TNO innovation for life

Probabilistic graphical models for performance diagnostics

Methods and applications to a print quality case

TNO Public › TNO 2025 R13164
17 March 2026

ICT, Strategy & Policy
www.tno.nl
+31 88 866 50 00
info@tno.nl

TNO 2025 R13164 – 17 March 2026

Probabilistic graphical models for performance diagnostics

Methods and applications to a print quality case

Author(s)	Leonardo Barbini, Alvaro Piedrafita Postigo, Micha Lipplaa
Classification report	TNO Public
Title	TNO Public
Report text	TNO Public
Number of pages	32 (excl. front and back cover)
Number of appendices	0
Project name	Carefree 2025
Project number	060.62910/01.01

Acknowledgements
The research is carried out as part of the Carefree project under the responsibility of TNO-ESI with Canon Production Printing as the carrying industrial partner. The research activities are co-funded by TKI HTSM via the PPP Innovation Scheme (PPP-I) for public-private partnerships.

Summary

This report introduces a methodology for diagnosing performance issues in high-tech production systems. The approach uses probabilistic graphical models to combine machine data with expert and design knowledge. By reasoning across multiple representations of the data, multiple resolutions, and across time, the approach can interpret defectivity patterns and infer their actionable root causes. Validation on an industrial inkjet printing case, focused on diagnosing patterns of non-jetting nozzles resulting in print quality issues, shows that the method achieves high diagnostic accuracy (86%) and aligns well with experts' expectations. The study presented in this report highlights the importance of scoped modeling for such complex diagnosic tasks, as well as the need for multi-representation reasoning, and expert involvement, and identifies opportunities for future improvements such as automatic learning of model parameters and handling continuous (random) variables.

Contents

1 Introduction

A number of companies in the Netherlands are global leaders in the design, manufacturing, and servicing of various high-tech production systems [1]. These are complex cyber-physical systems operating under stringent requirements for throughput and output quality. To meet these requirements, the systems are becoming increasingly complex, both in terms of hardware and software. Consequently, diagnosing a system that has incurred a problem presents a significant challenge for human personnel. This diagnostics root-cause analysis process can be significantly streamlined through digital assistance methodologies and accompanying tools. This report presents one of such methodologies.

High-tech production systems typically encounter a range of problems that can be broadly categorized into two main classes: hard-down situations and performance issues. In the former the system has completely stopped working, while in the latter the system is still operating but below specification. For more detailed information on the variety of problems that can arise, as well as other aspects pertinent to the diagnostic task within the high-tech domain, we refer the reader to [2]. This report introduces a methodology for the assisted diagnosis of performance issues, we refer the reader to [3, 4] for a methodology for hard-down issues.

Specifically, the focus of this report is on performance issues that manifest as detectable defectivity patterns on the product of a high-tech system, i.e. a deviation from the required quality at different locations on the product. Two from the many examples of performance issues in scope of the presented methodology are: print quality artifacts on a page printed by an industrial printer, and overlay problems on a silicon wafer in semiconductor manufacturing system. Illustrative examples of these performance issues are shown in Figures 1.1 and 1.2 respectively. More broadly, this methodology could be applied to patterns in derived domains, such as those produced via Fourier transform, rather than the spatial domain alone. This approach was not explored in this report, as spatial analysis proved sufficient for the current diagnostic requirements.

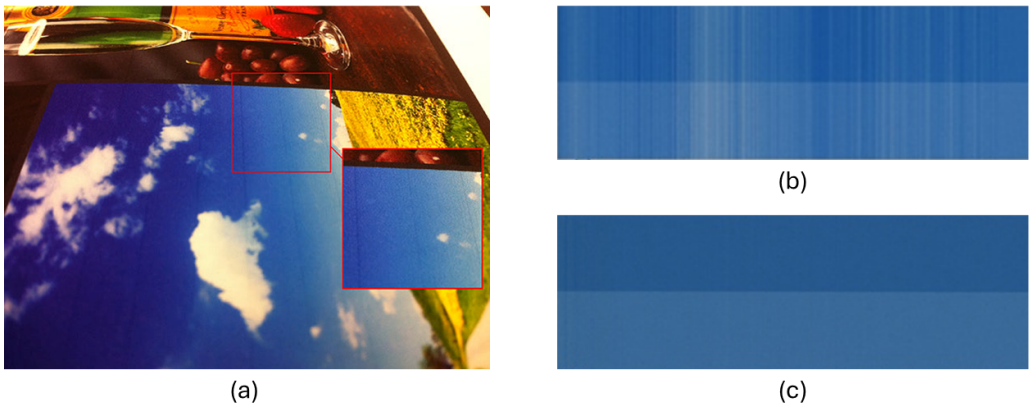


Figure 1.1: Example of performance issues in inkjet printing: print quality artifacts visible on a printed page. (a) Dark lines, from [5]. (b) Stripe artifacts (white lines) and corresponding desired output in (c), provided by Canon Production Printing.

