Improving the Productivity of a Production Line through Explainable Artificial Intelligence for Data Analysis

PELOUS Enzo
Université Grenoble Alpes
CNRS, Grenoble INP, G-SCOP
46 Av. Félix Viallet, Grenoble, 38000
enzo.pelous@grenoble-inp.fr

DAVID Pierre
Université Grenoble Alpes
CNRS, Grenoble INP, G-SCOP
46 Av. Félix Viallet, Grenoble, 38000
pierre.david@grenoble-inp.fr

GANNAZ Irène
Université Grenoble Alpes
CNRS, Grenoble INP, G-SCOP
46 Av. Félix Viallet, Grenoble, 38000
irene.gannaz@grenoble-inp.fr

SYLLA Abdourahim
Université Grenoble Alpes
CNRS, Grenoble INP, G-SCOP
46 Av. Félix Viallet, Grenoble, 38000
abdourahim.sylla@grenoble-inp.fr

***Abstract*—Improving the performance of industrial production systems through data analysis is a recurrent topic in both research and industry. However, most existing data-driven approaches in manufacturing focus on engineer level decision support, such as predictive maintenance or strategic resource planning and pay little attention to operator-centred decision support. At the same time, production data exhibit specific characteristics that make their analysis challenging: high-dimensionality, often heterogeneous, and naturally structured as time series and point processes. These gathered properties require dedicated modelling strategies insufficiently addressed in the literature.**

An additional and increasingly recognised challenge is the adoption of digital decision-support tools on the shop floor. Operators and line adjusters need to trust the models' outputs in order to integrate them into their daily practices. This calls for the design of explainable artificial intelligence approaches made for production contexts and for the needs of human operators.

The proposed approach is developed in a real factory, to support operators decisions based on the state of the production system.. We detail the decision model building process from dimensional reduction to combination with field knowledge.

Our goal is to build models that learn to recognise production situations similar to past ones that have already led to stops, so as to suggest relevant corrective actions to implement. A key aspect of our approach is the continuous confrontation of the modelling choices with field experience, in order to provide interpretable reasons for the recommendations.

***Index Terms*—Artificial Intelligence; Information Systems; Diagnosis**

I. Introduction

In this work, we study semi-automated production lines that operate with limited human intervention under nominal conditions, and where operators are mainly involved in critical situations such as extended production stops. The objective is not to react to a crisis, nor to redesign the physical infrastructure, but rather to **improve line productivity by reducing the average duration of production stops** through better use of the available production data.

Data analysis is widely used in industry, whether to improve manufacturing processes [2] or to enhance the reliability of preventive maintenance strategies [6]. Nonetheless, most existing contributions target decision levels above those of production operators, and develop tools intended primarily for engineers. For example, by optimizing maintenance schedules or predicting failures at the asset or plant level. By contrast, **few approaches explicitly address the needs of on site operators**, whose decisions must be taken under time pressure, on specific production lines, and with direct operational consequences.

Another important gap in the literature lies in the **nature of production data**. In many industrial settings, these data combine heterogeneous sources (alarms, sensor measurements, operator inputs, maintenance reports), are strongly **time-dependent** and each event can have a different consequence depending on the moment it occurs. These production data can naturally be modelled as **time series and point processes**, the former being continuous or discretely sampled, the second with a specific timestamp. Such data structures require appropriate statistical and machine learning frameworks; yet, methodological developments remain fragmented, and their integration into practical operator-support tools is still limited.

Finally, the **explainability** of AI-based decision-support tools has emerged as an important condition for their effective deployment in various Industry 4.0 contexts as discussed in [9]. Operators and line adjusters must be able to interpret model outputs, relate them to observable phenomena on the line, and understand the rationale behind recommended actions or detected problems such as the method described in [7]. Without such explainability, even accurate models may

face poor adoption in practice, which limits their impact on operational performance.

