12th Conference on Learning Factories, CLF 2022

Learning Demonstrator for Anomaly Detection in Distributed Energy Generation

Timo Pelchen[a], Gregor Thiele[a,*], Axel Vick[a], Marcel Radke[a], David Schade[b], Jörg Krüger[c]

[a] *Department for Process Automation and Robotics, Fraunhofer Institute for Production Systems and Design Technology IPK, Berlin, Germany*
[b] *AUCOTEAM GmbH, Department for Energy Industry, Berlin, Germany*
[c] *Department for Industrial Automation Technology, Institute for Machine Tools and Factory Management IWF, Technical University of Berlin, Berlin, Germany*

Abstract

Machine learning based anomaly detection methods on process data can be used to secure critical infrastructure. The design and installation of these methods require detailed understanding of both the facilities and the machine learning methods. Therefore, they are mostly incomprehensible for non-experts and thus acting as a barrier hindering the fast spread of such technologies. This article presents the systematic development of a demonstrator which enables presentations of anomaly detection on the example of a simulated wind farm. The specially designed user-interface allows a comprehensive experience. This article documents the use of the demonstrator for experts experienced in energy systems which are interested in the application of machine learning algorithms.

Peer-review statement: Peer-review under responsibility of the scientific committee of the 12th Conference on Learning Factories 2022.

Keywords: Anomaly Detection; Learning Factories; Distributed Energy Generation; Modelica; Functional Mock-up Interface

1. Introduction

1.1. Motivation and Project Context

The advancing integration of renewable energies is leading to a fundamental transformation of the supply network. Wind power, biomass and solar energy lead to a decentralized feed-in. The distributed stations are controlled via regular internet lines. This provides an attack surface for cyber-crime and evokes a need for new protection methods. So-called network intrusion detection systems (NIDS) are widely established in critical infrastructure. These software solutions analyze the network traffic regarding statistical phenomena, like extraordinary frequencies of packages or

* Corresponding author. Tel.: +49-30-39006-394.
E-mail address: gregor.thiele@ipk.fraunhofer.de, ORCID: 0000-0002-7108-5203

atypical addresses. However, the packages are considered without any interpretation of the payload. Doing so could improve the security against various cyber-attacks.

The EnerSec project aimed to develop a method that detects anomalies and distortions in data traffic based on the payload. The required automated analysis of process data gives rise to the use of machine learning (ML). The method developed accordingly utilizes a neural network with a long-short-term-memory (LSTM) architecture to predict the regular behaviour of the monitored process. So, manipulations, or anomalies in the payload can be detected via the physical meaning of its content, or rather the deviation from the predictions even if there are only minor changes in the traffic, [1, 2]. Nevertheless, the implementation and operation of such solutions necessitates fundamental understanding of both the monitored facility and the applied ML methods, which requires a thorough training. Therefore, this article discusses the development of a mobile demonstrator for a hands-on education in the context of anomaly detection for distributed power stations and related facilities.

1.2. Research Goals and Outline

A major aspect to convey is the anomaly detection *on the payload* of the analyzed traffic, in contrast to NIDS which only consider the structure of the packages and statistics of the traffic in general. Here, we discuss whether a physical demonstrator allows for a better understanding of this fundamental difference.

This paper is structured as follows. Sec. 2 gives the theoretical background regarding learning environments along with demonstrators and refers to the chosen development approach. The derived concept is presented in Sec. 3, followed by a description of the technical modules in Sec. 4 whilst Sec. 5 gives user feedback, a conclusion and outlook to further research.