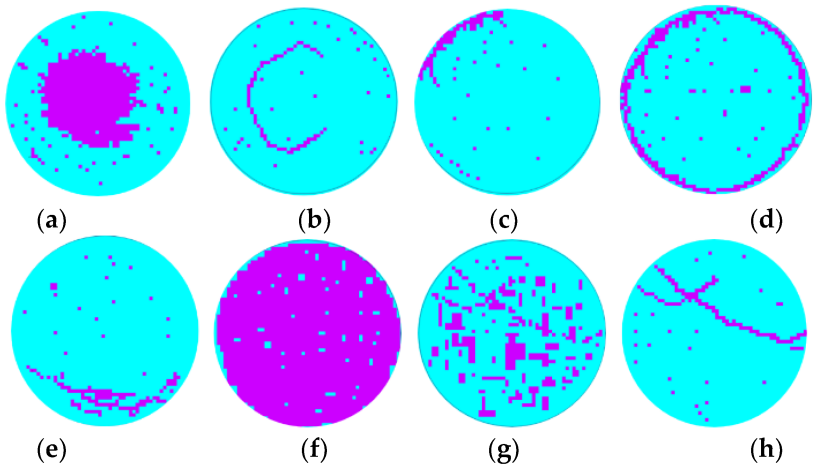


Figure 1.2: Example of performance issues in semiconductor manufacturing: eight common patterns of defective (shown in purple) dies on a wafer, from [6].

The diagnostic objective of the methodology presented in this document is to identify the actionable root causes corresponding to observed defectivity patterns. An actionable root cause is defined as one for which a specific remediation step, either automated or manually executed, can be applied to the system to eliminate or mitigate the cause. The identification of these actionable root causes facilitates a subsequent process optimization step. This involves selecting the most appropriate corrective action to restore the required product quality specifications. The established link between root cause and action (which is system or industry-specific and thus outside the scope of this document) ensures that the overall process maximizes system up-time.

The methodology presented in this document was developed in collaboration with Canon Production Printing (CPP) in the CareFree project [7]. Although initially created for the industrial printing sector, this methodology is generic and fully applicable to other high-tech industrial domains facing similar performance issues.

1.1 Research question

The activities throughout the research presented in this document were guided by the research question presented below[1]:

- How can we define and validate a scalable methodology for diagnosing performance issues that accounts for both discrete and continuous failure root causes, and for their compositions?

This question establishes several criteria for the development of methodology. Validation requires that the approach is successfully applied to real-world field data coming from complex high-tech systems, rather than relying solely on simple toy examples. Scalability requires the method to be both computationally and practically (implementation wise) feasible. It should handle performance diagnosis in complex industrial systems, potentially involving hundreds or thousands of interacting factors.

[1] From the CareFree project plan [7]

Furthermore, the methodology must handle two different classes of failure root causes as these both manifest in high-tech systems. Discrete root causes of performance issues, which are described by a set of distinct, countable states, e.g., a component is either working or failing, like a motor being either rotating or not rotating. Continuous root causes which are described by a varying range of state, e.g., gradually increasing environmental factors, such as rising pollution or temperature fluctuations in the operational environment of an industrial system.

1.2 Organization of this document

The document is organized as follows. Chapter 2 discusses the relevant state of the art and practice in the context of diagnosis of defectivity patterns from industrial systems. Then Chapter 3 presents the methodology introduced by this document, discussing mathematical details, key features and assumptions, and limitations. Chapter 4 details the methodology's application and validation through a real industrial case study from CPP: the diagnosis of print quality artifacts due to non-jetting nozzles in an industrial inkjet system. The presentation is kept generic, excluding CPP-specific details, to emphasize the methodology's generic applicability. Finally, Chapter 5 concludes the document with a summary of the results as well as directions for future work.

2 Background

This chapter details the state of the practice and the state of the art in the performance diagnostics of defectivity patterns on products in the context of high-tech industrial production systems.

2.1 State of practice

Current industrial practice for diagnosing defectivity patterns is primarily characterized by a reliance on expert-based approaches: engineers use their domain knowledge to interpret complex dashboards with visualizations of measured data and estimated features. This practice ultimately leaves the identification of a performance issue's root-cause to human judgment [2]. While leveraging engineering expertise, this approach is limited by the burden posed to experts and does not scale with the increasing complexity of modern systems and with the installed base of systems, i.e. not enough engineering expertise to cover all cases.

These limitations of current practice are compounded by how diagnostic knowledge is acquired and applied. During the design and engineering phases of the machine lifecycle, limited effort is typically dedicated to formal performance diagnostics, with engineers primarily utilizing high-level tools such as fishbone diagrams to anticipate (performance) failure modes. Consequently, the vast majority of performance diagnostic knowledge is learned only after the machines are operational in the field. When a difficult performance issue arises, diagnosis often relies on an inherent escalation based approach, where the problem moves sequentially through tiers of support, starting with on site personnel and culminating in the involvement of highly specialized product experts. This sequential escalation is a major driver of high time-to-resolution and increased operational cost.

