A Comparative Study and a New Industrial Platform for Decentralized Anomaly Detection Using Machine Learning Algorithms

Fabian Gerz
Cologne Lab of Artificial Intelligence and Smart Automation (CAISA)
TH Köln
Köln, Germany
fabian.gerz@th-koeln.de

Tolga Renan Bastürk
Cologne Lab of Artificial Intelligence and Smart Automation (CAISA)
TH Köln
Köln, Germany
tolga.bastuerk@outlook.com

Julian Kirchhoff
Cologne Lab of Artificial Intelligence and Smart Automation (CAISA)
TH Köln
Köln, Germany
julian.kirchhoff@icloud.com

Joachim Denker
Department of Research and Innovation
ASINCO GmbH
Duisburg, Germany
joachim.denker@asinco.de

Loui Al-Shrouf
MDZ RHEINLAND – Rhineland Digital Centre for SMEs
Gottfried-Hagen-Str. 62
51105 Köln
loui@digital-rheinland.de

Mohieddine Jelali
Cologne Lab of Artificial Intelligence and Smart Automation (CAISA)
TH Köln
Köln, Germany
mohieddine.jelali@th-koeln.de

***Abstract*—The occurrence of anomalies and unexpected, process-related faults is a major problem for manufacturing systems, which has a significant impact on product quality. Early detection of anomalies is therefore of central importance in order to create sufficient room for maneuver to take countermeasures and ensure product quality. This paper investigates the performance of machine learning (ML) algorithms for anomaly detection in sensor data streams. For this purpose, the performance of six ML algorithms (K-means, DBSCAN, Isolation Forest, OCSVM, LSTM-Network, and DeepAnt) is evaluated based on defined performance metrics. These methods are benchmarked on publicly available datasets, own synthetic datasets, and novel industrial datasets. The latter include radar sensor datasets from a hot rolling mill. Research results show a high detection performance of K-means algorithm, DBSCAN algorithm and LSTM network for punctual, collective and contextual anomalies. A decentralized strategy for (real-time) anomaly detection using sensor data streams is proposed and an industrial (Cloud-Edge Computing) platform is developed and implemented for this purpose.**

***Keywords*—Data streams, radar sensors, anomaly detection, IIoT platform, cloud-edge computing**

I. INTRODUCTION

The detection of faults, such as sensor failures or wear problems, holds great potential with regard to process optimization in the manufacturing industry. The occurrence of anomalies and unexpected, process-related faults is a major problem for manufacturing systems, which has a significant impact on product quality. Early detection of anomalies is therefore of central importance in order to create sufficient room for maneuver to take countermeasures and ensure product quality. Accurate detection of anomalies is a particular challenge in dynamically changing data streams.

Anomaly detection (or outlier detection) has been the topic of a number of surveys and review articles, as well as books. A comprehensive literature review on anomaly detection, analysis and prediction techniques is given in [1]. The authors of [2] extend these insights by applying anomaly detection to graphs for the identification of anomalous structures and present a comprehensive survey of deep learning methods. Recent anomaly detection survey papers [3], [4] were published in 2019 and 2020, respectively. Reference [5] provides a survey of deep learning-based anomaly detection techniques developed in recent years. Meng *et al.* [6] propose a traditional anomaly detection taxonomy that includes methods based on classification, clustering, distance, density and statistics. Munir *et al.* [7] analyzed and compared different (traditional and deep learning-based) anomaly detection methods on streaming datasets. Domingues *et al.* [8] survey unsupervised ML algorithms in the context of outlier detection. The recent books [9], [10] present the latest methods for outlier detection.

Other surveys and comparative studies of anomaly detection techniques focused on specific application areas are given, e.g., in [11] (manufacturing systems), [12] (internet of things (IoT) networks), [13] (maritime video surveillance), [14] (IP multimedia subsystems), [15] (core router systems), [16], [17] (network traffic attacks), and [18] (smart city wireless sensor networks). Furthermore, several works [19], [20], [21] investigate various implementation approaches for anomaly detection in IIoT systems and using federated learning [22] or deep learning [23], [24].

Fahim and Sillitti [1] state many research gaps and challenges in the research on the anomalous behavior, particularly:

- Formalization of ways *"to access data logs and sensory data streams, to build a model and validate it in real-life settings"*.
- Missing investigation of new anomaly detection model for industrial case studies.

This paper provides a promising contribution to closing these gaps. In addition, one goal is to develop a decentralized (Industrial IoT(IIoT)/Edge Cloud Computing) platform for anomaly detection that covers different algorithms and enables modular extensibility of the methods as well as easy scalability. Furthermore, the framework for concept drift detection is integrated in this platform and will be extended in future work. Moreover, a real-world case study from a hot strip mill is considered, where radar sensors are installed to realize a radar-based width measurement system of hot strips. Here, the radar sensor data were preprocessed through new detection methods presented in this paper.

978-1-7281-8671-9/22/$31.00 ©2022 IEEE

The remainder of this paper is organized as follows. The anomaly detection methods investigated in this paper are briefly described in Section II. Section III presents and discusses the comparative study and its results. The industrial platform developed for anomaly detection based on ML algorithms using sensor data streams is described in Section IV. Section V is devoted to the industrial real-world case study. Lastly, Section VI presents conclusions and future work.

II. INVESTIGATED ANOMALY DETECTION TECHNIQUES

Lai, Zhang, and Liu [25] categorize anomaly detection methods in *knowledge-based*, *statistics-based*, and *ML-based* methods. In this paper, the focus is on the latter methods. A variety of ML-based methods can be used to detect anomalies. These differ primarily in the learning task of the underlying algorithm. A further distinction can be made based on the complexity of the mathematical formulation and the implementation of the methods; see Table I.

A. Machine Learning-based Methods

This study includes the following ML-based anomaly detection techniques:

1) K-means Clustering: One of the well-known (centroid-based) clustering algorithms for anomaly detection is using K-means clustering [26]. Clustering is an unsupervised learning process, thus clustering-based anomaly detection does not require fully-labeled dataset, which is difficult to obtain in many cases.

