Anomaly Detection in a Production Line: Statistical Learning Approach and Industrial Application

Rida Kheirallah[1*][0009-0008-8862-2264], Anis Hoayek[1], Frederic Grimaud[1], Mireille Batton-Hubert[1] and Patrick Burlat[2]

[1]Mines Saint-Étienne, Univ Clermont Auvergne, INP Clermont Auvergne, UMR CNRS 6158 LIMOS, F – 4023 , Saint-Étienne, France
rida.kheirallah@emse.fr
anis.hoayek@emse.fr
frederic.grimaud@emse.fr
batton@emse.fr

[2] WipSim, 15 Rue de la Presse, 42000, Saint-Etienne, France
patrick.burlat@wipsim.fr

Abstract. This paper explores industrial engineering, particularly focusing on discrete processes and emphasizing real-time control of Production Lines within these processes. A critical component of this control involves the incorporation of a dashboard system, essential for providing workshop managers with valuable insights into the estimated time required for each Production Order (PO) to progress through the remaining stations in the production line.

The key contribution of this work is the conception and development of a novel mathematical model, applied to real-world industrial data, capable of detecting anomalies within the production line. These anomalies are defined as deviations from expected timeframes. Constructed using Statistical Learning techniques and Information Theory, the model can be integrated within the dashboard framework, offering prompt identification of anomalies and ensuring optimal performance and efficiency of the production process.

Keywords: Production Line, Anomaly Detection, Statistical Learning

1 Introduction

Efficient management of POs in discrete manufacturing is vital. POs progress from the initial raw materials storage area to the finished goods warehouse, with workflows regulated to maintain optimal work-in-progress levels. Dashboard systems provide real-time visibility and data, crucial for maintaining uninterrupted workflows and estimating workstation waiting times, which are essential for managing PO workflows. Despite management efforts, challenges in estimating waiting times persist, causing anomalies that affect efficiency and customer satisfaction.

Our research introduces a tailored anomaly detection method for production lines using real-time dashboard data. Aggregating advanced classification models like Isolation Forest (IF), AutoEncoder (AE), and Support Vector Machine (SVM) with Shannon Entropy, our model aims to improve detection and enhance productivity in manufacturing. In the following sections, our paper extensively examines anomaly detection in production lines.

Section 2 provides a thorough review of current research of anomaly detection in industrial contexts, especially within production line domains.

Section 3 starts by illustrating the database under analysis. It then defines anomalies mathematically within production lines, focusing on formalizing anomalies in PO flow.

Section 4 introduces the mathematical formalization and the construction steps of the aggregated model. This model will serve as a tool for anomaly detection outlined in the industrial application approach.

Section 5 showcases our anomaly detection model's practical application in industrial production lines. Using a dynamic, predictive approach, it identifies anomalies workstation by workstation emphasizing incremental detection over time.

Section 6 summarizes important findings, emphasizing insights from research and our combined model. It highlights the benefits our model offers to production line dashboards through proactive anomaly detection and suggests future research directions.

2 State of the Art

The success of businesses in the global economy depends on effective decision-making, especially in production lines where decision support is crucial. In the Industry 4.0 era, automated data analysis in industrial production lines is increasingly important. Intelligent methods for predicting and preventing equipment failures and optimizing maintenance operations enhance operational efficiency and contribute significantly to informed and strategic decision-making in these critical areas [1-3].

Detecting anomalies and analyzing root causes improve decision-making in production lines by identifying unexpected patterns. Efficient forecasting methods aid in balancing supply and demand, preventing understocking and enhancing production. Despite data overload, these techniques optimize inventory planning and decision-making. Many studies have concentrated on utilizing supervised and unsupervised classification models for anomaly detection. In [4], researchers explored unsupervised learning models like Skip-GANomaly, PaDiM, and PatchCore for industrial quality assessment, highlighting their competitive performance compared to supervised methods. They emphasized the effectiveness of fully unsupervised approaches in detecting anomalies in images. [5] provides a systematic mapping study on anomaly detection in industrial machinery using IoT devices and machine learning algorithms. It reviews 84 relevant studies from 2016 to 2023, identifying commonly used algorithms, preprocessing techniques, and sensor types. The study reveals that milling and cutting tools, hydraulic systems, and bearings are the most monitored machinery types, with vibration and temperature sensors being commonly employed. Various ML techniques, including supervised, unsupervised, and heuristic methods, are utilized, often combined with preprocessing techniques like Fast Fourier Transform and AutoEncoders to enhance accuracy.

However, challenges remain in detecting anomalies at the edge, integrating detecting anomalies into existing infrastructure, and adapting machine learning models to evolving industrial environments. [6] introduces GADPL (Generic Anomaly Detection for Production Lines) approach, for unsupervised anomaly detection in configuration-based production lines, focusing on predictive maintenance in industrial environments. It utilizes historical sensor data from a real reflow oven to anticipate failures, providing insights into heat and power consumption of individual fans. [7] investigates anomaly detection in manufacturing assembly data for two product series, utilizing various techniques such as HBOS (Histogram-Based Outlier Score), IF (Isolation Forest), KNN (K-Nearest Neighbors), CBLOF (Clustering-Based Local Outlier Factor), OCSVM (One-Class Support Vector Machine), LOF (Local Outlier Factor), and ABOD (Angle-Based Outlier Detcetion), with KNN and ABOD yielding the highest performance. Statistical root cause analysis revealed seven common rejection causes, with the top three accounting for 85% of rejection rates. Engineers reported reduced rejection rates after adjusting specifications based on the identified causes.

