Explainable Unsupervised Multi-Sensor Industrial Anomaly Detection and Categorization

1st Mina Ameli
Cognitive Assistants Systems Department
DFKI GmbH
Saarbrücken, Germany
Mina.Ameli@dfki.de

2nd Philipp Aaron Becker
Lab Department
deZem GmbH
Berlin, Germany
philipp.becker@dezem.de

3rd Katharina Lankers
Research & Development – Data Science Department
Schott AG
Mainz, Germany
katharina.lankers@schott.com

4th Markus van Ackeren
Research & Development – Data Science Department
Schott AG
Mainz, Germany
markus.vanackeren@schott.com

5nd Holger Bähring
Development Department
SEITEC GmbH
Erfurt, Germany
hb@seitec.info

6th Wolfgang Maaß
Smart Service Engineering Department
DFKI GmbH
Saarbrücken, Germany
Wolfgang.Maass@dfki.de

***Abstract*—Real-time Anomaly Detection is of great importance in industrial applications in order to have high-quality production and avoid downtime or failure of the system. In this paper, we study the application of anomaly detection over the multivariate data collected from Glass Production Industry. Our experiments utilize and compare different Unsupervised multivariate time series Anomaly Detection and Localization algorithms that have already demonstrated significant results on the state-of-the-art data sets. We propose a two-level multivariate anomaly detection approach that not only detects anomalous events in the production line but also categorize the different type of anomalies based on statistical pattern recognition. Furthermore, we localize the anomalous sensors by utilizing Explainable-AI approaches to help better decision-making in glass production monitoring. In this work, we propose an efficient pipeline for Anomaly Detection, Categorization and Localization which the experiments show promising results.**

***Index Terms*—Anomaly Detection, Anomaly Categorization, Statistical Pattern Recognition, Explainable AI, Glass Production, Multivariate Time series, Unsupervised Learning.**

I. Introduction

Anomaly Detection is a well-studied problem due to its wide range of applications. The task of anomaly detection refers to determining if a given data point fits the normal distribution; a non-fitting data point is called an anomaly. An anomaly may indicate various rare events such as production faults, systems defects, systems intrusions, service bottlenecks, or tissue degradation, which is therefore of primary focus in many applications. Therefore, anomaly detection for the recognition of rare sub-sequences in times series data is an important task with a considerable spectrum of applications ranging from industrial manufacturing processes to health care monitoring.

During any industrial process, a number of both wanted abnormal behaviors like process changes and unwanted events, including point anomalies, sensor defects, and corrosion-related drift, would occur, each of which requires a different measure on the part of the operator. However, a robust distinction between different types of changes or anomalies can often only be derived from the interaction of different sensor data.

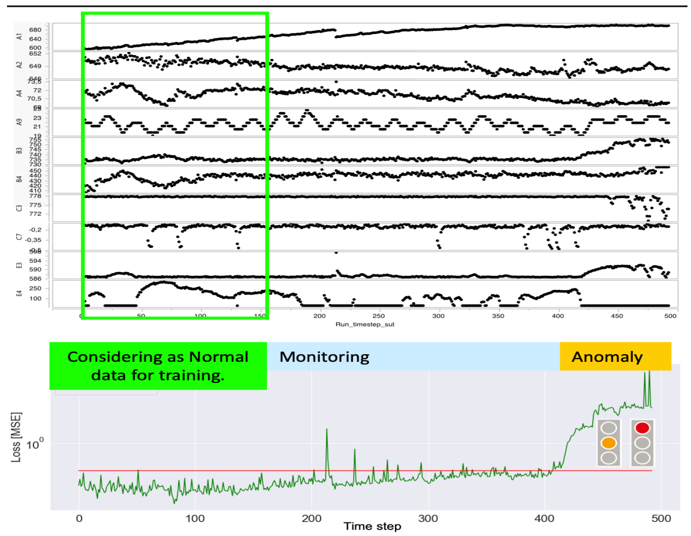

Fig. 1. General overview of Unsupervised Multivariate Time Series Anomaly Detection: a part of the data which is assumed as normal data is considered for training and anomaly score is utilized for Anomaly Detection and monitoring the system.

With classic methods from the field of statistical process control, however, only a subset of these phenomena can be detected, which often leads to downtime or failure of the system. Therefore, Early detection and differentiation of process drift caused by corrosion, point anomalies, and failure of individual sensors are required. Furthermore, these phenomena should be separated from the desired process setting changes.

Defect formation/anomaly should be detected as promptly as possible during the system run-time to intervene if necessary and avoid loss of quality of the product or the spread of the defect. It is also essential to know which area of the system the anomaly occurs; therefore, embedding multi-sensors in the whole system and localizing the anomalous sensors could be helpful for proper decision-making by domain experts. One of the challenges of anomaly detection in real-world scenarios is the lack of labeled data or anomalies, which necessitates the use of unsupervised or semi-supervised methods. For the purpose of multivariate time series anomaly detection, usually, the first part of the data would be considered normal data and would be utilized for training the model. Afterward, the test data would be fed to the model, and the reconstruction error is assumed as an anomaly score for monitoring the data and the system. The overview of training data splitting in Unsupervised Multivariate Time Series Anomaly Detection is depicted in 1. One of the drawbacks of such approaches is that they use a simple threshold on anomaly score [9] and this would lead to false positive alarms. There are some anomaly patterns that are not severe anomalies in the system, and they are representative of a special event in the system, like changes in the setting.