2) Density-based Spatial Clustering of Applications with Noise (DBSCAN): DBSCAN [27] is a density-based and unsupervised ML algorithm that classifies the data points into three different categories: core points, border points, and anomalies. In contrast to K-means, DBSCAN usually works well for noisy datasets and does not require the number of clusters as an input parameter; clusters can take any irregular shape.

3) Isolation Forest (IForest): This ML approach introduced by Liu *et al.* [28] to detect anomalies in time-series using a sliding window is purely based on the concept of binary trees isolating data points without employing any distance or density measure.

4) One-Class Support Vector Machines (OCSVM): OCSVM is used for detecting anomalies in time-series data by projecting time-series data vectors on to phase spaces [29]. Such a classification-based techniques does not require prior knowledge of the underlying data distribution.

5) Long Short Term Memory (LSTM) Networks: LSTM networks, developed by Hochreiter and Schmidhuber [30], belong to the recurrent neural networks (RNN) architectures, which have a feedback connection enabling them to use the output information for the next input of the sequence. The use of LSTM networks for anomaly detction is proposed in [31].

6) Deep Learning-based Anomaly Detection Approach (DeepAnt): Munir et al. [32] proposed deep convolutional neural networks (CNN) to forecast time-series and detect anomalies based on the error of the prediction. Deep learning neural networks have become very popular in the last decade, as they can model much more complex non-linear relationships than a shallow neural networks models.

TABLE I. OVERVIEW OF THE SELECTED ANOMALY DETECTION METHODS

Method	Learning task	Model	Complexity
K-means	Cluster analysis	Cluster controid	Low
DBSCAN	Cluster analysis	Dense regions	Low
IForest	Cluster analysis	Ensemble of decision trees	Medium
OCSVM	Classification	Support vector machine	Medium
LSTM	Regression	Recurrent neural network	High
DeepAnt	Regression	Convolutional neural network	High

III. COMAPARATIVE STUDY: RESULTS AND ANALYSIS

A. Benchmark Datasets

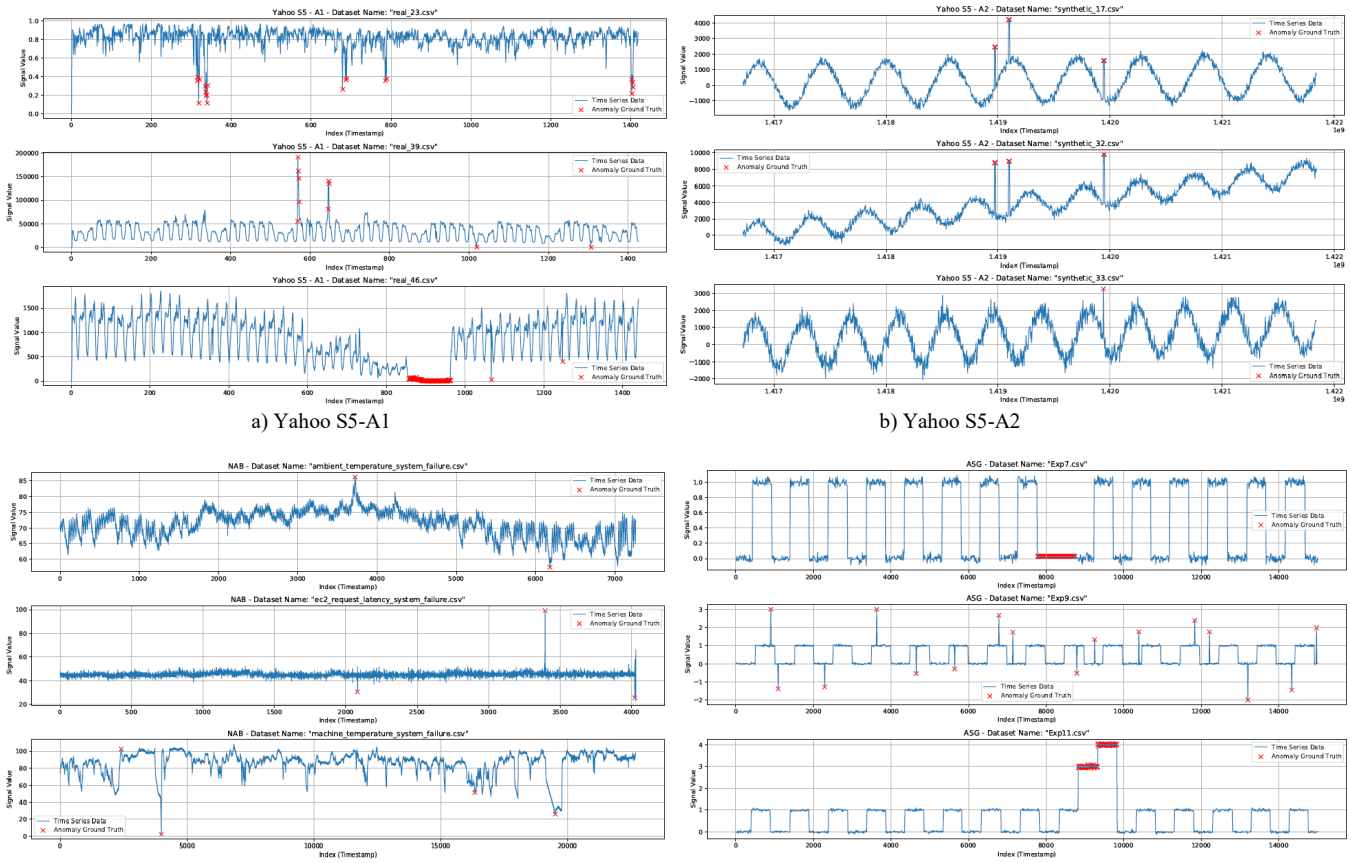
Three datasets have been considered to evaluate the performance of the selected anomaly detection methods described in Section II (some sample time-series are shown in Fig. 1):

1) The Yahoo S5 anomaly detection dataset: The dataset consists of a total of four separate datasets that contain real or synthetically generated data from the network traffic of the web service provider Yahoo [33]. The datasets consist of a large number of univariate time series that are labeled for performance investigation (or benchmarking) of anomaly detection methods. These datasets contain primarily point anomalies.