Beyond supply chains and production lines, anomaly detection and root cause analysis are vital in diverse fields such as intrusion and fraud detection [8-10].The literature is rich with methodologies utilizing neural networks, machine learning, and statistical approaches for novelty detection [11]. Detailed studies on techniques and applications in anomaly detection can be found in [12] and [13].

In our literature review, we found existing anomaly detection methods typically use single mathematical models, either supervised or unsupervised. Our research uniquely contributes by aggregating multiple models using Shannon Entropy to assess uncertainty and select the most effective approach for specific challenges like data type, scalability, interpretability, and robustness to noise. Thus, we can identify the most appropriate models for a given context and aggregate them to enhance detection performance. This adaptability to encountered problems is a very strong and essential result of the aggregation. In this article, the presented aggregated model is the result of combining the IF, SVM, and AE models, as these are extensively represented in the literature and are well-suited to our context, specifically the numerical type of data. However, the aggregated models may vary in a different context. In Table 1 below, we present, without loss of generality, a comparison among these individual models as well as between them and our aggregated model. This comparison assesses various criteria and performance indicators of statistical learning models encountered in the literature. In Table 1, a symbol of 1 indicates that the model meets the corresponding performance indicator, while 0 means it does not. Additionally, the symbol (1) denotes non-binary classification, indicating that the model's response is not conclusive and cannot be categorized as either a simple yes or no.

It is noteworthy that the primary strength of our model lies in its high capacity to differentiate between outliers and anomalies, attributed to Shannon Entropy. As depicted in this table, the aggregated model exhibits superior performance compared to each individual model. This superiority is attributed to its remarkable adaptability, allowing it to discern and select the most appropriate model tailored to the specific characteristics of the context at hand.

Our Python-based model is designed for easy integration into industrial dashboard systems. It offers seamless compatibility across diverse manufacturing environments, leveraging Python's versatility and built-in optimization for practical applicability.

Table 1. Performance comparison between individual models and the aggregated model

Criteria	IF	SVM	AE	Agg Model
High Capacity Distinguishing Anomalies/Outliers	0	0	0	1
Scalability	1	0	(1)	1
Interpretability	1	(1)	0	1
Handles Diverse Data Types	(1)	(1)	1	1
Robustness Noise	1	0	(1)	1

3 Anomaly Definition in Context of a Production Line

Before discussing and defining anomalies, Fig. 1 below illustrates the DB we are analyzing, provided by WipSim (a private software development company specializing in production line solutions), from a real production line. Thus, Fig. 1 presents an example of time archives for a PO with ID = 140 at a workstation named Conwip2_6_2. This workstation is situated in the initial fabrication stage of the fabrication process, where initial casts are made for components within an operational production line dedicated to manufacturing boring machines.

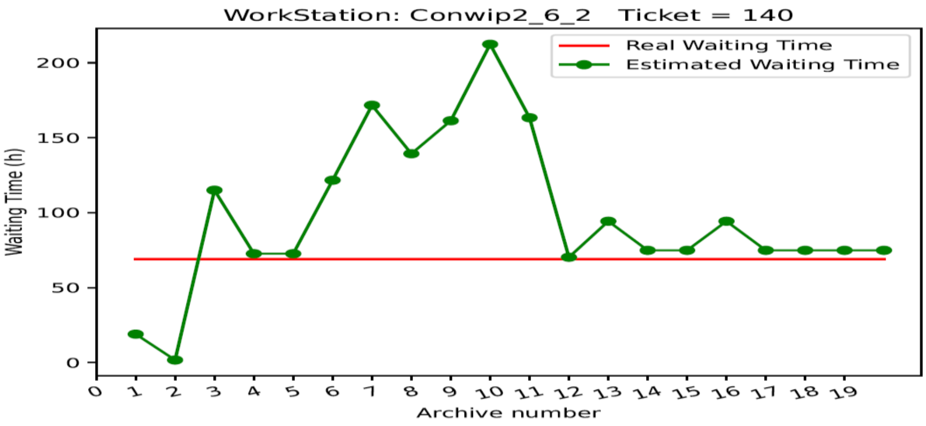


Fig. 1. Real Waiting Time VS Estimated Waiting Time of a PO on a WS_i

The X-axis of Fig. 1 represents all archives by their respective numbers, while the Y-axis represents the waiting time of the PO on the workstation Conwip2_6_2. Thus, this figure depicts the estimated waiting times of the PO on the workstation, associated with each archive from the first to the last archive (20 archives in total). These archives are captured before the PO traverses the workstation. At the moment of the first archive, the estimated waiting time for the PO to traverse the workstation is 19 hours, while it is 75 hours at the time of the last archive (20th archive). Fig. 1 also illustrates the convergence of estimated times toward the actual waiting time observed by the PO after passing through the workstation Conwip2_6_2, which amounts to 70 hours. This convergence phenomenon is indicative of a Markovian behavior, where the last estimation at any given instant takes into consideration all previous estimations. The Markovian nature of the chain allows for continuous refinement of estimations, leading to a more accurate representation of the actual waiting time on the workstation [14].