While studying the use case of anomaly detection on glass production, we learned level shift anomaly patterns in the anomaly score are representative of changes in the settings (non-critical anomaly) and should be distinguished among other kinds of anomalies (critical anomaly), and system recommendations and alerts should be of different kinds. This knowledge, besides the necessity of localization of anomalous sensors in a real-time scenario, lead our experiments, and we propose a pipeline to fulfill the challenges of this scenario. This paper proposes an efficient pipeline for solving anomaly detection, categorization, and localization in multivariate time-series real-world data from the glass production industry toward our domain expert's objective. Our main contribution can be summarized as follows:

- Comparison among different state-of-the-art unsupervised anomaly detection algorithms to find the proper candidate for the real-time scenario.
- A real-time anomaly indicator that aggregates the multivariate time series data to an anomaly score to provide early planning of repairs and measures.
- More stability by distinguishing between critical and non-critical anomalies by utilizing statistical pattern recognition on anomaly score.
- Rapid anomalous sensor localization and identification of causes through explainable AI.
- Distinguishing anomalies to provide recommendations based on the type of anomaly: critical and non-critical anomalies.

The remainder of the paper is organized as follows in Sect. II, we outline the related work. The Method and Background are explained in Sect. III. Experimental results are presented in Sect. IV. And Sect. V concludes the paper.

II. Related Work

As multivariate time series data is often very large, suffers from noise, and displays complex and different patterns upon each task, researchers have designed various specialized algorithms for anomaly detection to tackle such anomalous patterns. Over the years, the number and variety of these algorithms have grown considerably, each originating from various research areas such as deep learning, classical machine learning, data mining, signal analysis, stochastic learning, and statistics (regression and forecasting). Although deep learning methods have been favored recently for their high ability to generalize complex problem spaces, overall, they do not display stark and unchallenged performance in every task, as simple methods in cases nevertheless yield results almost as good as that of more sophisticated ones.

One of the stochastic models applied for capturing the relationship between multivariate time series for anomaly detection is a Vector Autoregression (VAR) model. Being a fast and easy-to-apply method, it's been scrutinized for various anomaly detection tasks. [15] constructed a multi-stage VAR model for the operation of the powerplant on time-series data from the combined cycle utility gas turbines, assuming sparsity in the association among variables. [16] applied VAR for dynamic fraud detection in graph spectral time series data. In another work, [17] utilized the same method for a continuous furnace activity in production to describe furnace measurements' dynamic behavior with the goal of identifying possible failures sufficiently far in advance.

Autoencoders are a type of neural network that encode the normal representation of the unlabeled data and reconstruct it from the learned latent space. To that extent, it has effectively been employed for various anomaly detection tasks to learn the normal pattern of data and assess the likelihood of whether the reconstruction of any given new data point from the learned encoding conforms to the given data point. There has been significant growth in the number of research works using the autoencoder for anomaly detection of multivariate time series data. Many different recurrent neural networks model has been adapted to autoencoder setting to learn the temporal dependencies in time series data. Among them, the Long Short-Term Memory (LSTM) autoencoders are one of the most applied methods for anomaly detection in multi-sensor time-series signals [5], [7], [9], [13], [25]. Furthermore, there has been a significant amount of work exploring the application of LSTM autoencoder further for various anomaly detection tasks, proposing more sophisticated structures such as LSTM-based variational autoencoder [26], [27] and LSTM models coupled with deep convolutional autoencoder to characterize spatial dependence of time series data [6], [28] to deal with industrial application, such as failure detection in water plants, production equipment, smart manufacturing [13], Spacecraft [25] and servers.

Some researchers have suggested unsupervised multivariate anomaly detection methods based on Generative Adversarial Networks (GANs) in the same autoencoder setting to deal with increasingly dynamic and complex systems such as cyber-physical systems [30], [31].

One of the drawbacks of conducting semi-supervised methods for unsupervised data is that they might train the model with anomalous samples which contributes to false alarms in some scenarios. To that extend, [14] proposed a self-supervised method to capture the relationship between univariate time series explicitly by addressing each as an individual feature and cooperating with a graph attention network to learn the complex interaction of time series in feature dimension as well as learning their overall temporal dependency. [29] used a similar method to learn graph relationships between features and detect deviation from these patterns, while incorporating sensor embedding.

However, one of the limitations of the aforementioned existing methods is that they use a simple threshold or rolling-threshold on anomaly score, and this would lead to false positive alarms. There are some anomaly patterns that are not critical anomalies in the system, and they are representative of a special event in the system, like changes in the setting. Our method differs from these, and we propose statistical pattern recognition on aggregated anomaly scores from multivariate data to distinguish critical and non-critical anomalies instead of using only threshold.

III. Method and Background

In this section, we describe the details of our proposed Multi-Sensor Anomaly Detection, Categorization and Localization pipeline in an unsupervised multivariate setup. The overview of the pipeline is shown in 2. In the first step, the model is trained with normal data. Afterwards, test data would be fed to the model and anomaly score is generated. Then the anomalies based on their pattern would be categorized. After finding the data points labeled faulty, we construct a wrapper to integrate an autoencoder with saliency XAI methods to calculate which sensors are related to the anomaly and how much they are contributing. Having metric (sensor) attribution values to anomalies, we provide meaningful interpretation to localize the anomalous sensors. In the following sections, we briefly discuss all the associated fields.

A. Anomaly Detection

For the purpose of anomaly detection in our use case, since we had available knowledge from domain experts that we could consider the first part of each run of the time series data, we did experiments over the widely used category of algorithms in application-based anomaly detection. Among different algorithms, we did experiments over Vector Autoregression, LSTM Autoencoder, and OmniAnomaly. In the following, we will introduce each of these algorithms, and we will discuss the results in the experiments section.