2) The Numenta anomaly benchmark (NAB) dataset: This consists of a variety of real and synthetic time series provided specifically for the study of anomaly detection methods by the company Numenta [34], [35]. The majority of the data it contains are univariate time series from real-world use cases. A total of three real and three synthetic time series are selected from the NAB dataset for the comparative study. The datasets contain point and collective anomalies.

3) Dataset from an artificial signal generator (ASG): In order to support the evaluation of the algorithms with own datasets, artificial anomalous time series are generated. For this purpose, the CODESYS implementation of the Artificial Signal Generator (ASG) by Kirchhoff [36] is used. The main advantage of artificially generated time series is that, in contrast to real time series, all occurring anomalies are correctly labeled. As a result, errors due to incorrect class labels can be excluded.

The ASG is used to generate and record synthetic time series during runtime in 18 experiments. To meet the requirement for plausibility of the artificially generated time series, sine, square and sawtooth waveforms are selected as base signals, respectively. These often occur in industrial applications and can be traced back to real voltage waveforms of sensor signals. The basic signals are then superimposed with white noise as well as with point, collective or contextual anomalies. Finally, the anomalous signals are sampled at 100 Hz. From the acquired data, a sequence of a total of 1500 contiguous data points is extracted for each experiment and stored as a time series. Finally, all 18 time series are combined into the ASG dataset.



a) Yahoo S5-A1 b) Yahoo S5-A2

c) NAB d) ASG

Fig. 1. Sample time-series from Yahoo Webscope (a, b), NAB (c) and ASG (d) datasets

B. Parameter Settings

All parameters of the compared approaches are set either according to the default setting or by trial-and-error to get an overall satisfactory performance; see Table II.

TABLE II. HYPERPARAMETERS USED FOR THE ANOMALY DETECTION ALGORITHMS

Model	Hyperparameter*	Value
K-means	init	random
	n_clusters	1
	max_iter	100
	random_state	0
DBSCAN	min_samples	20
	leaf_size	0
	p	0
	n_jobs	0
	ε	0.05
	window_size	8
IForest	number of estimators	25
	contamination	0.075
	max_samples	Auto
	max_features	1
	bootstrap	False
	random_state	0
OCSVM	degree	2
	max_iter	500
	upper bound of errors nu	0.01
LSTM	number of nodes	50
	number of hidden layer	2
	number of units in dense layer	1
	batch_size	64
	n_features	1
	activation function	relu
	optimizer	adam
	loss function	mean absolute error
	dropout rate	0.2
	number of epochs	20
DeepAnt	number of input nodes w	8
	number of output nodes pw	1
	number of filters (kernels)	32
	n_features	1
	activation function	relu
	optimizer	adam
	loss function	mean absolute error
	dropout rate	0.5
	number of epochs	20

* Hyperparameters not mentioned refer to the corresponding default values.

C. Performance Measures

A common quality criterion for measuring the performance of a classification is the F1-score, defined as:

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall} = \frac{TP}{TP + \frac{FN+FP}{2}}, \quad (1)$$

where TP is the number of true positives, FP the number of false positives, and FN the number of false negatives. F1-score should be as high as possible, perfectly unity (1.0). This evaluation metric is particularly suitable for use cases that have an unbalanced class distribution, as it is generally the case for anomaly detection applications [37]. For this reason, the F1 score is preferred over the commonly used Accuracy metric for performance evaluation of ML algorithms.

Moreover, the model quality is judged by finding the receiver operating characteristic (ROC) curve, which indicates the relation between the false positive rate and the true positive rate as the threshold is changed. A practical metric to quantify the information provided by the ROC curve is the area under the curve (AUC) [38]. The AUC is higher for the ROC curves approaching the perfect classifier.

D. Results and Discussion

The performances of the algorithms on all the datasets are presented by using the scores F1 and AUC in Table III, as well as the ROC graphs in Fig. 2.

TABLE III. EVALUATION METRICS FOR THE INVESTIGATED ALGORITHMS (THE BEST THREE ARE MARKED IN BOLD.)

	Yahoo S5-A1		Yahoo S5-A2		NAB		ASG	
Method	**F1**	**AUC**	**F1**	**AUC**	**F1**	**AUC**	**F1**	**AUC**
K-means	**0.941**	**0.981**	**1.000**	**1.000**	0.530	0.740	**0.899**	**0.945**
DBSCAN	**0.894**	**0.966**	**1.000**	**1.000**	**0.659**	0.686	**0.839**	0.845
IForest	0.884	0.947	0.836	0.983	**0.585**	**0.831**	0.688	0.837
OCSVM	0.679	0.896	0.435	0.815	0.303	0.711	0.591	0.773
LSTM	**0.894**	**0.974**	**1.000**	**1.000**	**0.570**	**0.839**	**0.888**	**0.953**
DeepAnt	0.803	0.952	0.897	0.998	0.448	**0.796**	0.801	**0.866**

For the Yahoo S5-A1 dataset, good results (AUC ≥ 95%) are obtained with most algorithms. The ROC curves of K-means, DBSCAN, and LSTM are very close to each other, with the K-means algorithm standing out with a slightly better result. Since the ML methods can infer the inner context of the examined time series, good results are obtained with all three ML algorithms in the detection of punctual anomalies. The highest F1-score of 0.941 is achieved by K-means, the highest AUC of 0.981 as well.