The industrial case considered in this work focuses on a factory that assembles and grinds metallic parts on identical production lines (consisting of several machines in series and in parallel), each supervised by a human operator. When a production line is stopped and the operator cannot restart it, a line adjuster is called. The adjuster must identify the cause of the stop and restore operation as quickly as possible. At present, this diagnostic and recovery process relies primarily on the adjuster's knowledge and experience of the specific production line.

The objective of the proposed approach is to leverage production data and expert knowledge in order to convert them into **operational recommendations** for operators and adjusters, as shown in Fig. 1. The global targeted functioning is from a stop detection to analyse the state of the production line (including the previous minutes) to be able to produce a prescription to the line Operator. Feedback from operators is continuously collected to improve the prescription model by updating it.

This article presents an end-to-end method from data collection and reduction to construct a production context representation suitable for machine learning, comparison with expert knowledge to ensure the logic of the approach and help future adoption and then the creation of a model capable of retrieving past problem-solving actions to new production situations.

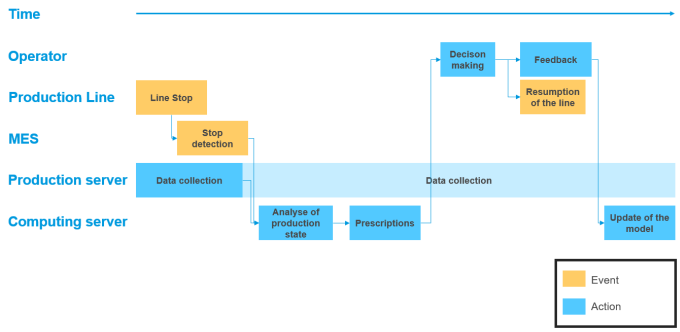

Fig. 1. Swimlane charts of the global approach

II. Data Description

Three main types of production data are available. At the initial stage of the study, data is collected over 2 months on 1 production line.

Alarm 1	01:00:03	ALARM
Alarm 2	01:01:03	ALARM
Alarm 2	01:01:32	CLEAR
Alarm 1	01:02:14	CLEAR
Alarm 3	01:08:07	ALARM

TABLE I

EXAMPLE OF ALARM DATA CONTAINING THE ALARM IDENTIFIER, TIMESTAMP, AND STATUS TRANSITION.

The first type describes the production context through **alarms**. When an incident occurs on a machine in the production line, an alarm is triggered, identifying the affected machine and providing a code that characterises the situation (similar to the systems presented in [3]) partially illustrated in Table I . The data format consists of lists of timestamps corresponding to the activation and return-to-normal times of each alarm. Several hundred distinct alarms are recorded for a single production line. The data are logged locally in real time, although some backups may be delayed by several hours or even days, making datasets difficult to synchronize.

01:00:03	-34	OK	NOK
01:01:04	-32	NOK	NOK
01:01:32	-30	NOK	NOK
01:03:14	-28	OK	OK
01:05:47	-32	OK	NOK

TABLE II

EXAMPLE OF SETTINGS DATA INCLUDING TIMESTAMP, NUMERICAL MEASUREMENT, AND BINARY QUALITY INDICATORS.

The second type of contextual data corresponds to the information returned by each machine at the completion of each part, which we refer to as **settings**. These data include, for example, cycle time, sensor measurements, and parameters modified by the operator as visible in Table II. They are stored jointly with the alarm data.

Finally, the third type of data originates from **stop declarations and adjuster reports** with both categorical labels and free-form textual notes such as the extract in Table III. When a line is stopped (no part has exited the last machine for a given time interval), the operator must classify the stop. In the event of an adjuster intervention, the latter records the actions performed on the line. These pieces of information are entered via a channel dedicated to communication external to the line and are stored more reliably on a remote server in real time.

Data collection, illustrated in Fig. 1, is continuously performed by an appropriate information system that collects relevant data on the production execution and associated events. One objective of this study is to access and process the collected data as soon as a stop is detected.