The estimated waiting times of a PO on a workstation WS_i (where $i \in \{1, \dots, \mathrm{L}\}$ and L is the total number of workstations) can be represented as a vector denoted $T^E_{(PO,WS_i)} \in \mathbb{R}^K$, such that,

$$T^E_{(PO,WS_i)} = (\, T^E_{(1)}{}_{(PO,WS_i)}, \dots ,\; T^E_{(j)}{}_{(PO,WS_i)}, \dots ,\; T^E_{(K)}{}_{(PO,WS_i)}\,), \tag{1}$$

where:

- K is the total number of archives,
- $T^E_{(1)}{}_{(PO,WS_i)}$ is the estimated waiting time of the PO at WS_i, recorded from the first archive at time $(t_1)_{(PO,WS_i)}$,
- $T^E_{(K)}{}_{(PO,WS_i)}$ is the estimated waiting time of the PO at WS_i, recorded from the last archive at time $(t_K)_{(PO,WS_i)}$,
- $T^E_{(j)}{}_{(PO,WS_i)}$is an intermediate estimated waiting time of the PO at WS_i, recorded from an intermediate archive at time $(t_j)_{(PO,WS_i)_i}$, where $(t_1)_{(PO,WS_i)} < (t_j)_{(PO,WS_i)} < (t_K)_{(PO,WS_i)}$.

Denoting $T^R_{(PO,WS_i)}$ as the actual waiting time of the PO on WS_i, we then define the error vector $E_{(PO,WS_i)} \in \mathbb{R}^K$, such that,

$$E_{(PO,WS_i)} \;=\; (\, E_{(1)(PO,WS_i)}, \dots,\; E_{(j)(PO,WS_i)}, \dots ,\, E_{(K)(PO,WS_i)}\,), \tag{2}$$

where $E_{(j)}{}_{(PO,WS_i)}$ is computed based on Equation 3:

$$E_{(j)}{}_{(PO,WS_i)} = T^R_{(PO,WS_i)} \;-\; T^E_{(j)}{}_{(PO,WS_i)}. \tag{3}$$

Thus, $E_{(j)}{}_{(PO,WS_i)}$ represents the error of the estimated waiting time corresponding to the archive recorded at time $(t_j)_{(PO,WS_i)_i}$ and $E_{(PO,WS_i)}$ represents the vector of all waiting time errors corresponding to each archive of the PO on WS_i. After defining

$E_{(PO,WS_i)}$, we consequently calculate from this vector its statistical median denoted as $Median\left(E_{(PO,WS_i)}\right)$ and given by:

$$Median\left(E_{(PO,WS_i)}\right) = T^{R}_{(PO,WS_i)} - Median\left(T^{E}_{(PO,WS_i)}\right), \tag{4}$$

where $Median\left(T^{E}_{(PO,WS_i)}\right)$ is the statistical median of $T^{E}_{(PO,WS_i)}$.

The three errors will allow us to define the Input Error Vector (IEV) for a PO at WS_i used by the Anomaly Detection Model and denoted as: $IEV_{(PO,WS_i)} \in \mathbb{R}^3$, such that,

$$IEV_{(PO,WS_i)} = \left(E_{(1)_{(PO,WS_i)}}, Median\left(E_{(PO,WS_i)}\right), \ E_{(K)_{(PO,WS_i)}} \right). \tag{5}$$

The concept of anomlay of a PO at a WS_i is based on the statistical analysis of the $IEV_{(PO,WS_i)}$. Then, we say that a PO has an abnormal behavior at WS_i if $f\left(IEV_{(PO,WS_i)}\right) = 1$, where,

$$\begin{gathered} f : \mathbb{R}^3 \to \{\,0, 1\,\}, \\ IEV_{(PO,WS_i)} \to f\left(IEV_{(PO,WS_i)} \right) = \begin{cases} 0 \Rightarrow PO \text{ is normal at } WS_i \\ 1 \Rightarrow PO \text{ is abnormal at } WS_i \end{cases}. \end{gathered} \tag{6}$$

The function f defined by the Equation 6 is the heart of our work, it represents the model that we constructed through the aggregation of multiple anomaly detection classification models, coupled with the theory of information through the Entropy of Shannon. The construction of the $IEV_{(PO,WS_i)}$ for our anomaly detection model is grounded in a thoughtful consideration of the diverse characteristics inherent in error data derived from different archives: $E_{(1)_{(PO,WS_i)}}$ provides valuable insights into the initial state of the simulation software behind the generation of the waiting time estimations. This inclusion is vital as anomalies in the early stages of data collection may signify underlying issues that could propagate over time. $Median\left(E_{(PO,WS_i)}\right)$ is incorporated to enhance the robustness of our model. The median serves as a robust measure of central tendency, offering resistance to the influence of outliers that may be present in the error data. Anomalies resulting from sporadic, extreme errors can distort the overall analysis, and by including the median, we achieve a more stable representation of the typical error magnitude. This, in turn, contributes to the model's ability to discern genuine anomalies from fluctuations. As for $E_{(K)_{(PO,WS_i)}}$, it represents the error corresponding to the last archive. This strategic choice is guided by the understanding that the last archive often captures a system or process where updates have been implemented to the simulation software, and these updates are permanent. By including $E_{(K)_{(PO,WS_i)}}$, our model accounts for potential shifts or improvements in the error patterns, ensuring a comprehensive analysis of the data's temporal evolution. This consideration becomes especially relevant when dealing with dynamic systems where the accuracy of error measurements may undergo significant changes over time. In summary, the integration of $E_{(1)_{(PO,WS_i)}}, Median\left(E_{(PO,WS_i)}\right)$ and $E_{(K)_{(PO,WS_i)}}$ in $IEV_{(PO,WS_i)}$ provides our anomaly detection model with a nuanced and holistic perspective on the temporal and dynamic aspects of error data, contributing to the model's effectiveness in identifying anomalies across the entire dataset.