1) Vector Autoregression: The *VAR* is a stochastic model applied for capturing the relationship between multivariate time series as they evolve over time. Being fast, flexible, and easy to apply, it is one of the most applied methods for describing the dynamic behavior of time series for forecasting. VAR is an extension of univariate autoregressive models to multivariate time series data. Similar to that of univariate autoregressive, the VAR is modeled as a linear function of previous values where each variable has an equation that models its and other variables' evolution over time (1)

$$Y_t = c + A_1 Y_{t-1} + A_2 Y_{t-2} + \cdots + A_p Y_{t-p} + \varepsilon_t \quad (1)$$

where $t = 1, \ldots, T$ denotes the length of time series, $Y_t \in \mathbb{R}^k$ is vector of k temporal variables, p is a lag order value, $A_t \in \mathbb{R}^{k \times k}$ is a coefficient matrix which represents the relationship between the time series and their lagged values, $\varepsilon_t \in \mathbb{R}^k$ is the vector of zero-mean white noise. For example, for a bivariate VAR model with p=2, the equation them has the form

$$\begin{aligned} \begin{pmatrix} y_1^t \\ y_2^t \end{pmatrix} = & \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \begin{pmatrix} a_{11}^1 & a_{12}^1 \\ a_{21}^1 & a_{22}^1 \end{pmatrix} \begin{pmatrix} y_1^{t-1} \\ y_2^{t-1} \end{pmatrix} \\ & + \begin{pmatrix} a_{11}^2 & a_{12}^2 \\ a_{21}^2 & a_{22}^2 \end{pmatrix} \begin{pmatrix} y_1^{t-1} \\ y_2^{t-1} \end{pmatrix} + \begin{pmatrix} \varepsilon_1^t \\ \varepsilon_2^t \end{pmatrix} \end{aligned} \quad (2)$$

Notice that each equation includes the lagged values of exogenous variables as well, making the VAR model a seemingly unrelated regression model.

For the given formula (1), the input data is used to estimate the coefficients $\hat{A}$ using multivariate least-squares estimation. However, one of the main challenges is to determine the lag order for the estimation. The choice of the lag order p is determined by information criteria-based order selection. To that extend, we first fit the VAR models with orders $p = 0, \ldots, p_{max}$ to find $\hat{p}$ that minimizes the information criteria. The overall model selection criteria have the form

$$IC\left(p\right) = \ln \left|\tilde{\Sigma}\left(p\right)\right| + c_T \cdot \varphi(k, p) \quad (3)$$

where $\tilde{\Sigma}\left(p\right) = T^{-}1 \sum_{t=1}^{T} \hat{\varepsilon}_t \hat{\varepsilon}_t^{'}$ is the residual covariance matrix, c_T denotes the sequence indexed by T, $\varphi(k, p)$ is a penalty term that penalises the large VAR models. In our experiments, we employ Akaike (AIC) information criteria as the main selection criteria (4).

$$AIC\left(p\right) = \ln \left|\tilde{\Sigma}\left(p\right)\right| + \frac{2}{T} p k^2 \quad (4)$$

For foresting unseen multivariate time series, we use the parameters of the VAR estimated by the multivariate least squares. Then the h-step forecasts are obtained by the chain rule of forecasts as

$$Y^{'}{}_{T+h|T} = c + \hat{A}_1 Y^{'}{}_{T+h-1|T} + \cdots + \hat{A}_{\hat{p}} Y^{'}{}_{T+h-\hat{p}|T} \quad (5)$$

For aggregation, we first calculate the T-2 error between the forecasted time series and the real-time series based on the residual mean and variance of the initial estimation. Finally, the forecast score is aggregated by

$$S(h) = \Psi\left(h\right) \Sigma \Psi\left(h\right)^{-1} \quad (6)$$

where $\Psi\left(h\right)$ is the T-2 error score for the forecast series, Σ denotes the residual covariance matrix of initial estimation. Ultimately aggregated score of the multivariate time series is used by the anomaly detection method for detecting the anomalous time series.

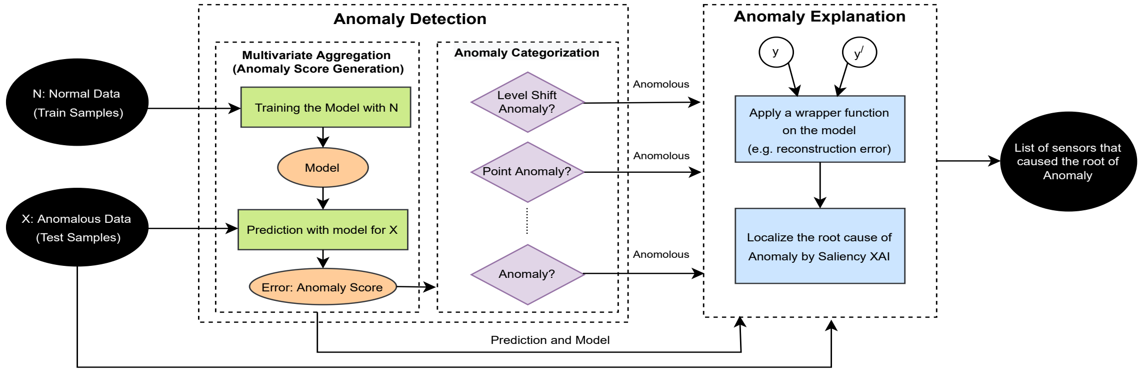

Fig. 2. Overview of the proposed pipeline for Unsupervised Multi-Sensor Anomaly Detection, Categorization and Localization in Glass Production.