III. Methods

To implement the complete method described above each production context leading to a stop must be saved in a merged object described in section III-E these contexts (matrices of data) must grasp all the underlying dependence between the different machines on the line to allow similarity measure between different stops. Before the merger, each data source is analysed depending of its form in sections III-A, III-B and III-C to reduced the number of input and adapt the saving format to future use. This refined selection is confronted to expert knowledge in section III-D. Then, a machine learning model is built (section III-F) to retrieve recommendations for the operators and the explainability of the models' prescriptions will be obtain using machine learning explainability methods (section III-G).

02:00:03	02:01:05	Machine A	Cleaning of sensor A	Ø
03:51:25	03:55:18	Machine B	Other	Hard Reboot and adjustment of operating speed
04:44:17	12:39:57	Complete Line	Preventive maintenance	Ø

TABLE III
EXAMPLE OF STOP DECLARATIONS INCLUDING TIMESTAMPS, MACHINE IDENTIFIER, STOP CATEGORY, AND OPERATOR COMMENTS.

A. Alarm Modelling and Reduction

Alarm data are modelled as **temporal point processes**, as defined in [8] and represented for prediction tasks in [4] with approximately 500 distinct alarms recorded on a single production line. For each alarm, an activity rate is computed, defined as the number of minutes during which the alarm is active divided by the total observation time. It appears that only 11% of alarms have an activity rate higher than the average as can be seen in Fig. 2. An analysis is then conducted on a machine-by-machine basis in order to maintain a coherent representation of the production context along the entire production line. For a first reduction step, the number of alarms is limited to five (this number is chosen arbitrarily) per machine, by retaining the most active ones.

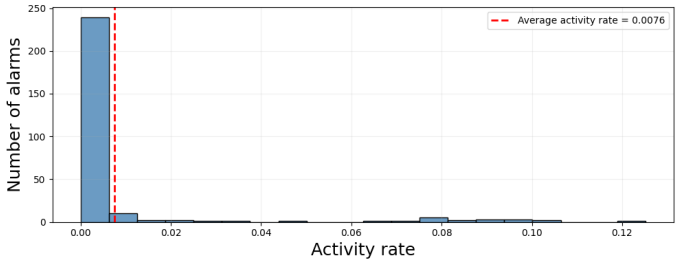

Fig. 2. Breakdown of alarms by activity rate

B. Settings Analysis and Dimensional Reduction

Regarding the settings, the problem is closer to **process monitoring**, as in [14], with also around 500 settings on the production line. The objective, however, is not fault or drift detection, but the reduction of the number of variables required to describe the state of a production line because the goal is to react to an undergoing stop rather than predicting one. Correlation coefficients are computed between the different setting variables, with sometimes very high correlation levels (for instance, 41% of correlation values exceed 0.9 for one of the assembly machines at the end of the line). This threshold is chosen arbitrarily but the method is largely inspired by the work of [17] and might might evolve over time depending on the performance of the model constructed later and the field feedback.

C. Analysis of Stops and Textual Information

For production stops, the structured data (choice of machine, standardised stop reason) are already normalised. The richest but least structured part of the information lies in the **free-text comments** entered by operators and adjusters, as they describe the actions taken to restart the production system. In a first step, a simplified text analysis is implemented, at a less advanced level than some studies in the literature [16]. A vector representation of words, as proposed for example in [12], is used to identify the most similar sentences. These vector representations can then be used by the model described below to find similarities between stop labels.

D. Expert Involvement and Variable Selection

The reduction criteria obtained from the statistical analyses are then systematically confronted with **field expertise**. On the alarms data the question is : which alarms can be identified by the operators during production ? Regarding the settings : which data are monitored by the operators and adjusters before and during a stop ? About the stop qualification : what operators perceive as similar between two stops ? ?. Interim results are presented to managers and to the various actors along the production line, in order to discuss the relevance of the selected variables and, where appropriate, adjust the selection. The involvement of future users is maintained throughout the development process, with a dual objective: fostering **appropriation** of the tool and enhancing the **explainability** of the results. This participatory approach is consistent with the conclusions of [9] on the importance of involving end-users in explainable AI projects.