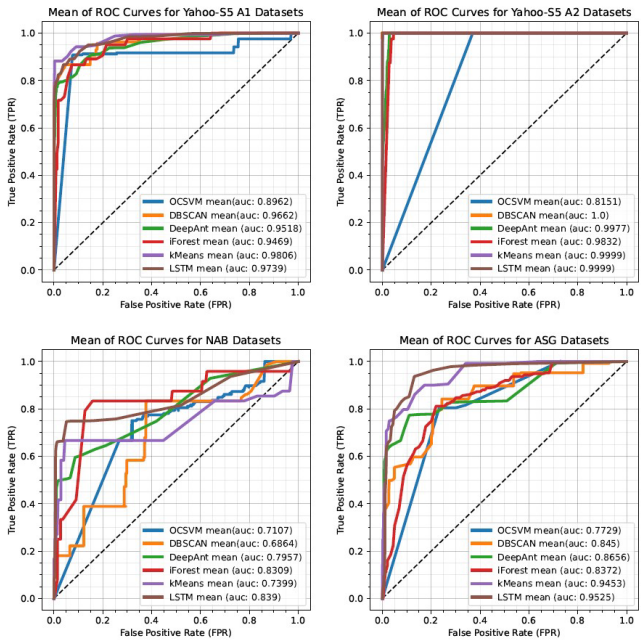


Fig. 2. ROC curves for the investigated anomaly detection methods

The evaluation results for the Yahoo S5-A2 dataset turn out very well for all algorithms, except OCSVM. This is probably due to the fact that the dataset contains only synthetically generated time series and point anomalies. As a result, three algorithms (K-means, DBSCAN, and LSTM) achieve an F1-score of 1.0 and an AUC score of 1.0.

The NAB dataset has a small number of positive class labels (anomalies). This directly affects the evaluation result, since the class labels are an essential basis for calculating the evaluation metrics. Nevertheless, it can be seen in the test results that the selected methods have a good performance. The F1-scores of DBSCAN, Isolation Forest, and LSTM are 0.659, 0.585, and 0.570, respectively. The AUC scores suggest Isolation Forest (0.831), DeepAnt (0.796), and LSTM (0.839) as the best performing algorithms.

The ASG dataset poses the greatest challenge to anomaly detection methods. While K-means and LSTM allow the detection of collective and contextual anomalies, the IForest algorithm reaches its limits here. Due to the basic idea of isolation, contextual anomalies cannot be detected, so that when examining the ASG dataset, IForest achieves an F1-score that is about 20% lower compared to K-means, DBSCAN, and LSTM.

In total, the results demonstrate that the algorithms DBSCAN, K-means, and LSTM outperform the other algorithms. DBSCAN is preferred over K-means because DBSCAN is more robust against noise and irregularities in the shape of clusters.

Based on these results, the K-means and DBSCAN algorithms are selected for implementation at the edge, and the LSTM and DeepAnt for implementation at the cloud. The reason for separating the algorithms into different levels of automation is due to the level of complexity / computation time of the learning procedures (computation time for training or execution). Further details on the implementation of the algorithms and the realization of the overall IIoT system are discussed in Section IV.

IV. INDUSTRIAL PLATFORM FOR DECENTRALIZED ANOMALY DETECTION

Within IIoT, more and more devices have been joined together and produce massive industrial data every day, which requires powerful computing resources. When these data are transmitted to the cloud, network latency and bandwidth become a bottleneck. [39]

To overcome these problems, a decentralized strategy for (real-time) anomaly detection using sensor data streams is proposed and an industrial platform has been developed and implemented for this purpose. Some necessary computing tasks / algorithms are made close to the machine, i.e., in the sensor or in the edge. Other tasks / algorithms are performed or results are delivered and stored in the cloud center. In this way, workloads of the cloud center can be reduced massively.

For this purpose, the design of a suitable system architecture is required. Furthermore, a large number of software components are needed to implement the functional scope of the overall system. The development of the associated application levels, interfaces, and software modules are presented in this section.

A. Design of the Overall System

The design of the overall system is carried out considering the following requirements placed on the IIoT platform:

- Interfaces to all relevant automation levels.
- Hierarchy design of an IIoT platform.
- Modular expandability of algorithms.
- Scalability.
- Validation of anomaly detection against simulation environment.

- User-friendly application and maintenance.

The system design follows the topology of an IIoT platform with edge-cloud computing architecture. Accordingly, it is necessary to couple a local system level with a superordinate cloud system via interfaces. The fusion of both layers couples the advantages of edge computing of low latency and real-time capability as well as high data-transfer rates [40] with the advantages of cloud computing [41] in the form of scalable computing capacities and location-independent data access. The overall structure of the platform is schematically illustrated in Fig. 3, which is divided into three basic levels: the field level, the control level, and the supervisory level.

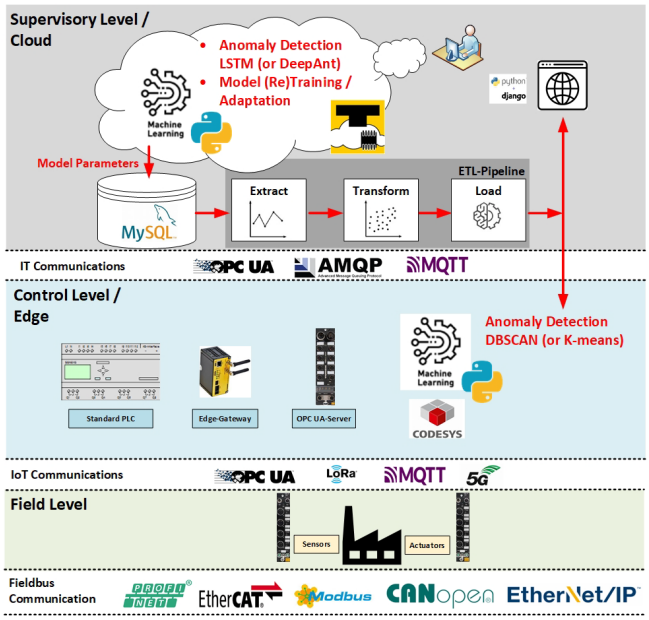


Fig. 3 Structure and functioning of the IIoT platform for decentralized anomaly detection in data streams

In the edge controller, measurement data are recorded, analyzed (preprocessing, feature extraction), and cyclically transferred to the cloud. The primary task of the cloud is the storage of structured data from the sensors, parameters for the algorithms, and status parameters for control. Another task of this level is the training process for the algorithms. Due to the flexible computing capacity, the algorithms are trained decentrally in the cloud and their parameters are communicated to the edge layer as a result. Finally, the overall system is completed by a human machine interface (HMI) application, allowing the user to intervene in the entire process. Thus, the user is able to observe current measured values of the sensors as well as predictions of the models. In addition, the model parameters can be modified manually or the training process be initialized to optimize the parameters.