2) LSTM Autoencoder: LSTM is an extension of recurrent neural networks which enables information flow from previous time steps to be used for the current step, allowing the model to learn from previous information. An LSTM unit is composed of a cell, input, output, and forget gates. The cell stores the list of previous information and recognizes the values over arbitrary time intervals, and gates control the propagation of information in and out of the cell. *LSTM Autoencoder* (LSTM-AE) is type of unsupervised neural network that learns how to effectively reconstruct the given input data using LSTM layers, thus capturing the possible dependence in time series variables. LSTM-AE has a bottle-neck structure where the input x_i is first mapped to small latent space space z_i through the series of hidden layer with decreasing number of nodes in encoder network, as shown in (7)

$$z_i = f_e \left(w_e x_i + b_e \right) \tag{7}$$

where f_e is an activation function for the encoder, w_e is the weight matrix, b_e is bias.

Then the latent variable z_i is transformed back to the original input space x_i in the decoder network through the series of the hidden layer with the opposite behavior, as shown in (8)

$$x_i = f_d \left(w_d z_i + b_d \right) \tag{8}$$

where f_d is an activation function for the decoder, w_d is the weight matrix, b_d is bias.

In our experiments, we train LSTM-AE for multivariate time series using four hidden layers, where the number of nodes in each layer is halved in the encoder network and doubled in the decoder network. Moreover, mean squared error is utilized as the main loss function to minimize the overall difference between the real and reconstructed samples along time series sequences. Later for the unseen data, reconstructed values are aggregated into a reconstruction score by performing mean absolute error.

A small overall reconstruction score would mean that the input x_t is constructed with firm certainty, and it can be expected to follow the normal behavior of time series data. On the other hand, a high reconstruction score would induce that the observation might not follow the normal pattern of time series, hence most likely being an anomaly. As in the previous example, we use the same anomaly detection method on the reconstruction score to make any prediction about the anomalous state.

3) OmniAnomaly: OmniAnomaly [23] is a stochastic recurrent neural network designed to model temporal dependencies in multivariate time series data. It incorporates Gated Recurrent Unit (GRU) [18], Variational Autoencoder (VAE) [19], and Normalizing Flows (NF) [20] to achieve robust representation of time series in terms of both stochasticity and temporal dependence of multivariate time series.

GRU is a variant of the gated recursive neural networks to capture temporal dependence between multivariate observations. In addition to having a forget gate such as LSTM cells, GRU has fewer parameters as it does not utilize an output gate. Hence GRU is computationally less expensive compared to LSTM. However, the overall structure of *OmniAnomaly* has more parameters and requires more computational power than a simple LSTM-AE.

VAE is a deep Variational Bayesian method to transfer the high-dimensional input x_t to a smaller latent space z_i, then the reconstruct x_t from the latent space z_i. Input x_t at time t is reconstructed by being sampled from posterior distribution $p_\theta \left(x_t \mid z_t \right)$ and a prior $p_\theta \left(z_t \right)$ by maximizing the likelihood of reconstruction data.

$$p_\theta \left(x_t \right) = \int p_\theta \left(z_t \right) p_\theta \left(x_t \mid z_t \right) \tag{9}$$

However, data likelihood $p_\theta \left(x_t \right)$ can not be directly computed as it is intractable to compute $p_\theta \left(x_t \mid z_t \right)$ for every z. Intractable data likelihood also renders posterior density distribution $p_\theta \left(z_t \mid x_t \right)$ intractable in its turn, where $p_\theta \left(z_t \mid x_t \right) = \int p_\theta \left(x_t \mid z_t \right) p_\theta \left(z_t \right) / p_\theta \left(x_t \right)$. Therefore, in addition to decoder network modelling $p_\theta \left(x_t \mid z_t \right)$, VAE uses an encoder network $q_\phi \left(z \mid x_t \right)$ to approximate the $p_\theta \left(z_t \mid x_t \right)$, where ϕ and θ represent parameters of encoder and decoder networks, respectively. With the aforementioned encoder and decoder setting, data likelihood is formulated as

$$\begin{aligned} \log p_\theta \left(x_t \right) = & \ \mathbb{E}_z \left[\log p_\theta \left(x_t \mid z_t \right) \right] - D_{KL} \left(q_\phi \left(z \mid x_t \right) || p_\theta \left(z_t \right) \right) \\ & + D_{KL} \left(q_\phi \left(z \mid x_t \right) || p_\theta \left(z_t \mid x_t \right) \right) \end{aligned} \tag{10}$$

As Kullback-Leibler term $D_{KL}\left(q_\phi\left(z_t \mid x_t\right) || \, p_\theta\left(z_t \mid x_t\right)\right)$ is intractable and always greater than zero, $\log p_\theta\left(x_t\right)$ is thus represented only by variational lower bound (ELBO) $\mathcal{L}(x_t, \theta, \phi)$.

$$\mathcal{L}(x_t, \theta, \phi) = \mathbb{E}_z\left[\log p_\theta\left(x_t \mid z_t\right)\right] - D_{KL}\left(q_\phi\left(z \mid x_t\right) || \, p_\theta\left(z_t\right)\right) \tag{11}$$

To that extend, VAE is trained with the objective of maximising the likelihood of lower bound

$$\theta^*, \phi^* = arg \max_{\theta,\phi} \sum_{i=1}^{N} \mathcal{L}(x_t, \theta, \phi) \tag{12}$$

Ultimately, the final equation (12) can be computed using the Monte Carlo assumption [22].