To ensure the relevance of the subset of features selected, another subset of values is carried out all along the same pipeline from context creation to the decision making model to compare the results of the approach. This subset is directly constructed from the settings and the alarms an operator can see and used during production. After constructing both data sets we calculated the overlap between the two. With almost three times as much alarms and twice as much settings, the statistical set contains all the indicators used by the operators when the production line is running. This supports the conclusions of our preliminary analysis, although this does not constitute a formal validation

Expert knowledge is also used at a second level, beyond feature selection, to refine the notion of similarity between stops. Indeed, similarity cannot be entirely inferred from raw production logs or structured labels such as the machine, the category or the duration of the stop. In practice, the operational relevance of a retrieved past case depends on whether an expert would judge the corresponding situation as truly comparable in terms of diagnosis and corrective action. For this reason, a subset of stop pairs is manually re-annotated by a field expert using a four-level ordinal scale ranging from 0 (“no similarity”) to 3 (“almost the same stop”). This re-annotation process provides a more operational notion of similarity than the one derived solely from structured labels, and it is later used to evaluate the recommendation system.

At this stage, the variables retained to characterize a stop remain the machine involved, the stop category, the duration, and the free-form notes, in this order of priority. However, the expert re-annotation allows us to go beyond this initial proxy and to assess whether the learned latent space effectively captures similarities that are meaningful for production staff.

E. Low-level Data Fusion and Context Representation

Based on the reduced data, a **low-level data fusion** step is carried out, in the spirit of the methods reviewed in [5]. The aim is to link each stop episode to its production context, described in particular by the alarms and settings observed during the period preceding the stop (for example, the 2 previous hours). For alarms, the ON/OFF status of the selected variables is sampled every minute. For settings, one value per minute is also retained; in the absence of a new part during the interval, the last observed value is propagated. Each stop is thus associated with a matrix of dimension

$$\text{number of time steps} \times (\text{number of alarms} + \text{number of settings}),$$

providing a detailed description of its context. In our case 120 minutes are considered to build the context.

F. Learning and evaluating a similarity space for stop recommendation

The next step is to define a representation of production contexts that makes it possible to retrieve past stops similar to a newly observed one. The final objective is not only to cluster stop situations in a latent space, but to support operators and line adjusters by recommending previously encountered situations together with the corrective actions that were historically taken in similar contexts.

Each stop episode is represented by a multivariate temporal context built from the production data observed during a fixed time window preceding the stop. In our case, each stop i is associated with a matrix $X_i \in \mathbb{R}^{T \times F}$, where T is the number of time steps and F the number of selected alarms and settings after reduction and fusion. The goal is to learn an encoder f_θ that maps each stop context X_i to a latent vector $z_i = f_\theta(X_i) \in \mathbb{R}^d$, such that contexts corresponding to operationally similar stops are close in the latent space.

1) Initial representation learning from stop contexts: To encode the temporal dimension of stop contexts, we use a single-layer LSTM [10] followed by a linear projection onto a latent space of dimension d. This choice is motivated by the sequential nature of the data and by the ability of LSTM architectures to handle temporal dependencies while mitigating the vanishing and exploding gradient issues commonly encountered in standard recurrent neural networks as explained in [1].

In the first version of the approach, the encoder is trained through a triplet-based objective. For each anchor stop i, a positive stop j and a negative stop k are selected according to a proxy notion of similarity derived from the available labels. Let $z_a = f_\theta(X_i)$, $z_p = f_\theta(X_j)$ and $z_n = f_\theta(X_k)$ be the corresponding embeddings. The model is trained by minimizing the standard triplet margin loss:

$$L_{\text{triplet}} = \max\left(0,\ \|z_a - z_p\|_2 - \|z_a - z_n\|_2 + m\right),$$

where $m > 0$ is a margin hyperparameter. This objective encourages the embedding of the anchor to be closer to the positive than to the negative by at least the margin.