B. Anomaly Detection at Edge

In the context of the automation pyramid, the edge controller represents an interface to the field level and the cloud. The goal is to move the measurement data analysis as close as possible to the node of measurement data collection. In this way, critical measurement data can be analyzed in a timely manner.

For implementing the algorithms in the edge controller, a database library and an algorithm library were developed. The database library enables the communication with a MySQL database and is highlighted in [42]. In this application, it is first used for cyclic updating of the model parameters, signal parameters, and the control flanks. The result of the measurement data collection as well as analysis by the algorithms is finally transferred to the MySQL database.

In addition to data acquisition, data preprocessing and anomaly detection are further functionalities of the edge controller (i.e. the preprocessing and anomaly detection algorithms are implemented in the edge). Since K-means and DBSCAN show promosing results in the early development stages (see Section III.D), both algorithms were implemented in the edge controller. The anomaly detection is divided into the training and testing components. In this case, the test component corresponds to the implementation of the algorithm in the edge controller. The model (re)training is outsourced to the cloud, the result of which represents the kernel points of the normal cluster in the feature space. Both the sequence length parameter and the kernel points are transferred as parameters to the edge controller. Using a control variable, the parameters can be transferred from the database to the edge controller either at the beginning of the runtime or when needed again. According to the definitions of the DBSCAN algorithm, each new data instance counts to the cluster of core points, as far as they are density reachable to one of the core points. In the first step, the features are calculated from the last sequence using a feature extractor.

C. Anamoly Detection at Cloud

The cloud system is primarily used for data backup as well as long-term data analytics. In this context, the ML models are (re)trained for bigger data amounts. As an interface to the edge controller, the parameters of the models are communicated, whereas control signals for training the algorithms are transmitted from the HMI application to the cloud. For the structure of the database, reference is made to [42].

Short-term prediction of anomalies is implemented close to the sensors at the edge. To achieve a long-term overview of the data, analysis as well as prediction is embedded in the cloud system characterized by flexible computing capacity. Thus, even complex (deep) learning models / algorithms can be used for prediction or anomaly detection, here LSTM (or DeepAnt) in a “confirmation” stage. A data pipeline along the lines of an ETL (extraction-transformation-loading) pipeline provides a batch of the last recorded measurement data from the first step. In addition, the current model parameters and scaling factors are transferred. In the next step, the data are transformed into the form suited for the application in the model. In the case of the DeepAnt model, the time series are divided into sequences as well as their target value. Then, the current model is used to make predictions over a subset from a specific time interval (window) of the data stream. These are communicated to the database to provide them to the user as information of the model prediction quality.

To ensure robust anomaly detection despite process changes, the predictions have to be analyzed in the context of a concept drift detection. The purpose of the detection is to find out if the distribution of the signal has changed. In this paper, the platform has been prepared for implementing concept drift detection and adaptation methods, but implementation is still a work in progress.