2. Theoretical Background

Learning Environments. Recent trends like digitalization and the ever-increasing complexity of technologies put new demands on employees to keep pace with these fast developments [3, 4]. Apart from pure textbook-knowledge, this harsh environment requires competences, which include the ability to tackle challenges in a creative manner and dealing with indeterminacy. Unfortunately, these qualities cannot be trained directly [4]. Learning environments have proven to be an effective instrument for conveying these competences, via actively involving trainees in action-oriented concepts and application-related experiences, opposed to purely transfering knowledge [3, 4]. In production-related areas mostly industry-scale environments, so-called learning factories are utilized, which comprise multiple stations picturing an authentic process and representing a real factory as closely as possible, including the technical and organizational aspects. However, due to their size, trainees are required to visit the learning factory physically [3]. Concentrating on the application field of energy generation, 1:1 scale test benches additionally pose safety risks (high voltage/current levels, etc.) on the participants. Both issues, i.e. the immobility and the safety hazards, can be mitigated via the use of small-scale demonstrators, which nevertheless are based on the same working principles (while being harmless) [5]. Hence, these demonstrators can be shown at trade fairs or on-site workshops [3].

Development Approach. In their paper, Tisch et al. [6] discuss the development of action-oriented, competency-based learning factories in detail, while Weyand et al. [3] adapt this approach for mobile demonstrators. This paper shows how the utilized approach can be applied to set up a mobile demonstrator exemplifying ML-based anomaly detection algorithms for distributed power generation.

3. Concept

Didactic Transformations. First, we need to derive a suitable demonstrator plan and the learning content from the education demands of the respective target group. As discussed in [3], there are two didactic transformations needed before the technical implementation. The first didactic transformation identifies the target group, relevant aspects and phenomena, as well as deriving adequate learning goals. The target groups, their state of knowledge and the determined learning goals are listed in Tab. 1. In the second didactic transformation, the examined system is simplified and relevant

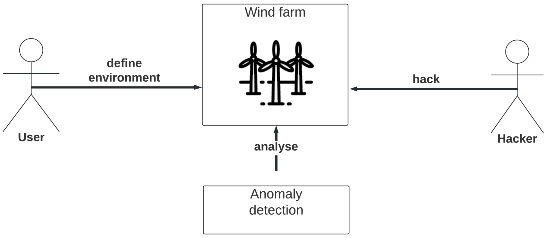


Fig. 1: Scenario.

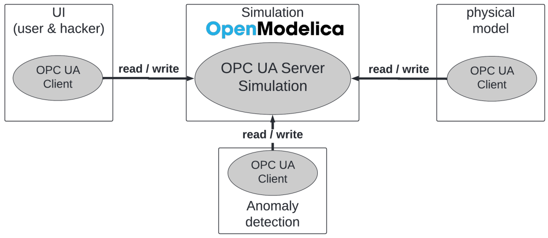


Fig. 2: Architecture of the communication.

aspects are extracted and abstracted. Finally, analogies are sought, which then can be implemented technically. The actual choice of analogies for this project are shown in Tab. 2. Cells with the chosen analogies are marked grey.

Scenario. During the two didactic transformations, a clear scenario was defined. The system consists of
- simulated wind turbines,
- a user defining the environmental parameters (wind speed and power demand)
- and a hacker manipulating some of the values measured internally.

The anomaly detection presented in [1] analyses the process data of the wind turbines and detects irregularities. The basic setup is also shown in Fig. 1.

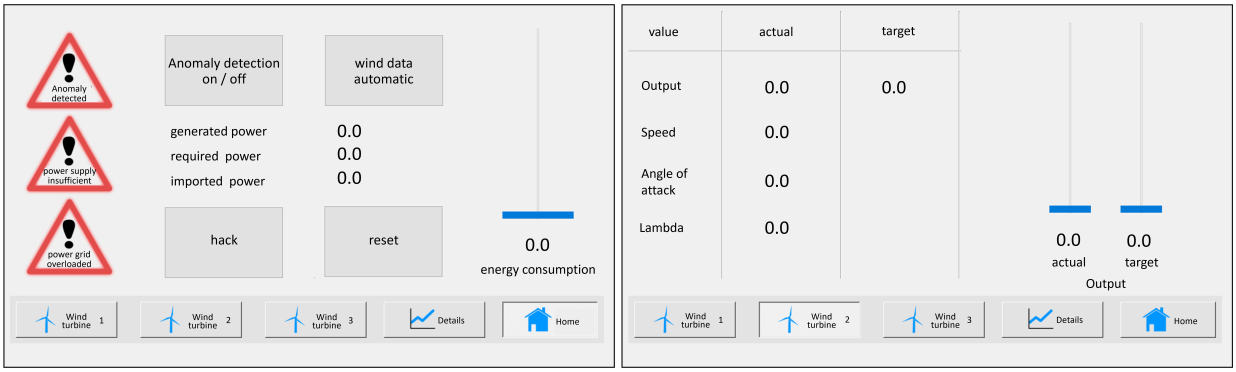
A first attack considers the power measured at the plant. In a stable operation point, the hacker reduces the (by the power control) measured value of the generated power to zero, while the actual output remains static. If the turbine was operating at the defined power limit and therefore being throttled (e.g. via pitch control), the hack would now lead the power control to believe the turbine would not generate any power, thus lifting all countermeasures. An overshoot of production would follow, leading to instabilities of the grid and technical defects in the worst case.