Before presenting and detailing our model in Section 4, Appendix2 Table 2 (to access the Appendix2 repository on GitHub, please use the following link: https://github.com/rida87/APMS/blob/main/Appendix%202.pdf) exhibits the anomalies detected by our model for workstation Conwip2_6_2. Thus, this table illustrates the real DB of POs (identified by their IDs) that passed through the workstation Conwip2_6_2 from the first archive to the last. Column $T^{R}_{(PO,WS_i)}$ shows the actual values of the waiting time for the POs on Conwip2_6_2. Columns $T^{E}_{(1)_{(PO,WS_i)}}$, $Median\left(T^{E}_{(PO,WS_i)}\right)$ and $T^{E}_{(K)_{(PO,WS_i)}}$ indicate the estimated values of POs' waiting time. The columns $E_{(1)_{(PO,WS_i)}}$, $Median\left(E_{(PO,WS_i)}\right)$ and $E_{(K)_{(PO,WS_i)}}$ represent the components of $IEV_{(PO,WS_i)}$ corresponding to each PO. The last column $f\left(\ IEV_{(PO,WS_i)}\right)$ represents the output of our model. An output equal to one indicates an abnormal PO on Conwip2_6_2, while an output equal to zero signifies a normal PO on the workstation (Equation 6).

It is worth noting that in Appendix2 Table 2, each value v in the column $E_{(1)_{(PO,WS_i)}}$ is obtained from the Equation 7 below:

$$v = \ U(\,0\,,0.1\,)\ \mathbb{1}_{\{\,z\,\leq\ 0\,\}} + z\ \mathbb{1}_{\{\ z > 0\ \}}, \tag{7}$$

where $z = (\,T^{R}_{(PO,WS_i)}\ -\ T^{E}_{(1)_{(PO,WS_i)}}\,)$ and $U(\,0\,,0.1\,)$ represents a pseudo-random number drawn from a uniform distribution ranging between 0 and 0.1.
Analogously, each value of the column $Median\left(E_{(PO,WS_i)}\right)$ and the column $E_{(k)_{(PO,WS_i)}})$ is derived from Equation 7 using a similar approach.

The concept behind computing the values of $IEV_{(PO,WS_i)}$ from Equation 7 is to address situations where an error is negative. In such cases, (for example the PO with ID=885 in Appendix2 Table 2: first row of the table), it is systematically substituted with a random value generated from uniform distribution between 0 and 0.1 (Equation 7). This substitution serves a dual purpose: firstly, it aligns with the fact that a negative error signifies that the actual time is less than the estimated time. In the context of delivery-related processes, a negative error implies that the delivery is ahead of schedule, meeting or even exceeding client expectations. Secondly, by introducing this replacement mechanism, we strategically filter out positive errors, focusing the model's attention on potential anomalies associated with delayed deliveries. This nuanced approach enhances the sensitivity of our model to deviations from expected delivery timelines, contributing to a more refined anomaly detection capability.

It is crucial to highlight, as evident in Appendix2 Table2, that the errors associated with the first error are relatively higher compared to those corresponding to the median and the last errors. This trend is clearly reflected in the average, where $mean_{E_{(1)_{(PO,WS_i)}}} = 15.07$ hour while $mean_{Median\left(E_{(PO,WS_i)}\right)} =\ 7.28$ hour and $mean_{E_{(K)_{(PO,WS_i)}}} = 0.47$ hour. This is expected and not surprising, considering that during the initial estimations, the estimation parameters of the simulation software were not sufficiently up-to-date in contrast to the estimation parameters at the instant of the last estimation. This discrepancy accounts for the observed elevated error.

The anomaly detection by the model singling out the POs with IDs 504, 418, 371 and 772 (highlighted in red in Appendix2 Table 2), is substantiated by a meticulous analysis of the individual error components of $IEV_{(PO,WS_i)}$. Regarding the PO with ID=504, its first error (97.75 hour) and median error (81.33 hour) are extremely high compared to other observations. Concerning the second abnormal PO (ID=418), its first error (35.67 hour) and median error (59.4 hour) are relatively high compared to their averages ($mean_{E_{(1)(PO,WS_i)}}$ = 15.07 hour and $mean_{Median}\left(E_{(PO,WS_i)}\right)$ =7.28 hour). For the abnormal PO with ID=371, its first error (2693.33 hour) is the highest among all the values of the first error vector and similarly for its median error (194.4 hour). Lastly, for the last abnormal PO with ID=722, its last error (7.7 hour) is the highest among all values of the last error vector.

In this section, we have presented the DB sourced from a real production line. Moreover, we have explained what constitutes an anomaly and its significance within the production line. In the next section, we will explore the complexities of our aggregated model designed for anomaly detection within production lines.

4 Aggregated Model for Anomaly Detection

In this section, we outline the mathematical formalization of our aggregated model for anomaly detection in industrial applications discussed in Section 5. This model integrates three distinct anomaly detection models: IF [15], SVM [16] and AE [17]. Initially, each model is individually trained on a dataset (Train_Data). For each observation in Train_Data, scores are computed using each model and standardized for equitable comparison. Subsequently, Entropy derived from these scores quantifies data uncertainty and is also standardized. A pivotal aspect of our approach involves setting anomaly detection thresholds based on quantiles of standardized scores and Entropy. When applied to Test_Data, observations surpassing these thresholds in scores and Entropy are identified as anomalies.