This first stage is useful to structure the latent space and to obtain compact contextual representations. However, it relies on a coarse approximation of stop similarity, which remains only partially aligned with the final recommendation task.

2) Proxy similarity derived from structured labels: The initial triplet generation relies on a similarity function built from the stop descriptors available in the information system. Each stop i is associated with a structured label:

$$y_i = (\text{machine}_i,\ \text{category}_i,\ \text{duration}_i,\ t_i),$$

where t_i denotes a vector representation of the free-form notes provided by operators and line adjusters. The textual component is represented using TF–IDF weighting [12], and textual similarity is computed using cosine similarity between the corresponding vectors.

The global similarity between two stops initially combines several criteria: equality of the machine, equality of the category, proximity in duration, and textual similarity between free-form comments. In practice, the first implementation remains strongly driven by the machine identity, while the other descriptors play a more limited role. This proxy is sufficient to bootstrap a first latent representation, but it remains too coarse to faithfully capture the practical notion of similarity required by operators and line adjusters.

3) Expert re-annotation of stop pairs: To better align the model with the operational objective, a subset of stop pairs is manually re-annotated by a field expert. For each pair, the expert assigns an ordinal similarity score on a four-level scale:

- 0: no similarity,
- 1: low similarity,
- 2: medium similarity,
- 3: high similarity or almost the same stop.

This annotation is performed by presenting the expert with the main characteristics of both stops, including their contextual descriptors and free-form notes. The purpose of this re-annotation is to capture a practical notion of similarity, i.e., whether two stop situations would be considered comparable from the point of view of diagnosis and corrective action.

This expert-annotated dataset serves two purposes. First, it allows us to confront the proxy similarity derived from labels with a more operational judgement. Second, it provides a dedicated evaluation set for the recommendation system.

Finally, once the method has been rolled out, operators will be able to annotate the similarity between new stops and the recommended ones on an ongoing basis. This will enable both user involvement and the updating of the model.

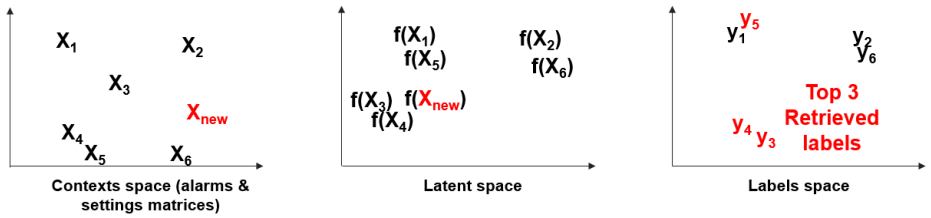

Fig. 3. Top 3 Similar labels retrieval when a new stop's context is presented to the model

4) Recommendation-oriented retrieval: Once the encoder has been trained, each stop context is represented by a latent embedding. Given a new stop, the system computes its embedding and retrieves the most similar past stops according to the cosine similarity between latent vectors. Cosine similarity is chosen because it is well suited to compare normalized latent representations and to rank neighbours in an embedding space.

The retrieval mechanism is designed to support the final use case: when a new stop occurs, the system proposes a ranked list of past situations whose contexts are close to the current one. The associated notes and historical corrective actions can then be presented to the operator or line adjuster as candidate recommendations.

5) Evaluation protocol and metrics: The initial learning phase is monitored through standard optimization metrics such as training and validation losses. However, these metrics remain difficult to interpret from an operational perspective. Indeed, a low triplet loss does not directly indicate whether the retrieved neighbours are useful for production staff.

For this reason, we complement the representation-learning evaluation with a recommendation-oriented protocol based on the expert re-annotated stop pairs. Retrieval performance is assessed using top-k metrics, namely Precision@k and Recall@k. For a given query stop, the model retrieves the k most similar past stops according to the cosine similarity (chose according to the survey presented in [11]) between latent embeddings. The retrieved neighbours are then compared with the pairs judged relevant by the expert annotation.