Table 1: Target group and learning goals

	Target Group	**Learning Goals**
Automation engineers for energy systems	- Expertise on energy systems - Basic skills in programming - Inexperienced but interested in ML	- Illustrate capabilities of ML - Show chances for transparency in ML
Software engineers for energy control systems	- Programming professionals - Sound understanding of algorithms - Some knowledge about ML	- Illustrate potentials of ML - Inspire for using ML in own projects - Discuss issues and debugging

Table 2: Morphological analysis for simplifications of relevant aspects adapted from [3]

Initial System	**Abstract System**	**Analogies**	
Distributed generation	energy resource	physical	simulated
		wind turbines	solar power plant
Environment	system input	user	auto-generated
Anomaly	irregularity	auto-generated	hacker
Anomaly detection	artificial intelligence	python script	
Network	communication	MQTT	OPC UA



(a) Front-end with action buttons and alert messages (b) Front-end with control sliders and process data monitoring

Fig. 3: The front-end of the user interfaces allows for both manipulation and control settings.

4. Technical Implementation

This section introduces the main components of the demonstrator. First, the simulation of the dynamic behaviour and the ICT communication between all entities are presented in Sec. 4.1. Then, the user interface is explained in Sec. 4.2, while the miniature wind park is illustrated in Sec. 4.3. The anomaly detection and the set-up are shown in Sections 4.4 and 4.5, respectively.

4.1. Simulation using Modelica and Communication via Middleware

The simulation is the key component of this demonstrator, since it has to enable all the other components to access the generated data. We chose the Modelica based open-source simulation environment OpenModelica because of its implementation of the middleware OPC UA, and the availability of a ready-made Modelica library[1] offering a basic model of a wind turbine, which was presented by Eberhard et al. [7].

The implemented OPC UA acts as a platform for communication across a broad variety of entities. Therefore, OpenModelica offers an interface for exchange with other programs at runtime [8]. More specifically, OpenModelica starts an OPC UA server during interactive simulations and mirrors all simulation values onto that server. These values can then be read and modified by other entities via OPC UA clients. This is the case for the UI and the anomaly detection in the presented demonstrator. The communication architecture of this demonstrator is shown in Fig 2.

4.2. User Interface

The mobile demonstrator needs to offer user interaction. More precisely, next to the user mode, where environmental parameters can be defined (see Fig. 3a), there should be another mode allowing the trainee to act as a hacker who attempts to harm the wind park simulated in the scenario (Fig. 3b). Accordingly, a touch panel allows the user to manipulate specific values of the simulation. A size of seven inches is the trade-off between portability and clear overview. The panel is connected to a Raspberry Pi 3 Model B+ (RPi). The interactive dashboard is implemented using Qt.

4.3. Physical Wind Energy Plant

The physical set-up is designed to be portable and robust. So, we chose a metal box and mounted a miniature landscape in the lid, including a wind park as shown in Fig. 4. Three miniature wind turbines act as a visualization of the simulation. Using H-Bridges, the turbines are powered via an external 5 V supply and controlled via a pulse width

[1] https://github.com/christiankral/WindPowerPlants

Fig. 4: Picture of the demonstrator.

