deployed in safety-critical applications, such as chemical plants: Failure to report anomalies may result in imminent hazards to the environment and harm to human life. Contrary, false alarms may lead to substantial financial or scientific loss due to unnecessary downtime of a plant. In this paper, we present a benchmark suite for verifying neural networks used in anomaly detection for the batch distillation chemical process.

This work presents the abstract for a talk at AISoLA 2023.

Anomaly detection is the task of detecting behavior which diverges from the norm. Motivated by its the success in other applications, there has been a rise of interest in developing deep learning approaches for anomaly detection in recent years. Before deploying these models in safety-critical applications, such as chemical plants, we need to ensure they work safely and reliably: Failure to report anomalies may result in imminent hazards to the environment and harm to human life. Contrary, false alarms may lead to substantial financial or scientific loss due to unnecessary downtime. In this talk, we present a benchmark suite for verifying neural networks used in anomaly detection for batch distillation, a chemical process. This benchmark suit arises from an interdisciplinary project on deep learning on chemical process data.

Batch distillation is a chemical process that uses a so-called distillation column (see Figure 1) to separate a liquid mixture into multiple, possible unknown chemical components. To this end, the mixture is filled into the distillation still, a vessel connected to a heat source, and heated up until it boils. The resulting vapor rises up through the rectifying column, containing a sequence of packings or plates, until it is condensed in the condenser at the top of the column. The condensate is cooled and split into two separate streams. One part of the condensate is withdrawn as distillate while the other, often larger fraction of the condensate called reflux is returned to the column top. There, the countercurrent of downflowing liquid and uprising vapor is constantly in contact, enabling a material flow improving the separation of the components in the mixture.

By measuring, for instance, the temperature, pressure, and concentration of the individual components throughout multiple experiments we collect a data

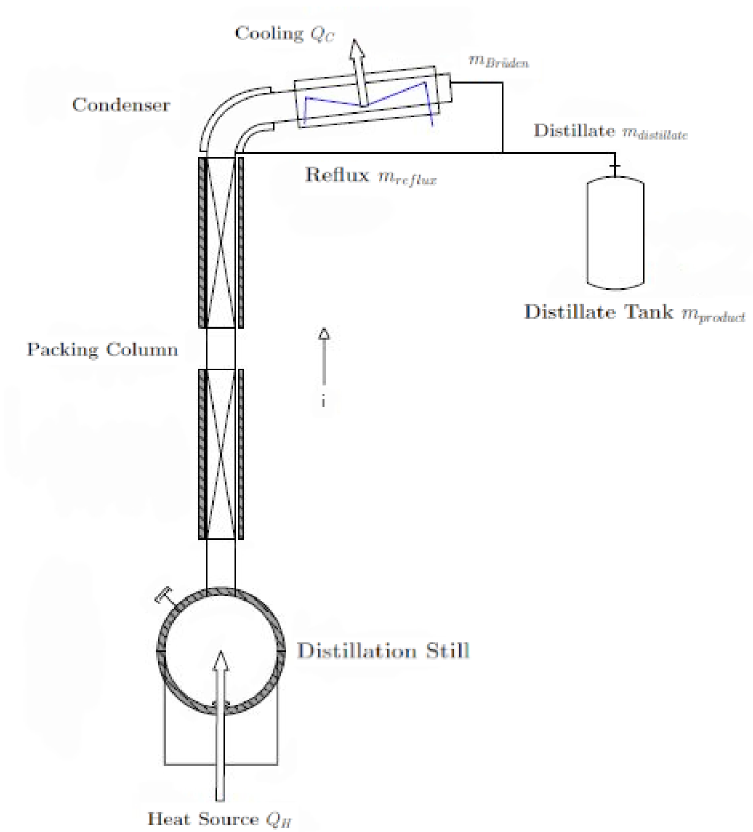

Fig. 1. The simplified schematics of a distillation column

set of multi-modal sequential data on which we train a variety of (deep) anomaly detectors. These models take multi-modal sequences of a fixed length k as input (typically the values of the last k points of time) and predict whether the current time point is an anomaly or not.

To verify the safety and reliability of these anomaly detectors, we composed a list of specifications over multi-modal sequences which need to be fulfilled when the batch distillation works properly. Conversely, whenever one of the properties is violated the anomaly detector should report an anomaly.

In order for an automated verifier to process theses specifications, we formalize them in a specification logic. Working on multi-modal sequences of measurements over time, this specification logic must be able to express two main properties: On the one hand, we need to express relations between real-valued variables, e.g., the values of a temperature sensor. One the other hand, we need to model temporal relations and properties, for instance, that a condition will always hold. To this end, our specification logic combines the theory of Linear Real Arithmetic (LRA), to express relations between real-valued variables, and Linear Temporal Logic [1], to express the temporal properties. Note that we consider only sequences of equal and finite length which would allow us to just use LRA by introducing a variable for each point in the sequence. To not clutter this section to much we omit a formal definition of our specification language at this point. Whenever we refer to "measurements", we mean a specific input to the neural network at a fixed position in its input.

We categorize our formal specifications into three groups: The first group consists of properties that define bounds on the measurements. This includes properties like 'The pressure must always be greater than zero'. Recall that the inputs to our network are multi-modal sequences of, for instance, temperature, pressure, or concentrations of a component s measured over time at different positions in the distillation column. We denote each such sequence by t_i, p_i, and c_i^s, respectively, where the index indicates the position within the column (from distillations still to the top of the column). Using the temporal operator G (globally) to indicate that a property has to hold at any point of the sequence, we formalize the above property as follows:

$$\mathrm{G}(\bigwedge_{i=0}^{n_p} p_i > 0)$$

The second group of specifications defines relations between two measurements, either of different sensors or at different points in time. These include properties like 'The temperature has to decrease from distillation still to the top of the packing column' or 'The temperature in the distillation still increases over time'. For instance, we formalize the former specification as:

$$\mathrm{G}(\bigwedge_{i=0}^{n_t-1} t_i > t_{i+1})$$

Due to the nature of the ongoing chemical processes, the conditions within the distillation column can not change too rapidly but must exhibit a certain degree of continuity. Preserving this continuity, for instance, over time, is expressed by the specifications of the third group. One example for such a specification is 'Continuity over time for pressure (i.e., the value may only change by at most ε_p)' which we formalize as:

$$\mathrm{G}(\bigwedge_{i=0}^{n_p} p_i - \varepsilon_p \leq \circleddash p_i \wedge \circleddash p_i \leq p_i + \varepsilon_p)$$

In conclusion, this talk will present a benchmarks set arising from chemical process engineering. More precisely, the neural networks trained are deep anomaly detectors, and the specifications express sanity checks on these networks. Until the conference in October, we plan to train a variety of (deep) anomaly detectors. Afterwards, we will evaluate our benchmark suite and make the results as well as the benchmark suite publicly available.

References

[1] Amir Pnueli. "The temporal logic of programs". In: *18th Annual Symposium on Foundations of Computer Science (sfcs 1977)*. 1977, pp. 46–57. DOI: 10.1109/SFCS.1977.32.