This evaluation framework is closer to the actual purpose of the system (the method is illustrated Fig. 3) than loss-based optimization metrics alone, since it directly measures the capacity of the model to retrieve operationally meaningful past cases.

G. Explainability

Even if the input features have been selected according to the expert analysis conducted on the field it might be difficult to propose several actions to the operators without pointing out the most important input data. With an encoder based on a LSTM block one solution could be to apply the SHAP method (presented in [13]) such as proposed by [15]. The only limitation of the method , in our application, is that the importance of each feature cannot be directly computed at the inference. The idea, then, is to present the result of the method on critical examples to the field experts throughout the use of the tool on the production line.

When giving to operators and adjusters former stops similar to the one occurring they might be more incline to try the historical solutions if they are convinced that the model is able to capture the important features.

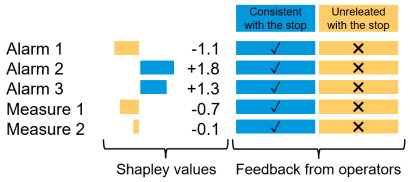

Fig. 4. Illustration of SHAP-based feature importance analysis for comparison with expert feedback. (in this example we hope that the alarms 2/3 make sense in this stop context)

IV. Preliminary Results and Limitations

The first training experiments show that the model is able to organize stop contexts in a latent space and to reduce the ranking loss over the training set. Validation loss follows a broadly similar trend, suggesting that the model captures part of the structure underlying the stop contexts. In addition, the ranking accuracy indicates that the model learns, to some extent, to place contexts judged more similar above contexts judged less similar for the same anchor stop.

However, these optimization-oriented metrics remain insufficient to assess the practical value of the system. In particular, a lower training or validation loss does not necessarily imply that the retrieved neighbours are useful from the point of view of operators and line adjusters. This is a critical issue in the present application, where the final objective is not classification but recommendation: for a new stop, the system must retrieve past situations that are operationally relevant and likely to suggest appropriate corrective actions.

For this reason, the learned representation is also evaluated in a retrieval setting using the expert-annotated stop pairs introduced in Section III-D. A subset of stop pairs was manually assessed on a four-level similarity scale from 0 to 3, and these annotations were used to define relevant neighbours in the evaluation phase. In the current experiments, stops annotated with similarity levels 2 and 3 were considered relevant neighbours for retrieval evaluation. Table IV reports the retrieval performance obtained on the corresponding test queries.

The current results remain quite modest, the primary objective at this stage is to validate the methodology. Precision@k

stays very low for all tested values of k, while Recall@k increases with k, indicating that some relevant neighbours can be found when expanding the number of retrieved stops, but that they are not yet ranked consistently among the very first positions. In other words, the model begins to capture part of the similarity structure between stops, but the resulting recommendations are still not accurate enough for deployment (the two metrics are displayed in the Table IV).

k	Precision@k	Recall@k
1	0.0526	0.0526
3	0.0526	0.0789
5	0.0526	0.1184
10	0.0526	0.2018

TABLE IV
MODEL PERFORMANCE ON THE 19 REQUESTS OF THE TEST SET

Several directions for improvement are therefore envisaged.

A first priority is to enrich the expert-annotated dataset. Rather than annotating isolated random pairs, future annotation campaigns should focus on groups of candidate neighbours around the same query stop, in order to provide denser and more informative local rankings.

A second direction concerns the explainability of the recommendations. Beyond the retrieval of similar stops, the system should highlight the contextual features and temporal patterns that contributed most to the recommendation, so that operators and line adjusters can better understand the rationale behind the retrieved cases.

Finally, the visualization of the learned latent space and the interactive exploration of neighbouring stops appear as promising complements to numerical evaluation. Such tools could help experts assess whether clusters and neighbourhoods correspond to meaningful operational situations, and thus support both validation and appropriation of the recommendation system.

At this stage, the complete pipeline has been validated offline, but the retrieval performance is still insufficient to consider an on-site deployment, even as a prototype. Nevertheless, the introduction of expert re-annotation and retrieval-based evaluation constitutes an important step towards a more operational and explainable similarity-based decision-support system for production stops.