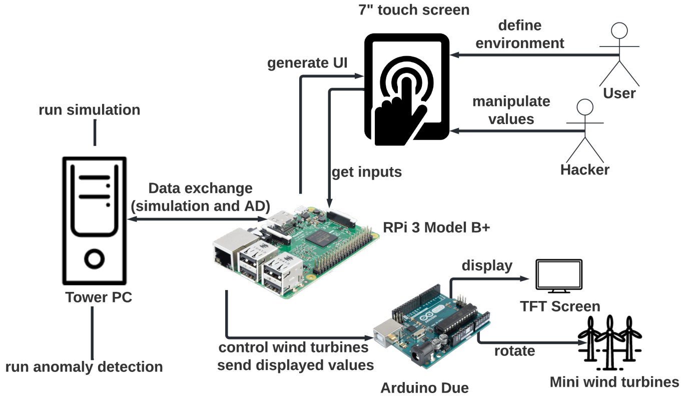


Fig. 5: Set up of the demonstrator.

modulation using the general purpose input/output (GPIO) pins of an Arduino Due mirroring the values given from the simulation. Additionally, a small in-ground display shows the actual value of the electrical power produced.

Following the narrative, these physically correct values are then manipulated in the connection to the supervising control and further displayed on the touchscreen mentioned in Sec. 4.2. Therefore, the *real* state of the wind park is visible in the miniature models and the in-ground display, while the *manipulated* state can be seen on the touchscreen. This allows for an intuitive understanding of causalities and the mechanisms related to the anomaly detection.

4.4. Anomaly Detection

The method for anomaly detection applied in the demonstrator was explained in [1]. The utilized model predicts the short-term system responses and evaluates whether the occurring behaviour deviates from the prediction, thus detecting anomalies. For accurate predictions, the ML-based anomaly detection algorithm has to be initialized.

The required training data was obtained through randomizing the model inputs (wind speed, wind direction, and power constraint) for a larger time span and therefore operating in as many working points as possible, generating the desired variety of the system responses (see Fig. 6). The resulting behaviour was exported directly from OpenModelica as a csv-file and utilized to train the model for the anomaly detection.

4.5. Set-up

The simulation and the anomaly detection were performed on a desktop computer, which was connected to the used RPi via LAN. The GUI was executed on a RPi and displayed on the touchscreen panel mentioned in Sec. 4.2. An Arduino Due was connected to the RPi via USB and displayed the generated power and the wind speed on a TFT. Additionally, the GPIO pins of the Arduino were used to control the miniature wind turbines.

5. User Feedback, Conclusion and Outlook

The demonstrator was used during a training dedicated to engineers specialized in energy control systems. The participants were either experts in software development or in implementation of energy control systems. According to the participants' feedback, the demonstrator helped them in grasping the technology of ML and gathering first impressions regarding potential use cases, especially for analysing the payload of packages.

It was shown how the guidelines from Sec. 2 were implemented for a mobile demonstrator exemplifying an ML-based anomaly detection algorithm. In future, an established NIDS can be added to illustrate the difference between methods which consider the meaning of the payload and those which only involve statistics of the network traffic.