The major advantages of this approach lie in its ability to aggregate multiple classification models for more robust detection, coupled with Entropy to quantify data uncertainty, allowing for adaptable anomaly detection. All this with adjustable hyperparameters and thresholds to optimize performance on various types of data. The reason for calculating Entropy is its use as a measure of diversity and uncertainty. When discriminating between abnormal and outlier observations, Entropy can be considered as an additional reason to make decisions.

In the following Subsection 4.1, we first present a mathematical formulation for each of the anomaly detection classification models. Next, in the Subsection 4.2, we outline the steps involved in constructing our aggregated model and demonstrate how the model detects anomalies.

4.1 Mathematical Formulation of Anomaly Detection Models

IF algorithm is a tree-based anomaly detection method that works by isolating instances in a dataset. Each instance is isolated through a random partitioning process in a binary tree. The algorithm exploits the fact that anomalies are often isolated instances and can be identified by shorter average path lengths in the trees. Mathematically, the isolation score $s(x,n)$ for an instance x in a tree of height n can be defined as the average path length in the tree, standardized by the expected average path length for random instances. The anomaly score is inversely proportional to $s(x,n)$, making instances with lower scores more likely to be anomalies. By applying the IF algorithm, these anomaly scores can be calculated, providing a quantitative measure of the anomaly likelihood for each instance.

SVMs are utilized in anomaly detection as one-class classifiers, aiming to find a hyperplane in a high-dimensional space that maximizes the margin between data points of one class and the rest. This hyperplane separates normal instances from potential anomalies, with the margin representing the distance to the nearest data points (support vectors). SVMs optimize a cost function to minimize misclassifications, distinguishing between normal and anomalous instances. Additionally, SVMs can calculate anomaly scores effectively, providing an extra method for anomaly identification within datasets.

AE is a type of neural network designed to learn efficient representations of input data by encoding it into a lower-dimensional latent space and then reconstructing the input from this representation. Mathematically, an AE consists of an encoder function $f : X \to Z$ and a decoder function $g : Z \to X$, where X is the input space and Z is the latent space. The goal is to minimize the reconstruction error, typically measured by a loss function like Mean Squared Error. The encoder and decoder functions are parameterized by neural network weights, and the training process involves adjusting these weights to find a compact representation of the input data. Importantly, it should be noted that anomaly scores derived from the Mean Squared Error can be effectively calculated using the AE algorithm, providing an additional means for discerning anomalies within the dataset.

4.2 Model Construction

Model Training and Anomaly Score Assignment. Each of the models of the adapted techniques is trained with a training dataset: Train_Data. Following this training, an anomaly score from each model is associated to each observation in Train_Data. This scores are then standardized, ensuring that each scores falls within the range of zero to one. When referring to standardized scores, it is important to note that each score corresponds specifically to the classification model employed. Consequently, three standardized scores, falling between zero and one, are associated with each observation in Train_Data. The standardization of the scores serves a crucial purpose in our analysis. By standardizing the scores obtained from the IF, SVM, and AE techniques, we ensure that the scores are on a consistent scale, making them directly comparable. This is essential for fair and meaningful comparisons between the different anomaly detection techniques, as it prevents any one technique from dominating the analysis due to score scale variations. Standardization also aids in the interpretation of the results and facilitates a more effective evaluation of the performance of each classification model, allowing us to draw robust conclusions from the comparisons.

Score into Probability Distribution and Entropy. For each observation in Train_Data, the three standardized scores calculated are transformed into a probability distribution. Thus, for a given observation of Train_Data, after calculating the following three standardized scores: $score_{SVM}, score_{AE}$ and $score_{IF}$, the probability corresponding to the SVM model (denoted as prb_{SVM}) is calculated using Equation 8:

$$\mathrm{prb}_{\mathrm{SVM}} = \frac{\mathrm{score}_{\mathrm{SVM}}}{\mathrm{score}_{\mathrm{SVM}} + \mathrm{score}_{\mathrm{AE}} + \mathrm{score}_{\mathrm{IF}}} \ . \tag{8}$$

Analogously, prb_{AE} and prb_{IF} are calculated in the same manner. Then after calculating prb_{SVM}, prb_{AE} and prb_{IF}, we calculate from these probabilities the Entropy of the observation using the following Equation:

$$\mathrm{Entropy} = -\sum_{\mathrm{j=1}}^{\mathrm{j=3}} \mathrm{prb_j} \times \log_2\left(\mathrm{prb_j}\right), \tag{9}$$

where $\mathrm{prb}_1 = \mathrm{prb}_{\mathrm{SVM}}$, $\mathrm{prb}_2 = \mathrm{prb}_{\mathrm{AE}}$ and $\mathrm{prb}_3 = \mathrm{prb}_{\mathrm{IF}}$.

Once calculated, Entropy is standardized and integrated into our decision criteria, serving as an additional factor when distinguishing abnormal observations. The combination of standardized Entropy with scores from individual anomaly classification models provides a comprehensive view. This nuanced approach enables the identification of anomalies, not solely based on extreme values but also on intricate and unpredictable patterns. The decision criteria, including predefined thresholds for both standardized scores and Entropy ensure accurate anomaly classification, reflecting our commitment to capturing the multifaceted nature of anomalies.