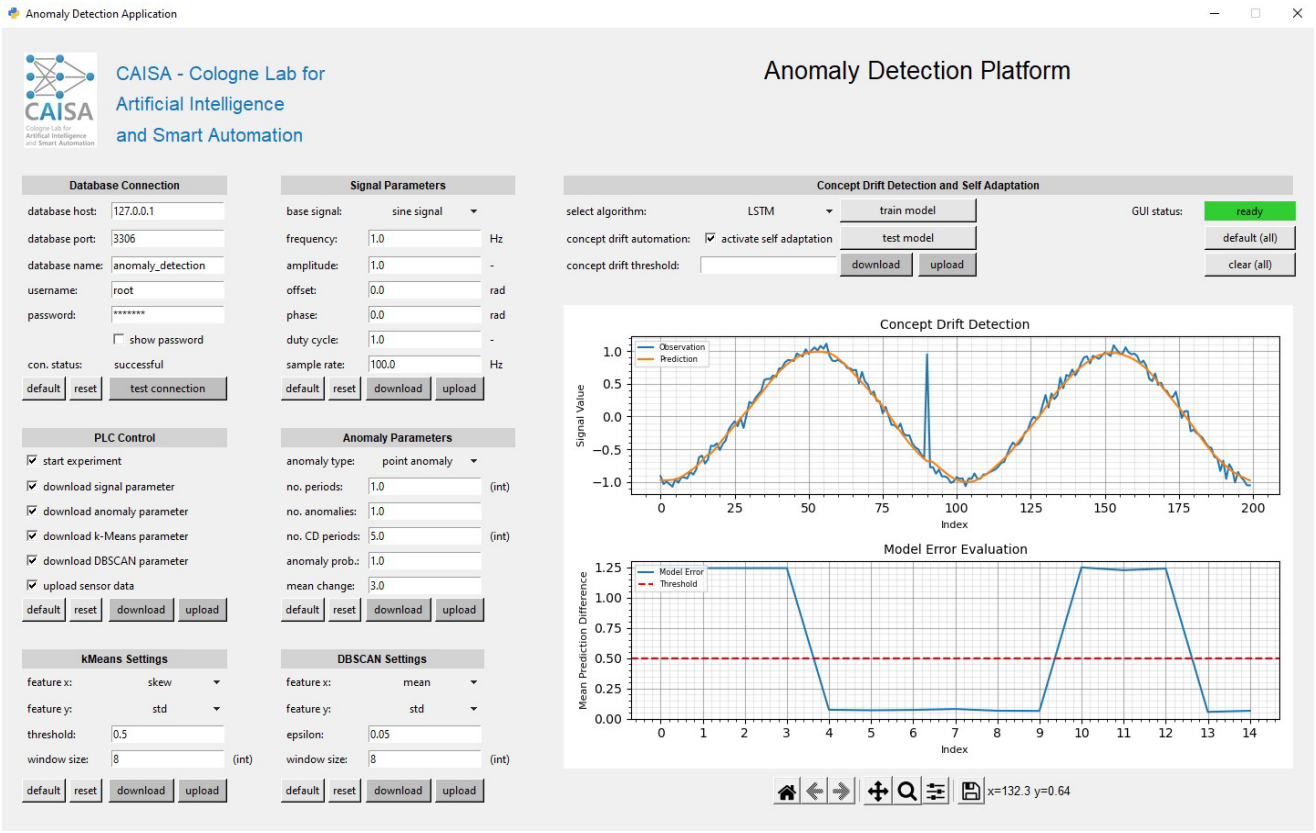


Fig. 4 User interface of the developed platform

D. *Supervisory Level*

The third component of the overall system is the HMI application in the supervisory level, the user's interface to the cloud system. It allows setting control signals and monitoring the acquired signals as well as the related predicted signal curves. The user interface (Fig. 4) is divided into different subsystems. In the left part of the user interface, the user can start the database communication. In addition, the signal and anomaly parameters can be adjusted. If the system has to be (re)started, there are checkboxes that update the control signals in the database. If it is necessary to change the anomaly detection parameters, they can be adjusted for both the DBSCAN (or K-means) and LSTM (or DeepAnt) algorithms. There are also two graphs on the right side prepared for concept drift detection.

V. Application to Real-World Datasets: Radar Sensor Data from Hot Strip Mill

Flat steel production is a multistage process including steel making, continuous casting, hot and cold rolling stages. Today's hot rolling mills handle heavier loads and oprerate with faster velocities than ever before, leading to high temperatures, pressures, and degradation [43]. Fault and anomaly detection is thus very important to ensure highest product quality and highest production rates.

Due to the extreme working conditions in metal production, essential measurement technologies, e.g., optical and laser-based, reach their physical limits. Thus, observation of relevant process parameters stays fragmented over the whole process chain. Still, hot strip production requires perceptible strip width, which directly lead to increased material, raw material, and energy consumption. Only with a precise width measurement, it is possible to perform an exact pre-calculation or adjustment of the pre-strip width.

Radar-based measurement systems are inherently insensitive to harsh conditions (e.g., high temperature, dust, air humidity, mist from rolling emulsion or oil) and thus highly suitable for recording necessary measurement variables with sufficient precision. Due to advancement in innovative signal processing algorithms, this will be achieved without any discernible disadvantages in the future compared to existing measurement technologies for material detection and tracking, width measurement and other geometric property measurements for quality assurance during production, as proven in [44].

Figure 5 shows typical (normalized) radar sensor data acquired in the commissioning phase of a newly developed radar-based width measurement system for strip in front of a roughing (hot strip) mill. At that stage, temporary sensor failures often occurred, due to harsh working conditions (strong steam development, high temperature, etc.), strip edge texture, and strip deformations, etc., leading to anomalies in the data streams (distance measurement data). Exemplarily, different faulty and non-faulty behaviors are depicted and examined via the different algorithms.

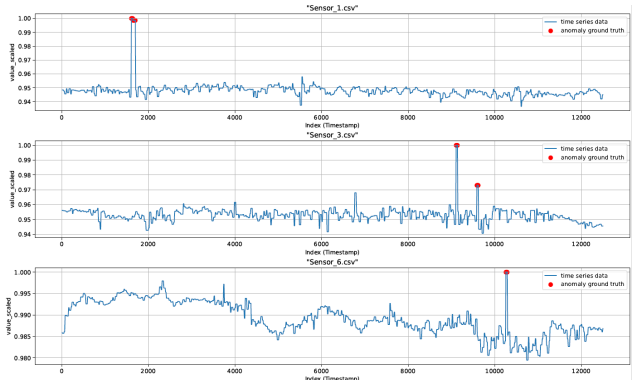


Fig. 5 Sample data from radar datasets

Figure 6 shows the ROC curves obtained for the examined methods and radar datasets. It can be ascertained that IForest and OCSVM provide the best performance; K-means, DeepAnt, and LSTM are also good enough and show an increasing tendency for larger datasets; DBSCAN gives the worst performance in this case study. This practical example shows that the selection of a suitable method cannot be determined in a generalized way, as this result contrasts with the results of the Comparitve Study in Section III. Therefore, a differentiated consideration is necessary depending on the application area, the size of the data set and the expected type of anomalies, in order to obtain an optimal result. In principle, it is advisable to perform tests of different methods and make an assessment with regard to their accuracy, complexity and model speed.