The output of the encoder network is usually regarded as diagonal Gaussian distribution. Therefore, usually, simple approximations techniques are employed to estimate the posterior $q_\phi\left(z \mid x_t\right)$ to allow for efficient reconstruction. However, using simple approximation may have a significant impact on the quality of the model because $q_\phi\left(z \mid x_t\right)$ would not always necessarily follow Gaussian distribution, thus suffering from underfitting. For that, the planar NF method is applied to transform $q_\phi\left(z \mid x_t\right)$ using invertible mapping function to enable learning non-Gaussian posterior density $q_\phi\left(z \mid x_t\right)$.

The overall architecture of *OmniAnomaly* is composed of decoder and encoder networks. Encoder network uses the latent representation $z_{t-T:t}$ to reconstruct the input $x_{t-T:t}$ at time t, where $t-T:t$ denotes the sequence of observation from time $t-T$ to T. The encoder network is optimized by the second part of the variational lower bound to approximate the ϕ that is close to prior. Input observation x_t and the hidden variable e_{t-1} in GRU from the previous time step are passed to GRU unit to generate e_t. The hidden variable e_t is crucial for capturing temporal dependence in multivariate time series data over time. Hence authors concatenate e_t with z_{t-1} and fed into dense layer to generate ϕ_z parameters (mean μ_{z_t} and standard deviation σ_{z_t}). Once ϕ_z parameters are created, the output of encoder network z_t^0 is sampled from $\mathcal{N}_\phi\left(\mu_{z_t}, \sigma_{z_t}^2 I\right)$. The output z_t^0 has diagonal Gaussian distribution. Hence, in the last step, to learn non-Gaussian posterior distribution of $q_\phi\left(z \mid x_t\right)$, z_t is approximated by passing z_t^0 through a chain of k planar mapping transformations f^k. The final output of the encoder is the result of the planar mapping, $z_t = z_t^k$.

The structure of the decoder network, which reconstruct x_t from z_t, is similar to that of the encoder. The main difference is that it uses Linear Gaussian State Space Model [21] to capture dependence among the z-space variables (i.e., θ_z). The dependence is described as

$$z_t = O_\theta\left(T_\theta z_{t-1} + v_t\right) + \varepsilon_t \tag{13}$$

where O_θ and T_θ are transition and observation matrices, v_t and ε_t are transition and observation noises. At the time t, temporary dependent z_t approximated earlier is later sent to the decoder GRU cell along with the hidden variable d_{t-1} of the same cell to produce d_t. Next, d_t is directly passed to dense layer to generate θ_z parameters for the reconstruction. The output of decoder network x_t is then sampled from $\mathcal{N}_\theta\left(\mu_{z_t}, \sigma_{z_t}^2 I\right)$.

The reconstruction score or probability is denoted by conditional log probability as

$$S(t) = \log\left(p_\theta\left(x_t \mid z_{t-T:t}\right)\right) \tag{14}$$

The high $S(t)$ score indicates that the input x_t is constructed with high confidence, and it follows the normal behavior of time series data. On the other hand, a small value of the reconstruction score would mean observation can not be created with high certainty; it might not follow the normal pattern of time series, hence being an anomaly. OmniAnomaly uses its automatic threshold selection method to detect anomalies from reconstruction scores. However, in our experiments, we only use the aggregated reconstruction score. Instead of using the same anomaly detection method proposed by the authors, we use a different anomaly detection method on $S(t)$ to detect anomalies from the reconstruction score.

B. Anomaly Categorization

In reality, the nature of anomaly varies over different cases. We can see different patterns of anomaly, and a single anomaly detection method may not work universally for all anomaly detection problems simply because they would differ in behavior (pattern). For example, in some cases, there are certain spikes or a level shift in the data caused by parameter changes in the sensory input. Although these shifts or spikes are not due to any fault (rather due to changes in purposes), they might be labeled as enormous by standard anomaly detection algorithms. For example, in some cases, the normal behavior of the target sensor would be constant upward increase, where certain detectors would be more flexible to that, threshold anomaly detection algorithms would suffer starting from the point where increase exceeds the upper bound. Therefore, choosing and combining detection algorithms are key to building an effective detection module that can robustly detect different types of anomalous occurrences. To this end, we used Anomaly Detection Toolkit (ADTK) [24]. Anomaly Detection Toolkit (ADTK) is a Python package for unsupervised/rule-based time series anomaly detection.

1) Point Anomalies: Compares each time series value with its previous values. It is used to detect positive changes in the value. In [24], it is implemented as a transformer Double Rolling Aggregate. Double Rolling Aggregate rolls two sliding windows side-by-side along a time series, aggregates using a selected operation, and tracks the difference of the aggregated metrics between the two windows. This may help track changes in statistical behavior in a time series [24].

2) Threshold Anomalies: Compares each time series value with given thresholds. It always looks for values that lie below of upper threshold point and above the lower threshold point.

3) Level-Shift Anomalies: By monitoring the difference between the median values in two adjacent sliding time periods, Level-Shift Anomalies detect shifts in value level. It can be a useful option if noisy outliers occur regularly because it is not sensitive to sudden surges [24].

4) Volatility-Shift Anomalies: Tracks the difference between the standard deviations at two adjacent sliding time periods to identify changes in the level of volatility. It's typically employed to find the rise in fluctuation amplitude's volatility [24].

C. Anomaly Explanation

Most state-of-the-art anomaly localization approaches either depend on the anomaly detection pipeline (i.e., reconstruction score) or describe the likelihood of feature contributions through statistical tests. For example, one of the simple ways to localize the anomalies would be to construct the localization based on the individual reconstruction score of each metric by declaring the metrics with the highest anomaly score as the most anomalous metrics and so on. However, this setting does not provide extensive insight as it suffers from the same setbacks of not being able to benefit from the information available on the remaining metrics. In the anomaly explanation part, we use an saliency XAI method in multivariate time series for anomaly localization. Thus here, we focus on a method that can benefit from the information available at other metrics and their integration.