Identification of Quantiles. Following the calculation of standardized scores and the calculation of standardized Entropy, this step involves predefining the threshold percentages from which the quantiles of the standardized scores and standardized Entropy will be determined. Thus, in this step, we predefine the following four threshold percentages: IF Percentage threshold, SVM Percentage threshold, AE Percentage threshold, and Entropy percentage threshold. Once these percentages are defined, we identify from the standardized scores, for each of the three anomaly detection techniques, the quantile corresponding to the technique's threshold. We also identify the Entropy quantile corresponding to the Entropy percentage threshold from the standardized Entropy. After the identification of these quantiles, they are recorded. The three trained models are also saved to be later loaded for detecting anomalies in a test DB: Test_Data.

Anomaly Detection. Finally**,** for each observation in Test_Data, three standardized scores corresponding to each of the three classification models and the standardized Entropy are calculated. Then, for every observation, we considered it an anomaly if it satisfied the following two conditions:

1. The standardized scores are greater than predefined quantile values.

2. The standardized Entropy is greater than the predefined quantile value of the standardized Entropy.

Note the trained models are aggregated by combining their output scores assigned to Train_Data. This aggregation is achieved through a quantile-based approach and integrating Entropy calculations. Subsequently, the trained models are utilized to assign scores to the observations in Test_Data. These scores are then employed to detect anomalies within Test_Data. Through harnessing the diverse perspectives provided by multiple models, our aggregation method enhances the resilience and efficiency of anomaly detection in complex datasets.

5 Industrial Application

This section explores the industrial application of our anomaly detection model that operates workstation by workstation in real time. It dynamically and predictively trains on specific datasets for each workstation and detects anomalies using live data from recent activities. Workstations exceeding a set anomaly threshold are flagged as abnormal. This method continually identifies anomalies with each new data update, injecting flexibility into the process. Future sections will elaborate on this approach further.

5.1 Dynamic Segmentation Approach

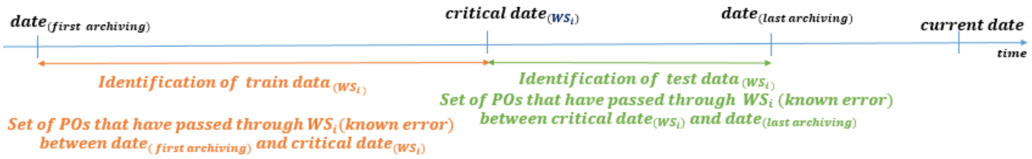
Diag. 1 below visually illustrates the dynamic segmentation approach described in this section. The initial step entails identifying the dataset All_Data for WS_i, which consists of errors from all POs traversing WS_i within the timeframe spanning from the first to the last archive date. Proceeding to the critical step of identifying Test_Data for WS_i. In this step, we first compute the Average Real Time, denoted as $Mean\ Real\ Time_{(WS_i)}$ required for a PO to navigate WS_i. Next, we calculate $Critical\ Date_{(WS_i)}$ using the following Equation:

$$\text{Critical Date}_{(\text{WS}_\text{i})} = \text{Date of the last archiving} - \left(\mu_{\text{Real Time}_{(\text{WS}_\text{i})}} + 3 \times \sigma_{\text{Real Time}_{(\text{WS}_\text{i})}}\right). \quad (10)$$

$\mu_{Real\ Time_{(WS_i)}}$ and $\sigma_{Real\ Time_{(WS_i)}}$ represent respectively, the mean and the standard deviation of the real waiting time of all POs on WS_i. The addition of the 3-sigma technique [18] is employed to include a sufficient range to cover approximately 99.7% of the possibilities in a normal distribution. It allows for a more comprehensive and exhaustive sampling technique in assessing the critical date.

Test_Data is in fact, the collection of $IEV_{(PO,WS_i)}$ from all the POs that passed through WS_i at a date later than $Critical\ Date_{(WS_i)}$. Conversely, Train_Data, extracted from All_Data by excluding Test_Data, serves as the training dataset, ensuring segregated evaluation methodology and preserving data integrity.

This segmentation approach (Diag. 1) captures dynamic data over time, providing an evolving representation rather than a static one. It identifies Test_Data to detect errors and timing inaccuracies synchronized with recent workstation activities. It also ensures integrity by separating training and test datasets for experiments.


$date_{(first\ archiving)}$
$critical\ date_{(WS_i)}$
$date_{(last\ archiving)}$
$current\ date$
$time$
$Identification\ of\ train\ data_{(WS_i)}$
$Set\ of\ POs\ that\ have\ passed\ through\ WS_i (known\ error)$
$between\ date_{(first\ archiving)}\ and\ critical\ date_{(WS_i)}$
$Identification\ of\ test\ data_{(WS_i)}$
$Set\ of\ POs\ that\ have\ passed\ through\ WS_i\ (known\ error)$
$between\ critical\ date_{(WS_i)}\ and\ date_{(last\ archiving)}$