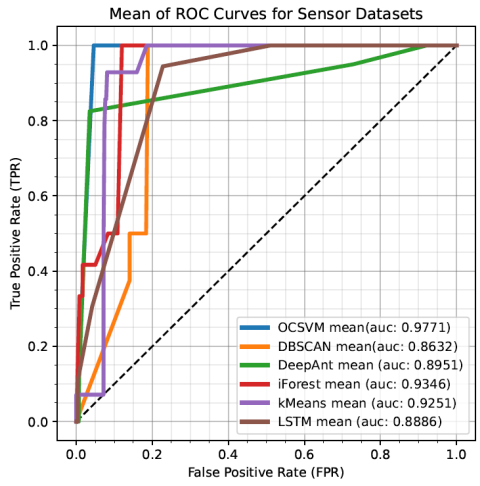


Fig. 6. ROC curves for the radar datasets

VI. Conclusions and Future Work

This paper investigates the performance of ML algorithms with respect to anomaly detection in sensor data streams. A comparative study of six ML algorithms (K-means, DBSCAN, Isolation Forest, OCSVM, LSTM-Network, and DeepAnt) has been performed on publicly available datasets, own synthetic datasets, and novel industrial datasets. The results revealed a high detection performance of K-means algorithm, DBSCAN algorithm and LSTM network for punctual, collective and contextual anomalies. Moreover, a decentralized strategy for (real-time) anomaly detection using sensor data streams was proposed and an industrial (Cloud-Edge Computing) platform was developed and implemented for this purpose.

The platform developed here does not claim to include state of the art, but compare methods of different complexity levels, and will be extended by further anomaly detection methods in the next step. Especially, the investigation of deep learning methods like different transformer variants [45], [46], but also the consideration of methods like Xgboost [47], [48] offer further insights in the field of anomaly detection. Furthermore, concept drift represents a central challenge, which will be addressed by the implementation and evaluation of concept drift detection algorithms. In addition, the implementation and evaluation of the platform in real industrial factories are targeted.

Moreover, the ML algorithms for anomaly detection need to be analyzed in more detail and compared to other hybrid approaches for anomaly detection under drifting concepts, such as those recently developed in [49], [50].

References

[1] M. Fahim and A. Sillitti, “Anomaly detection, analysis and prediction techniques in IoT environment: A systematic literature review," in IEEE Access, vol. 7, 2019, pp. 81664–81681.

[2] X. Ma, J. Wu, S. Xue, J. Yang, C. Zhou, Q.Z. Sheng, H. Xiong, and L. Akoglu, “A comprehensive survey on graph anomaly detection with deep learning,” IEEE Transactions on Knowledge and Data Engineering, 2021.

[3] H. Wang, M. J. Bah, and M. Hammad, “Progress in outlier detection techniques: A survey,” IEEE Access, vol. 7, pp. 107964–108000, 2019.

[4] M. Braei, and S. Wagner, “Anomaly Detection in Univariate Time-series: A Survey on the State-of-the-Art,“ ArXiv abs/2004.00433, 2020.

[5] R. Chalapath and S. Chawla, “Deep learning for anomaly detection: A survey,” eprint arXiv:1901.03407, 2019.

[6] F. Meng, G. Yuan, S. Lv, Z. Wang, and S. Xia, “An overview on trajectory outlier detection,” Artificial Intelligence Review, vol. 52, no. 4, pp. 2437–2456, 2019.

[7] M. Munir, M. A. Chattha, and S. Ahmed, “A comparative analysis of traditional and deep learning-based anomaly detection methods for streaming Data,” in 18th IEEE International Conference on Machine Learning and Applications (ICMLA), 2019, pp. 561–566.

[8] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, “A comparative evaluation of outlier detection algorithms: Experiments and analyses,“ Pattern Recognition, vol. 74, pp. 406–421, 2018.

[9] N. N. R. Ranga Suri, Narasimha Murty M, G. Athithan, Outlier Detection: Techniques and Applications. Springer, 2019.

[10] X. Wang, X. Wang, and M. Wilkes, New Developments in Unsupervised Outlier Detection. Springer, 2021.

[11] K. Zope, K. Singh, S. H. Nistala, A. Basak, P. Rathore, and V. Runkana, “Anomaly detection and diagnosis in manufacturing systems: A comparative study of statistical, machine learning and deep learning techniques,” in Annual Conference of the PHM Society, vol. 11, no. 1, 2019.

[12] A. Diro, N. Chilamkurti, V.-D. Nguyen, and W. Heyne, “A comprehensive study of anomaly detection schemes in IoT networks using machine learning algorithms,“ Sensors, vol. 21, no. 24, 8320, 2021.

[13] B. Auslander, K. M. Gupta, and D. W. Aha, “A comparative evaluation of anomaly detection algorithms for maritime video surveillance,” in

Proc. of the Society of Photographic Instrumentation Engineers Conference Orlando, FL, 2011.

[14] M. A. Akbar, Z. Tariq, and M. Farooq, “A comparative study of anomaly detection algorithms for detection of SIP flooding in IMS,“ in 2nd International Conference on Internet Multimedia Services Architecture and Applications, 2008, pp. 1–6.

[15] S. Jin, Z. Zhang, K. Chakrabarty, and X. Gu, Anomaly-Detection and Health-Analysis Techniques for Core Router Systems. Springer, 2020.

[16] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, Network Traffic Anomaly Detection and Prevention. Springer, 2017.

[17] D. Kwon, K. Natarajan, S. C. Suh, H. Kim, and J. Kim, “An empirical study on network anomaly detection using convolutional neural networks,“ in IEEE 38th International Conference on Distributed Computing Systems (ICDCS), 2018.

[18] V. Garcia-Font, C. Garrigues, and H. Rifà-Pous, “A comparative study of anomaly detection techniques for smart city wireless sensor networks,“ Sensors, vol. 16, no. 6, 868, 2016.

[19] B. Sharma, L. Sharma and C. Lal, "Anomaly Detection Techniques using Deep Learning in IoT: A Survey," 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), 2019, pp. 146–149.

[20] Y. Wu, H. -N. Dai and H. Tang, "Graph Neural Networks for Anomaly Detection in Industrial Internet of Things," in IEEE Internet of Things Journal, 2021.

[21] G.E.I. Selim, E.E.-D. Hemdan, A.M. Shehata, N.A. El-Fishawy, “Anomaly events classification and detection system in critical industrial internet of things infrastructure using machine learning algorithms,” Multimedia Tools and Applications 80, 12619–12640, 2021.