Regarding anomaly Localization, there are some studies related to explainable anomaly detection by using Explainable-AI approaches in the literature [1], [3]–[5] which focus on additive feature-based Explainable AI that as the main approach they experimented with Shapely values to explain the feature importance. However, in [2], we did experiments over different families of Saliency Explainable AI approaches and utilized explainable AI for the application of multi-sensor anomaly localization in time series data for the first time. We show Saliceny Explainable AI anomaly localization method outperforms the OmniAnomaly Localization [23] and also the reconstruction error baseline. Accordingly, we used this idea in our pipeline.

IV. Experiments

We trained three different algorithms and compared the running time to choose the best one for the real-time training and detection in our pipeline. In table I, the running time for each run of the data is presented. VAR and LSTM-AE have the minimum training and testing time respectively. Since real time anomaly detection while testing data is of great importance, we choose LSTM-AE. In the categorization phase, we differentiated among critical(point) and non-critical (level-shift) anomalies. In the explanation phase, the anomalous sensors were localized as a root cause of anomalies.

A. Data

The data set is collected from the sensory input of glass production line. It contains process data from 8 production runs at one plant. For each run, 31 measuring points are recorded; the designations of the measuring points reflect the assignment to individual areas that follow one another in the production process. Part of embedded sensors could be observed in 3. Each of the eight production runs contains 10,000 - 30,000 chronologically ordered data records, depending on how long the run lasted. The exact size of each run could be found in I. At the beginning of each run, it is to be assumed that the plant is in the "Normal state" and has no defect. Therefore, the first part of data sets is ideally as few as possible, a maximum of 3-4000 can be used for training. Subsequently, we use the set containing the first 3000 timesteps of the data as a training set, timesteps between the first 3000th and 4000th samples are used for the validation set, and the remaining part of the data sets starting from the 4000th samples are used for the experimental testing purposes.

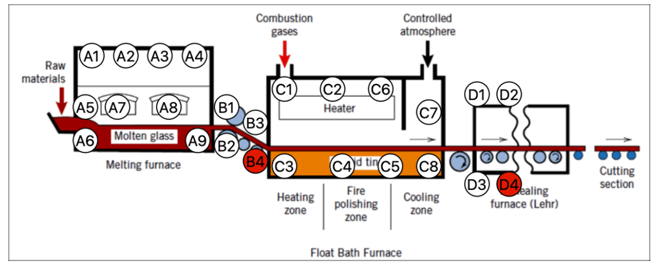

Fig. 3. Schematic diagram showing the float process for making sheet glass from [32] and overview of partial embedded sensors naming (A1-D4). The red sensors are anomalous sensors that localized by Explainable AI.

B. Experiment Setting

In our experiments, based on running time in I, we utilize LSTM autoencoder consisting of a two-layer encoder and decoder as a baseline model. Training the model is unsupervised and we considered the first part of the data as normal data. Firstly, features are encoded and then the latent feature representation is reversed in the same order in the decoder part to reconstruct the initial data. The increase in reconstruction error for anomalous data will lead us to detect anomalies. During the preprocessing, the data is normalized by min-max scaling, and then it is segmented into sequences through a sliding window of length 50. Moreover, if any data is missing, then the corresponding row is removed from the dataset. We use the ADAM optimizer and stochastic gradient descent with a learning rate of 10^{-3} with a mini-batch size of 64 to train the model. We train the model with an early stopping technique on the validation data back from 20% of the training. Furthermore, to deal with the gradient overflow and prevent the exploding gradients, we incorporate a gradient clipping of norm with 5.0 as a limit. We compute feature (metric) importance utilizing a saliency XAI method. Moreover, we use the Mean Absolute Error Equation (MAE) as the wrapper function rather than Mean Squared Error (MSE) to reduce the effect of sudden sharp increases in the reconstruction error, which might result from the noise in the data.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{15}$$

where y_i is prediction, x_i is true value, and n is the total number of data points.

C. Results

The overall goal of this work is designing a pipeline to detect anomalies in glass production line, localize the anomaly root sensors and provide recommendations for cost-effective maintenance. For doing so, we first scrutinized what anomaly detection architecture is better suited both for quality anomaly detection and explanation generation pipeline. We did that considering the accuracy of anomaly detection, time cost required to train the model, and the compatibility of the model with a proper XAI method. Having analyzed these factors, we decided to proceed with LASTM-Autoencoder which is fairly accurate, considerably fast, and easily adaptable to the XAI methods.

Later we aggregated the generated multivariate samples from the LASTM-Autoencoder into univariate time series to capture the interaction of the features in a single variable and use it with a robust and adaptive anomaly detection score. To distinguish different kinds of anomalies, in the next step, we employed four different types of detector algorithms with unique abilities. Furthermore, we demonstrated what kinds of different anomaly patterns we have observed in the data and how the suggested detectors can perform on them. The final stage is Explanation Generation which creates the meaningful localization for the found anomalous regions. In figure 4, the result of anomaly detection over eight production runs is demonstrated. The blue signal in plot is the anomaly score (MAE); the level shift patterns as non-critical anomalies are shown in yellow and point anomalies as critical anomalies in red region. As the last step, anomalous sensors are localized with saliency XAI approaches; an example could be found in figure 3. The red sensors are the root cause of the anomaly. The algorithms that we utilized have already shown good results over state-of-the-art datasets and the results over real world data are promising based on domain experts' analysis.