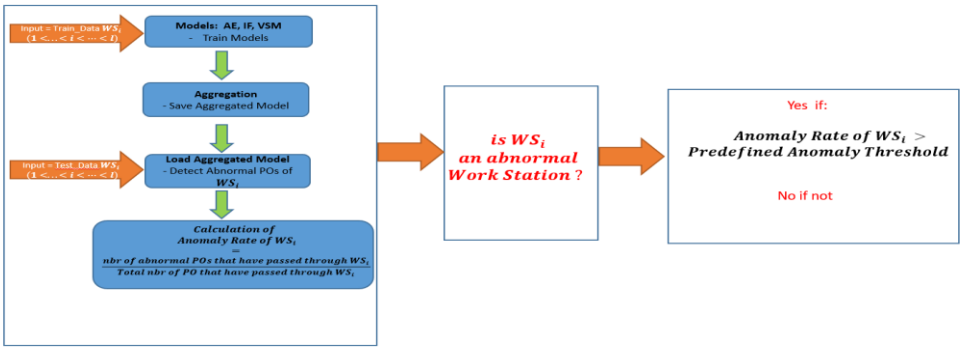
Diag. 1. Determination of Test_Data and Train_Data for a workstation WS_i

5.2 Anomaly Detection.

After identifying Train_Data and Test_Data for WS_i, each of the three techniques (IF, AE, and SVM) is trained using Train_Data. The trained models are subsequently aggregated. The resulting aggregated model is saved for later use (load Model) to detect abnormal errors (and thus abnormal POs) from Test_Data (Diag. 2). A PO is considered abnormal at workstation WS_i if the estimation error corresponding to the PO has been detected as an anomaly by the anomaly detection model. This allows us to calculate, for each workstation, an anomaly rate using the following Equation:

$$\text{Anomaly Rate of } \mathrm{WS_i} = \frac{\text{Nbr of Abnormal PO that have passed through } \mathrm{WS_i}}{\text{Total nbr of PO that have passed through } \mathrm{WS_i}} \; . \tag{11}$$

After associating an anomaly rate per workstation, workstations with anomaly rates higher than predefined threshold rates are identified and considered as abnormal workstations (Diag. 2).


Input = Train_Data WS_i $(1 < \ldots < i < \cdots < I)$
Models: AE, IF, VSM
- Train Models
Aggregation
- Save Aggregated Model
Input = Test_Data WS_i $(1 < \ldots < i < \cdots < I)$
Load Aggregated Model
- Detect Abnormal POs of WS_i
Calculation of Anomaly Rate of WS_i = $\frac{nbr\ of\ abnormal\ POs\ that\ have\ passed\ through\ WS_i}{Total\ nbr\ of\ PO\ that\ have\ passed\ through\ WS_i}$
is WS_i an abnormal Work Station ?
Yes if:
$Anomaly\ Rate\ of\ WS_i > Predefined\ Anomaly\ Threshold$
No if not

Diag. 2. Flow of the Industrial Application of the Anomaly Detection Approach

As part of applying the flow of the industrial application approach outlined in Diag. 2, in Appendix1 Fig. 2 (to access the Appendix1 GitHub repository, please use the following link: https://github.com/rida87/APMS/blob/main/Appendix1.pdf) we show a three-dimensional representation for the Conwip2_8_6 workstation, utilizing a real DB. This representation includes the components of the Input Error Vector (IEV). Each point in the figure corresponds to an observation in the Test_Data, comprising 4 POs, which have traversed workstation Conwip2_8_6 after $\text{Critical Date}_{(\text{Conwip2\_8\_6})}$. Notably, in this figure, only one anomaly was detected. We determine the anomaly rate depicted in Appendix1 Fig. 2 resulting in a percentage of anomalies equal to 25%. In Appendix1

Fig. 3, we further analyze each $\mathrm{IEV}_{(\mathrm{PO},\mathrm{WS_i})}$ represented in Appendix1 Fig. 2 by representing each pair of their components in a two-dimensional space. However, it is crucial to consider that the percentage of anomalies represented in Appendix1 Fig. 2, which is equal to 25% (the specific values of the observations of Appendix1 Fig. 2 are detailed in Appendix2 Table 3), is based on a relatively small Test_Data set comprising only four observations. This limited sample size may contribute to a higher anomaly rate, as even a single anomaly out of four observations can yield a significant percentage. Nevertheless, this does not discredit the quality or the anomaly detection performance of our model, since our anomaly detection model successfully identifies the third observation (Appendix2 Table 3) as abnormal, while the remaining observations are classified as normal. Notably, the third observation stands out with substantially higher values for First Error, Median Error and Last Error compared to the other observations, highlighting its anomalous nature. This underscores the model's ability to discern anomalies based on the magnitudes of error metrics. Conversely, the second and fourth observations, characterized by lower error values, are correctly classified as normal. Furthermore, the model provides its capability to differentiate between anomalies and outliers, as evidenced by the first observation (Appendix2 Table 3) which exhibits relatively high First Error and Median Error values but is not flagged as abnormal. This observation exemplifies the added value of aggregating classification models compared to single models. Without aggregation, this observation would have been flagged as abnormal in a single IF model, as its IF score is equal to 1 (Appendix2 Table 3). However, within our aggregated model, it remains unflagged due to its AE score of 0.74, which falls below the predefined AE anomaly threshold. This threshold, as outlined in Appendix2 Table 4 and equal to 0.79, represents the 90th percentile value among the AE scores derived from training the models using Train_Data (Appendix2 Table 4). Model effectiveness is evident through outputs and figures, highlighting its anomaly detection and data insight capabilities. In Appendix1 Fig.4 and Appendix1 Fig.5, we show additional anomalies detected at the Conwip2_11_3 workstation. Appendix2 Table 5 and Appendix2 Table 6 present errors and scores for Train_Data and Test_Data. In Appendix2 Table 6 anomalies are highlighted in red.

