Abstract—The gradual increase in successful attacks against Industrial Control Systems (ICS) highlights the pressing need to implement defense mechanisms to accurately and quickly detect resulting process anomalies. Industrial systems are critical and very sensitive systems and an anomaly or a simple intrusion could have serious consequences. Considering their particularity and their criticality, it becomes urgent to set up Intrusion Detection Systems (IDS) in order to be able to detect process-oriented attacks. Signature-based IDSs that already exist do not meet our requirements because they usually identify abnormal behavior by matching it against pre-defined patterns of events that describe known attacks. As part of this work, we will set up an anomaly-based IDS which profiles normal behavior and attempts to identify anomaly patterns of activities that deviate from the defined profile in order to be able to detect new attacks. We will use artificial intelligence techniques for behavioral detection of anomalies and the objective is not only to have good performance of our model, but also and above all to understand how the model makes predictions. The model that we will train must be explainable so that operators can trust it.

Index Terms—Industrial control systems, Anomaly-based, Intrusion detection, Explainable Artificial Intelligence

I. BRIEF OVERVIEW OF KEY CONCEPTS

A. Industrial Control Systems

An industrial control system (ICS) is a set of devices, networks, and software designed to monitor and control industrial processes in various environments, such as factories, power plants, chemical plants, production systems, and other industrial installations. The main objective of an industrial control system is to automate and monitor industrial operations, thereby ensuring that processes operate efficiently, securely and consistently.

Industrial systems are distinguished from traditional computer networks (IT systems) on the one hand by the presence of a physical domain for the ICS (sensors, actuators).

On the other hand, the security of ICS is much more crucial due to the critical nature of industrial processes. Furthermore, in ICS, priority is given to availability, then integrity and finally confidentiality (AIC) which is the opposite than for IT systems which operate according to the CIA priority triangle. This is due to the fact that industrial control systems (ICS) typically operate in real time, providing real-time monitoring and regulation of physical equipment. In this context, immediate responsiveness is essential to prevent failures, accidents or disruptions. Thus, system availability is imperative to ensure a rapid and efficient response.

B. Intrusion Detection Systems and Explainable AI for ICS

As part of this work, we will develop a Network Anomaly-based IDS that analyses the frames payload and using Machine Learning techniques.

The explainability of artificial intelligence (AI) models refers to the ability to capture and detail the decision-making process of the model. It encompasses clarity of the internal working of the model, providing users, developers and stakeholders with the ability to grasp the elements and traits that impact the results produced by the model. These are some reasons which justify the importance of the explainability of machine learning models specifically for the detection of intrusions in the context of ICS:

- **The need for trust**: when an intrusion is detected, the operator needs to trust the relevance and accuracy of this detection and for this, he needs a clear explanation of the decision-making process.
- **Understanding of contexts**: in the context of ICS, an instance is considered an anomaly depending on the context of the process. Explainability allows us to understand why an instance in a specific context is judged as problematic and in another it can be seen as normal.
- **Identifying and reducing false positives**: it may happen that the model makes a mistake declaring an instance as an intrusion when it is not. This error in ICS can have important consequences especially if action is taken after detection; an explainable model allows us to better understand why the model was wrong and this allows us to better distinguish a real anomaly from a model error.
- **Model optimization**: a clear understanding of how the model interprets features allows the model to be better adjusted for better (more accurate) anomaly detection.

II. STATE OF THE ART

In the project on SCADA Anomaly Detection [1], the aim is to determine the selection criteria for machine learning algorithms for anomaly detection in SCADA systems, and to examine the effectiveness of several machine learning algorithms.
algorithms on SCADA systems. The advantage of this work is that they use several Machine learning algorithms to carry out tests on three completely different datasets (water, gas and electricity). The disadvantage, however, is that the datasets that are used (water and gas) contain several pieces of data that are correlated with each other and which literally distorts the results returned by the machine learning algorithms.

In most existing works, the authors identify attack scenarios and train the models on various ML algorithms. This is the case for example of [4]. One of the limits of this approach is that the description of attack scenarios does not have a common basis, and can vary from one paper to another. Furthermore, the algorithms used to train the models are mostly chosen empirically which can be problematic because in ICS systems we deal a lot with sequential data. We should therefore intentionally choose the algorithms to use. Regarding the explainability of these models, the survey published in 2022 [5] lists and classifies the explainability methods for classification in cybersecurity. The authors highlighted the methods that can be used for i) transparency and trust, ii) classifier performance and iii) error explanation.

III. Work In Progress

Our goal is to be able to use real data sets, therefore coming from a company with machines in which processes run. But, for a start, we begin with a simple model in order to obtain a dataset on which we can easily train the data.

Assume that we have a controller and the program running on this controller will remotely open or close a valve while periodically reading the state of the valve. The valve can be in only two states: open (1) or closed (0). The different actions of the automaton are Write(W) controls to the valve and Read(R) and the valve state which returns the response through Answer(A). Figure 1 depicts the process thus described.

![Fig. 1. Process Description](image)

We consider the following three assumptions:
1) **First assumption**: After a write the state changes.
2) **Second assumption**: The valve state changes only after a write.
3) **Third assumption**: There are no "useless" writes (like sending an "open" command if the valve is open).

By considering these assumptions, we can generate a dataset of normal behavior of our process using a python script. We also generate another dataset in which we introduce some abnormal lines as presented in Table I

The objective is to train our machine learning models using this data and to first see if the lines that we consider to be attacks are detected and secondly to understand how the model predicts them. The data as described is sequential while its interpretation depends on the previous state (data). A line, for example A 0 is interpreted as an attack only if the latest Write action preceding it was W 1 which clearly shows that the model must take previous actions into account. To solve this problem and to make things easier for our model which learns line by line, we add to our dataset a context which is made up of the last action, the last state of the valve observed and the last written value (Table II). Thus, the model will be able to learn gradually by taking into account previous actions.

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Normal</th>
<th>Attack</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>W 1</td>
<td>A 1</td>
</tr>
<tr>
<td>2nd</td>
<td>R -</td>
<td>R -</td>
</tr>
<tr>
<td>3rd</td>
<td>A 1</td>
<td>W 1</td>
</tr>
</tbody>
</table>

IV. Evaluation

We are doing unsupervised learning, and we are starting to train our models on Recurrent Neural Networks, which are well-suited to modeling data sequences. In particular, we are starting with Long Short-Term Memory, which models complex data sequences with long-term dependencies, enabling us to capture subtle patterns in the data that might indicate abnormal behavior. Later, we will also train the model on autoencoders and Convolutional Neural Networks.

Our work will be evaluated according to the performance of the model on the one hand and on the other hand according to its degree of explainability. In terms of performance, the model must maximize True Positives and True Negatives while minimizing False Positives and False Negatives.

REFERENCES