V. Conclusion

We presented a pipeline for anomaly detection, categorization, and localization over unsupervised real-world multivariate time series data from glass production. We made a Comparison among different state-of-the-art unsupervised anomaly detection algorithms to find the proper candidate for the real-time scenario. A real-time anomaly indicator that aggregates the multivariate time series data to an anomaly score to provide early planning of repairs and measures is conducted. Moreover, we presented a more stable approach due to distinguishing between critical and non-critical anomalies by utilizing statistical pattern recognition on anomaly score. Furthermore, we have demonstrated anomalous sensor localization through explainable AI.

Acknowledgment

This work was conducted as part of the SPAICER project. SPAICER is funded by the Federal Ministry for Economic Affairs and Climate Action (BMWK). The proposed approach is part of the Smart Resilience Services. We thank all SPAICER members and partners for their support.

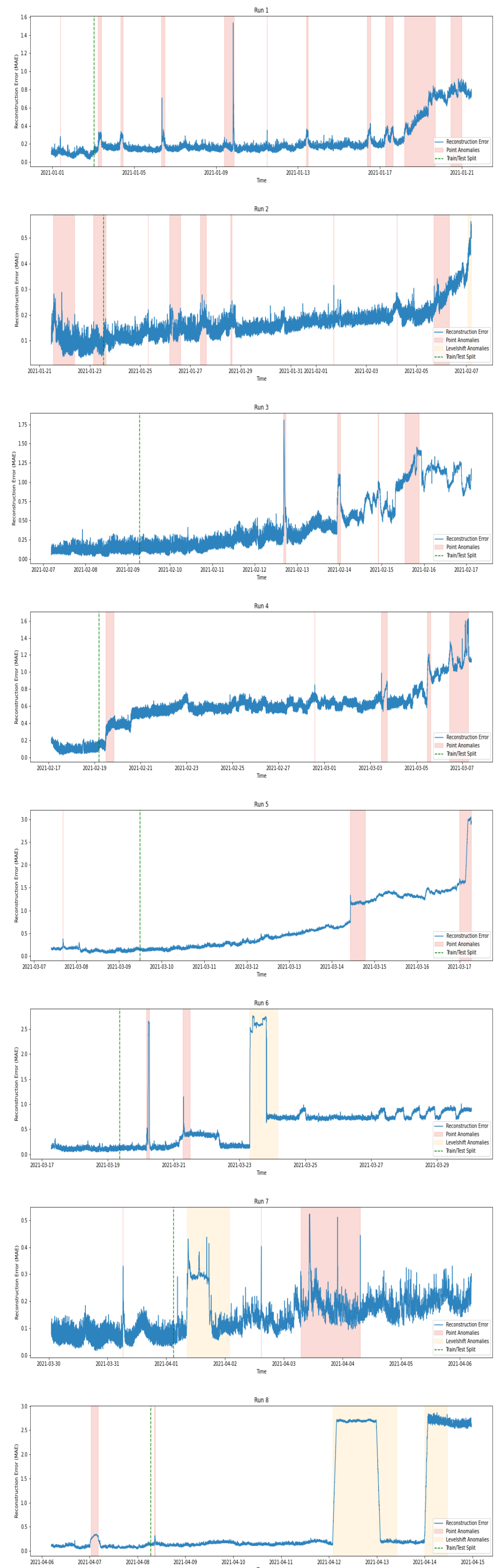

Fig. 4. The result of Anomaly Detection over the eight runs of glass production data: The blue plot is aggregated anomaly score as a result of the anomaly detection algorithm. Red areas are point anomalies, and yellow areas are level-shift anomalies.

TABLE I
DATA SIZE RELATED TO EACH RUN AND COMPARISON OF MODELS TRAIN AND TEST RUNNING TIME ON DIFFERENT ALGORITHMS IN SECONDS.

Run ID	**Data Size**			**Model running times (in seconds)**					
				LSTM Autoencoder		***Vector Autoregressor***		***OMNI***	
	Whole	***Train Samples***	***Test Samples***	***Train Time***	***Test Time***	***Train Time***	***Test Time***	***Train Time***	***Test Time***
1	32475	2940	29535	148.06	1.03	0.42	7.87	2026.34	31.99
2	27086	3000	24086	150.87	0.85	0.48	5.97	2051.75	26.23
3	17297	3000	14297	152.43	0.56	0.51	3.12	2063.85	15.52
4	29331	3000	26331	153.17	0.94	0.66	7.78	2048.91	28.6
5	17214	3000	14214	152.62	0.5	0.57	3.27	2053.05	15.41
6	21380	3000	18380	152.69	0.66	0.48	4.07	2112.75	19.95
7	13303	3000	10303	152.92	0.37	0.55	2.2	2138.54	11.19
8	15643	3000	12643	152.83	0.44	0.53	2.88	2118.54	13.93
Average	*21716*	*2992*	*18723*	*151.94*	***0.66***	***0.52***	*4.64*	*2076.71*	*20.35*

REFERENCES

[1] M. Carletti, C. Masiero, A. Beghi, GA. Susto, "Explainable machine learning in industry 4.0: Evaluating feature importance in anomaly detection to enable root cause analysis," In 2019 IEEE international conference on systems, man and cybernetics (SMC), pp. 21–26, October 2019.

[2] M. Ameli, V. Pfanschilling, A. Amirli, W. Maaß, K. Kersting. "Unsupervised Multi-sensor Anomaly Localization with Explainable AI," In IFIP International Conference on Artificial Intelligence Applications and Innovations, pp. 507–519, Springer, Cham, 2022.