[22] Y. Liu, S. Garg, J. Nie, Y. Zhang, Z. Xiong, J. Kang, and M.S. Hossain, "Deep Anomaly Detection for Time-Series Data in Industrial IoT: A Communication-Efficient On-Device Federated Learning Approach," in IEEE Internet of Things Journal, vol. 8, no. 8, pp. 6348–6358, April 15, 2021.

[23] D. Utomo and P. -A. Hsiung, "Anomaly Detection at the IoT Edge using Deep Learning," 2019 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW), 2019, pp. 1–2.

[24] D. Wu, Z. Jiang, X. Xie, X. Wei, W. Yu and R. Li, "LSTM Learning With Bayesian and Gaussian Processing for Anomaly Detection in Industrial IoT," in IEEE Transactions on Industrial Informatics, vol. 16, no. 8, pp. 5244–5253, Aug. 2020.

[25] Y. Lai , J. Zhang, and Z. Liu, “Industrial anomaly detection and attack classification method based on convolutional neural network,” Hindawi Security and Communication Networks, vol. 2019, Article ID 8124254.

[26] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1: Statistics, Berkeley, Calif., 1967, University of California Press, pp. 281–297.

[27] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in Proc. 2nd International Conference on Knowledge Discovery and Data Mining, KDD’96, AAAI Press, 1996, pp. 226–231.

[28] F. T. Liu, K. M. Ting, and Z. h. Zhou, “Isolation forest,” in ICDM’08: Proc. 8th IEEE International Conference on Data Mining, IEEE Computer Society, 2008, pp. 413–422.

[29] M. Junshui and S. Perkins, “Time-series novelty detection using one-class support vector machines,” in Proc. of the International Joint Conference on Neural Networks, 2003, August 26, Portland, OR, USA.

[30] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[31] T. Ergen, A. H. Mirza, and S. S. Kozat, “Unsupervised and Semi-supervised Anomaly Detection with LSTM Neural Networks,” arXiv preprint arXiv:1710.09207, 2017.

[32] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed. DeepAnT, “A deep learning approach for unsupervised anomaly detection in time series,” IEEE Access, vol. 7, pp. 1991–2005, 2019.

[33] Yahoo! Webscope Research, “S5 - A Labeled Anomaly Detection Dataset, version 1.0(16M)”. https://webscope.-sandbox.yahoo.com/catalog.php?datatype=s&did=70

[34] Numenta, “Numenta Anomaly Benchmark (NAB) – The first benchmark for evaluating anomaly detection in streaming data,” 2021. https://numenta.com/machine-intelligence-technology/numenta-anomaly-benchmark/

[35] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, “Unsupervised real-time anomaly detection for streaming data,” Neurocomputing, vol. 262, pp. 134–147, 2017.

[36] J. Kirchhoff, Entwicklung und Integration einer Anomalieerkennung in Datenströmen auf Basis des maschinellen Lernens in einer Cloud-IIoT Infrastruktur. Master Thesis, TH Köln, 2021, unpublished.

[37] M. Sokolova, N. Japkowicz, S. Szpakowicz, “Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation,” in Sattar, A., Kang, Bh. (eds) AI 2006: Advances in Artificial Intelligence. AI 2006. Lecture Notes in Computer Science(), vol 4304. Springer, Berlin, Heidelberg, 2006.

[38] C. X. Ling, J. Huang, and H. Zhang, “AUC: A better measure than accuracy in comparing learning algorithms,” in Proc. Advances in Artificial Intelligence, 16th Conference of the Canadian Society for Computational Studies of Intelligence, AI 2003, Halifax, Canada, June 11–13, 2003, pp. 329–341.

[39] C. Wang, B. Wang, H. Liu, and H. Qu, “Anomaly detection for industrial control system based on autoencoder neural network,” Wireless Communications and Mobile Computing, vol. 2020, Article ID 8897926, 2020.

[40] B. Varghese, N. C. Wang, S. Barbhuiya, P. Kilpatrick, and D. S. Nikolopoulos, “Challenges and opportunities in edge computing,“ in IEEE International Conference on Smart Cloud (SmartCloud), 2016, pp. 20–26.

[41] A. Sether, Cloud Computing Benefits, 2016. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2781593

[42] T. R. Bastürk, Entwicklung und Implementierung einer IIoT-Plattform zur Anomalieerkennung in Datenströmen auf Basis des maschinellen Lernens. Master Thesis, TH Köln, 2021, unpublished.

[43] K. Sarda, A. Acernese, V. Nolè, L. Manfredi, L. Greco, L. Glielmo, and C. Del Vecchio, “A multi-step anomaly detection strategy based on robust distances for the steel industry,” IEEE Access, vol. 9, pp. 53827–53837, 2021.

[44] M. Jelali, L. Al-Shrouf, and D. Zander, “Neue Radarmesssysteme zur Material-detektion und Breitenmessung in Warmwalzwerken,” Stahl + Technik, no. 1–2 (February), 2022.

[45] T. Lin, Y. Wang, X. Liu, and X. Qiu, „A survey of transformers,“ arXiv preprint arXiv:2106.04554, 2021.

[46] J. Xu, H. Wu, J. Wang, and M. Long, „Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy,“ ArXiv, abs/2110.02642, 2021.

[47] T. Chen and C. Guestrin, „XGBoost: A Scalable Tree Boosting System,“ In Proceedings of the 22nd ACM SIGKDD International SConference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 785–794, 2016.

[48] R. Shwartz-Ziv, and A. Armon, „Tabular Data: Deep Learning is Not All You Need“, Inf. Fusion, vol. 81, pp. 84–90, 2022.

[49] M. U. Togbe, Y. Chabchoub, A. Boly, M. Barry, R. Chiky, and M. Bahri, “Anomalies detection using isolation in concept-drifting data streams,” Computers, vol. 10, no. 1, 13, 2021.

[50] M. Jain and G. Kaur, “Distributed anomaly detection using concept drift detection based hybrid ensemble techniques in streamed network data,” Cluster Computing, vol. 24, pp. 2099–2114, 2021.