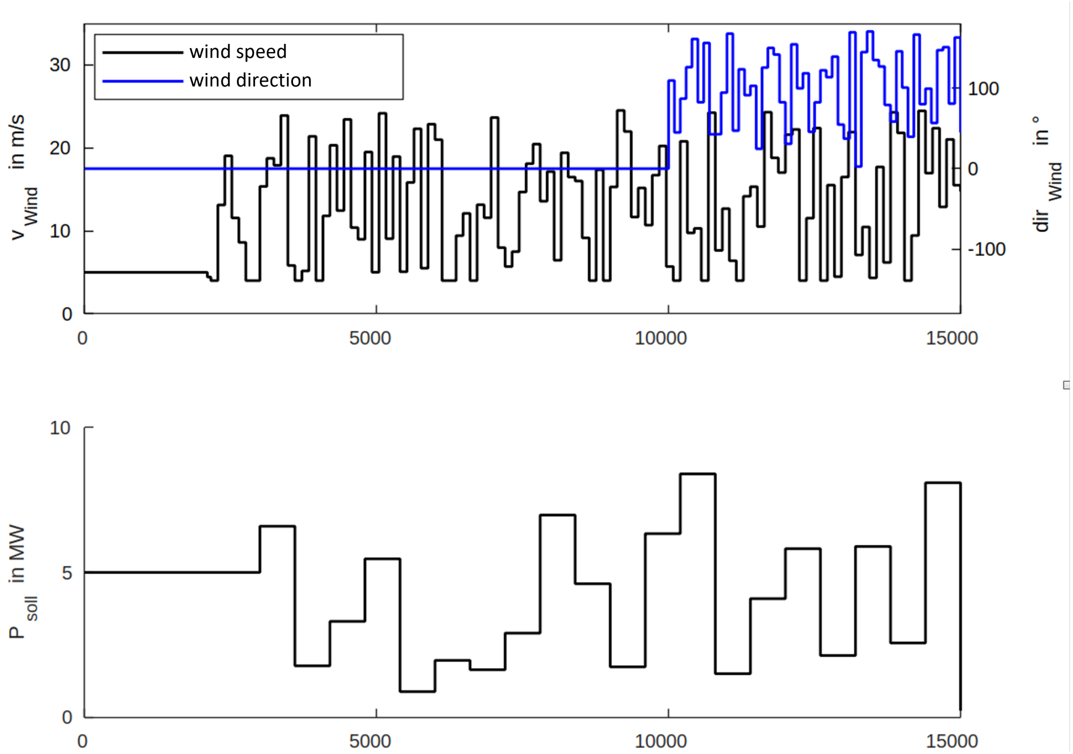


Fig. 6: Training phase of the anomaly detection model with time steps in seconds

Acknowledgement

This research has been partially funded by the German Federal Ministry of Education and Research in the framework of EnerSec (project number 16KIS0847). We would like to thank Max Klein and Adalbert Fono who contributed significant parts to the anomaly detection method.

References

[1] Max Klein, Gregor Thiele, Adalbert Fono, Niloufar Khorsandi, David Schade, and Jörg Krüger. Process data based anomaly detection in distributed energy generation using neural networks. In *2020 International Conference on Control, Automation and Diagnosis (ICCAD)*, pages 1–5. IEEE, 2020.

[2] Adalbert Fono, Gregor Thiele, Max Klein, and Jörg Krüger. Assistance method for the application-driven design of machine learning algorithms. In *Modern Materials and Manufacturing (MMM 2021)*, volume 1153 of *IOP Conf. Series: Materials Science and Engineerin*. IOP Publishing Ltd, 2021.

[3] Astrid Weyand, Stefan Seyfried, Benedikt Gosch, Daniel Fuhrländer-Völker, Niklas Panten, and Eberhard Abele. Method for transforming energy and production systems into mobile demonstrators. *Procedia Manufacturing*, 45:134–139, 2020.

[4] Felix Sieckmann, Natalie Petrusch, and Holger Kohl. Effectivity of learning factories to convey problem solving competencies. *Procedia Manufacturing*, 45:228–233, 2020. Learning Factories across the value chain – from innovation to service – The 10th Conference on Learning Factories 2020.

[5] Dario Assante, Clemente Capasso, and Ottorino Veneri. Internet of energy training through remote laboratory demonstrator. *Technologies*, 7(3), 2019.

[6] M. Tisch, C. Hertle, J. Cachay, E. Abele, J. Metternich, and R. Teberg. A systematic approach on developing action-oriented, competencybased learning factories. *Elsevier*, pages 1–7, 2013.

[7] Philip Eberhart, Tek Shan Chung, Anton Haumer, and Christian Kral. Open Source Library for the Simulation of Wind Power Plants. `https://2015.international.conference.modelica.org/proceedings/html/submissions/ecp15118929_EberhartChungHaumerKral.pdf`, 2015.

[8] Christoffer Fors Johansson. Openmodelica interactive simulation using an opc ua client. Bachelor's thesis, Linköping University, Department of Computer and Information Science, 2017.