6 Conclusion and Perspectives

In conclusion, our paper extensively explores anomaly detection in production lines. We provide an overview of current research, followed by a detailed explanation of the database and a mathematical definition of anomalies. We describe the methodology employed in constructing our aggregated model, and apply our online approach to detect abnormal POs passing through workstations in a real production line, workstation by workstation.

Our anomaly detection model enhances dashboard systems by providing real-time, reliable alerts for anomalies in workstations. Utilizing statistical learning and dynamic segmentation, it improves efficiency and quality control by accurately distinguishing between outliers and true anomalies.

Our study recognizes the limitations of individual anomaly detection models in production lines: IF struggles with high-dimensional data, SVM requires sensitive parameter tuning, and AEs face challenges with complex data. We address these by optimizing parameters tailored to product, and we are currently testing another model that could offer better performance.

We plan to integrate Manufacturing Execution System (MES) data for better root cause analysis and are enhancing model reliability through ongoing labeling efforts. A good perspective includes predictive analytics like Predictive Failure modeling and Remaining Useful Life prediction to proactively anticipate workstation failures.

Acknowledgments. We thank the AURA (Auvergne-Rhône-Alpes) region for their crucial financial support, essential to the project's success. Gratitude is extended to WipSim Company for access to their valuable database.

Disclosure of Interests. The authors affirm that this article is original, unpublished, and not under consideration elsewhere. They have reviewed and approved the manuscript and declare no conflicts of interest.

References

1. Achraya, A., Singh, S., Pereira, V., Singh P.: Big Data, Knowledge co-creation and decision-making in fashion Industry. International Journal of Information Management. DOI: 10.1016/j.ijinfomgt. DOI:10.1016/j.ijinfomgt.2018.06.008 .(2018)
2. Chen, Das, D., Ivanov, A.: Building resilience and managing post-disruption supply chain recovery: Lessons from the information and communication technology industry. International Journal of Information Management, Elsevier, 49(C), 330–342. https://doi.org/10.1016/j.ijinfomgt.2019.06.002. (2019)
3. Dolgui, A., Ivanov, D., Sokolov, B.: Reconfigurable supply chain: The X-network. International Journal of Production Research, 58(13), 4138-4163. Received 23 Jan 2020, Accepted May 2020, Published online: 2020/07/11
4. Zipfel, J., Verworner, F., Fischer, M., Wieland, U., Kraus, M., & Zschech, P.: Anomaly detection for industrial quality assurance: A comparative evaluation of unsupervised deep learning models. Computers & Industrial Engineering, 177, 109045. https://doi.org/10.1016/j.cie.2023.109045 (2023)
5. Chevtchenko, S.F., Rocha, E.S., dos Santos, M.C.M., Mota, R.L., Vieira, D.M., de Andrade, E.C., de Araújo, D.R.B.: Anomaly Detection in Industrial Machinery using IoT Devices and Machine Learning: a Systematic Mapping. IEEE Access. November (2023).
6. Graß, A., Beecks, C., Soto, J.A.C.: Unsupervised Anomaly Detection in Production Lines. In: Beyerer, J., Kühnert, C., Niggemann, O. (eds) .Machine Learning for Cyber Physical Systems. Technologien für die intelligente Automation, vol 9. Springer Vieweg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-58485-9_3, (2019)
7. Abdelrahman and P. Keikhosrokiani.: Assembly Line Anomaly Detection and Root Cause Analysis Using Machine Learning. IEEE Access, 8, 189661-189672. doi: 10.1109/ACCESS.2020.3029826, (2020)
8. Massa, D., Valverde, R.: A Fraud Detection System Based on Anomaly Intrusion Detection Systems for E-Commerce Applications. Computer and Information Science, 7(2), 117-140. (2014)

9. García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E.: Anomaly-based network intrusion detection: Techniques, systems, and challenges. Computers & Security, 28(1–2), 18–28. (2009)
10. Pourhabibi, T., Ong, K.-L., Kam, B. H., & Boo, Y. L.: Fraud detection: A systematic literature review of graph-based anomaly detection approaches. Decision Support Systems, 133, 113303. (2020)
11. Markou, M., & Singh, S.: Novelty detection: A review - part 1: Statistical approaches. Signal Processing, 83(12), 2481–2497. (2003)
12. Chandola, V., Banerjee, A., & Kumar, V.: Anomaly Detection: A survey. ACM Computing Surveys (CSUR), 41(3), 15. (2009)
13. Mehrotra, K., Mohan, C., Huang, H.: Anomaly Detection Principles and Algorithms. DOI: 10.1007/978-3-319-67526-8, Corpus ID: 11903501.
14. Scrivano, S., Tolio, T.: A Markov Chain Model for the Performance Evaluation of Manufac turing Lines with General Processing Times. Procedia CIRP, 103(7), 20-25. DOI: 10.1016/j.procir.2021.10.002. (2021)
15. Liu, F. T., Ting, K. M., & Zhou, Z. H.: Isolation-based anomaly detection. ACM Transac tions on Knowledge Discovery from Data (TKDD), 6(1), 3. (2012)
16. Cortes, C., & Vapnik, V.: Support-vector networks. Machine Learning, 20(3). (1995)
17. Tschannen, M., Bachem, O., & Lucic, M.: Recent Advances in Autoencoder-Based Repre sentation Learning. Computer Science & Machine Learning. Retrieved from arXiv:1812.05069 (2018)
18. Grant, E. L., & Leavenworth, R. S.: Statistical Quality Control. McGraw-Hill Eductio(2018)