V. CONCLUSION

This study presents an end-to-end method (From the production line to production workers) to use production data in order to recommend operational corrective actions, found in the production logs, to operators and adjusters to limit the stop duration.

The first step consists in collecting data and selecting the most representative features with a statistical analysis confronted to expert use of them. The goal is to minimise the number input fields to make it easier to identify the root causes but also to transform human inputs to more machine-readable representations.

Secondly a data set of characterised stops and the context of these stops is created. This data set is fed to a machine learning model aimed at learning a latent space capable of grouping operationally similar stops within the latent space.

Once the latter model reaches satisfactory performance, it must be able to retrieve similar labels for each new stop and assist operators and line adjusters in restoring production as quickly as possible.

The last point is still a work in progress.

REFERENCES

[1] A. A. Adekunle, I. Fofana, P. Picher, E. M. Rodriguez-Celis, O. H. Arroyo-Fernandez, and R. Zemouri. Optimizing deep learning predictive models: A comprehensive review of rnn and its variant architectures, 12 2025.

[2] S. N. Alasadi and R. Al-Sabur. Machine learning and exploratory data analysis for predicting tensile and thermal responses in friction stir spot welding. *International Journal of AI for Materials and Design*, 2:37, 12 2025.

[3] A. Bauer, A. Botea, A. Grastien, P. Haslum, and J. Rintanen. Alarm processing with model-based diagnosis of event discrete systems. Technical report, 2011.

[4] D. W. J. C. and Page. Forest-based point process for event prediction from electronic health records. In Kristian, N. Siegfried, Železný Filip Blockeel Hendrik, and Kersting, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 547–562. Springer Berlin Heidelberg, 2013.

[5] F. Castanedo. A review of data fusion techniques, 2013.

[6] X. Duan, A. Vasudevan, E. T. Bekar, K. Gandhi, and A. Skoogh. A data scientific approach towards predictive maintenance application in manufacturing industry. In *Advances in Transdisciplinary Engineering*, volume 21, pages 292–303. IOS Press BV, 4 2022.

[7] G. M. Díaz. Comparative analysis of explainable ai methods for manufacturing defect prediction: A mathematical perspective. *Mathematics*, 13, 8 2025.

[8] M. Farajtabar, Y. Wang, M. Gomez-Rodriguez, S. Li, H. M. Stewart, H. Zha, and L. Song. Coevolve: A joint point process model for information diffusion and network evolution * 3. Technical report, 2017.

[9] M. Garouani and . Ai. *Towards Efficient and Explainable Automated Machine Learning Pipelines Design : Application to Industry 4.0*. PhD thesis, 2022.

[10] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[11] A. Jadon and A. Patil. A comprehensive survey of evaluation techniques for recommendation systems. 1 2024.

[12] C.-Z. Liu, Y.-X. Sheng, Z.-Q. Wei, and Y.-Q. Yang. Research of text classification based on improved tf-idf algorithm. Technical report, 2018.

[13] S. M. Lundberg and S. I. Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 2017-December, 2017.

[14] A. Melo, M. M. Câmara, and J. C. Pinto. Data-driven process monitoring and fault diagnosis: A comprehensive survey, 2 2024.

[15] J. E. T. Radjabaycolle, E. M. C. Wattimena, and V. E. Pattiradjawane. Improving air quality forecasts with lstm and shap explainability: A case study in jakarta. *Journal of Embedded Systems, Security and Intelligent Systems*, 2025.

[16] E. Tonkin, E. Tonkin, G. J. L. Tourte, and G. Harris. *Working with text: tools, techniques and approaches for text mining*. Chandos Information Professional Series. Chandos Publishing, Amsterdam, Netherlands, 2016.

[17] F. Yang, S. L. Shah, D. Xiao, and T. Chen. Improved correlation analysis and visualization of industrial alarm data. *ISA Transactions*, 51, 2012.