[3] I. Giurgiu, A. Schumann, "Additive explanations for anomalies detected from multivariate temporal data," In Proceedings of the 28th acm international conference on information and knowledge management, pp. 2245–2248, November 2019.

[4] N. Takeishi, "Shapley values of reconstruction errors of pca for explaining anomaly detection," In international conference on data mining workshops (icdmw), pp. 793–798, IEEE, November 2019.

[5] L. Antwarg, RM. Miller, B. Shapira, L. Rokach, "Explaining anomalies detected by autoencoders using SHAP," arXiv preprint arXiv:1903.02407, March 2019.

[6] Y. Zhang, Y. Chen, J. Wang, and Z. Pan, "Unsupervised deep anomaly detection for multi-sensor time-series signals," IEEE Transactions on Knowledge and Data Engineering, 2021.

[7] S. Maleki, S. Maleki, and N.R. Jennings, "Unsupervised anomaly detection with LSTM autoencoders using statistical data-filtering," Applied Soft Computing, 108, p.107443, 2021.

[8] L. Jiang, H. Xu, J. Liu, X. Shen, S. Lu and Z. Shi, "Anomaly detection of industrial multi-sensor signals based on enhanced spatiotemporal features," Neural Computing and Applications, pp. 1–13, 2022.

[9] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, G. Shroff, "LSTM-based encoder-decoder for multi-sensor anomaly detection," arXiv preprint arXiv:1607.00148, July 2016.

[10] B. Siegel, "Industrial anomaly detection: A comparison of unsupervised neural network architectures," IEEE Sensors Letters, pp. 1–4, July 2020.

[11] J. Suschnigg, B. Mutlu, A.k. Fuchs, V. Sabol, S. Thalmann and T. Schreck, "Exploration of Anomalies in Cyclic Multivariate Industrial Time Series Data for Condition Monitoring," In EDBT/ICDT Workshops, pp. 1–8, 2020.

[12] D. Li, D. Chen, J. Goh, SK. Ng, "Anomaly detection with generative adversarial networks for multivariate time series," arXiv preprint arXiv:1809.04758, September 2018.

[13] RJ. Hsieh, J. Chou, CH. Ho, "Unsupervised online anomaly detection on multivariate sensing time series data for smart manufacturing," In 2019 IEEE 12th Conference on Service-Oriented Computing and Applications (SOCA), pp. 90–97, November 2019.

[14] H. Zhao, Y. Wang, J. Duan, C. Huang, D. Cao, Y. Tong, B. Xu, J. Bai, J. Tong, Q. Zhang, "Multivariate time-series anomaly detection via graph attention network," In 2020 IEEE International Conference on Data Mining (ICDM), pp. 841–850, November 2017.

[15] V. Goyal, M. Xu, J. Kapat, "Use of vector autoregressive model for anomaly detection in utility gas turbines," In Turbo Expo: Power for Land, Sea, and Air, American Society of Mechanical Engineers, Vol. 58608, p. V003T08A004, June 2019.

[16] Y. Li, A. Lu, X. Wu, S. Yuan, "Dynamic anomaly detection using vector autoregressive model," In Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 600–611, Springer, Cham, April 2019.

[17] F. Abbracciavento, S. Formentin, J. Balocco, A. Rota, V. Manzoni, and S. M. Savaresi, "Anomaly detection via distributed sensing: a VAR modeling approach," IFAC-PapersOnLine, pp. 85–90, 2021.

[18] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," In NIPS Workshop on Deep Learning, December 2014.

[19] D. P. Kingma, M. Welling, "Auto-Encoding Variational Bayes," arXiv preprint arXiv:1312.6114, 2013.

[20] D. J. Rezende, S. Mohamed, "Variational Inference with Normalizing Flows," Proceedings of the 32nd International Conference on International Conference on Machine Learning, - Volume 37, pp. 1530–1538, 2015.

[21] G. Kitagawa, W. Gersch, "Linear Gaussian State Space Modeling," Smoothness Priors Analysis of Time Series, New York, NY: Springer New York, pp. 55–65, 1996.

[22] C. Robert and G. Casella, "Monte Carlo Statistical Method," Technometrics, 2000.

[23] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network," Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining, Anchorage, AK, USA, pp. 2828–2837, 2019.

[24] https://arundo-adtk.readthedocs-hosted.com/en/stable/

[25] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding," In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining (KDD '18). ACM, New York, NY, USA, pp. 387–395, 2018.

[26] Daehyung Park, Yuuna Hoshi, and Charles C. Kemp, "A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-Based Variational Autoencoder," IEEE Robotics and Automation Letters 3, 1544–1551, 2018.

[27] X. Zhou, Y. Hu, W. Liang, J. Ma, and Q. Jin, "Variational LSTM Enhanced Anomaly Detection for Industrial Big Data," IEEE Transactions on Industrial Informatics, 2020.

[28] W. Luo, W. Liu, and S. Gao, "Remembering history with convolutional lstm for anomaly detection," in IEEE International Conference on Multimedia and Expo (ICME), pp. 439–444, IEEE, 2017.

[29] A. Deng, B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," In Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 4027–4035, 2021.

[30] A. Geiger, D. Liu, S. Alnegheimish, A. Cuesta-Infante and K. Veeramachaneni, "TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks," 2020 IEEE International Conference on Big Data (Big Data), 2020, pp. 33-43, doi: 10.1109/BigData50022.2020.9378139.

[31] D. Li, D. Chen, L. Shi, B. Jin, J. Goh, and S.-K. Ng, 'MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks," arXiv, 2019.

[32] WD. Callister, DG. Rethwisch, "Fundamentals of Materials Science and Engineering," Introduction to Materials and Chemistry, John wiley sons, 2009.