Another aspect of the current practice is the natural tendency for human experts to analyze individual defect contributors in isolation and to confine the search for the root cause to the corresponding single process where a fault is suspected. However, this localized view is insufficient because the observed defectivity patterns are often an aggregate effect resulting from interactions across multiple processes, components, and the product itself. The difficulty of performing simultaneous, system-level reasoning across all these interconnected aspects remains a challenge in current industrial practice.

2.2 State of the art

Current research primarily focuses on developing data-driven methods, notably deep learning [8]. The applicability of these techniques in classification of defectivity patterns critically depends on availability of labeled data, leading to a key division: supervised versus unsupervised or self-supervised techniques. Where large labeled datasets exist, supervised learning methods achieve successful results. For example, in the identification of wafer map failure patterns in semiconductor manufacturing. These maps are highly informative because specific equipment faults or process problems generate distinct, recognizable geometric

[2] The focus here is on difficult performance issues, as easy deviations are typically handled by built-in control loops that compensate for minor variations.

patterns, e.g., ring, scratch, random. By training on extensive, well-curated labeled historical data, such as the public WM-811K dataset [9], deep learning methods [6] obtain high-accuracy defect detection and classification. However, such complete training datasets are rarely available. Most industries struggle with the effort of creating a sufficiently large labeled dataset. This is a difficult organizational task, complicated by the highly skewed nature of industrial data. Performance issues are sparsely distributed: over time (a system is mostly normal), across systems (a few systems account for most issues), and across root causes (failure patterns result from many diverse, infrequent root causes). This sparsity, combined with the fact that the full spectrum of possible root causes only manifests over long-term, real-world operation, makes collecting large, reliably labeled datasets for supervised learning challenging.

In these data-scarce situations, supervised methods are not applicable, therefore research focuses on self-supervised or unsupervised methods like Autoencoders (AEs) and Variational Autoencoders (VAEs) [10, 11]. These reconstruction-based models can be trained on unlabeled data. Root cause classification is then attempted by clustering in the learned low-dimensional latent space (the compressed features in the AE's bottleneck layer), aiming to cluster cases with similar defectivity patterns [11]. However, these data-driven approaches often prove suboptimal as they run a significant risk of learning failures specific to a particular system rather than generic diagnostic patterns transferable across multiple systems.

Given these difficulties of applying purely data-driven methods, current research, is shifting its focus to hybrid approaches that combine available data with domain knowledge. Knowledge typically consists of two types: about the system's design [4, 12], and expert knowledge derived from historical failure analysis. For these hybrid approaches Probabilistic Graphical Models (PGMs) are particularly well-suited, as they provide a framework to explicitly combine these diverse knowledge sources with observational data [13, 14, 15]. The application of PGMs, especially Markov Random Fields (MRFs), is extensively documented in two-dimensional pattern recognition and image processing tasks, where they model spatial dependencies between neighboring pixels [16]. It follows that MRFs are directly applicable to the diagnostics of defectivity patterns, which are essentially 2D image data, as for the examples of wafer maps and print quality artifacts in Figures 1.1, 1.2. Specifically, in the semiconductor sector, MRFs have been applied in pattern recognition and defect classification for semiconductor manufacturing [17, 18]. In this document, we present such a methodology relying on PGMs for the root cause analysis of defectivity patterns, specifically designed for situations characterized by scarce labeled data but available system design and expert knowledge on failure patterns.

3 Methodology

This chapter is adapted from [19].

3.1 Description of the problem

In this report we address the problem of diagnosing performance issues affecting the quality of the products manufactured by a high-tech system. Issues relating to the performance of the machine in terms of productivity or efficiency are therefore out of scope. For the remainder of this report, when we use the nomenclature ”performance issue”, we refer to the first class of issues, i.e. issues affecting the quality of the product.

A performance (read, product quality) issue typically manifests in patterns of defects on the products themselves, also known as defectivity patterns. Defectivity patterns are caused by cyber-physical processes stemming from the product substrate, the equipment, the fabrication process or the equipment maintenance process. For example, defects on a wafer could be caused by irregularities on the wafer surface, malfunctions of the light-source, defects on a mirror, delays on the timing of the stepper, incorrect calibration of the machine after maintenance etc.

The characteristic features of the different causes for product defects can vary significantly. Some are instantaneous or product specific, such as defects on the substrate, some evolve in time, such as timing delays, residue buildup within the machine, equipment deformation, etc. Some have slow dynamics, some are fast. Some failures lead to localized defectivity patterns, some lead to global patterns. Finally, some are actionable, meaning that they can be fixed or prevented, while others are unavoidable. In addition, multiple such defectivity causes could be present in one single product.

While all such processes are potentially occurring within the machine or the product, we only have access to localized measurements on the product itself. For example, in a high-volume industrial printer, the state of the individual nozzles on a printhead is not measured, but rather, one measures if ink ejected from a nozzle is on the expected position on a test page, and is left with the task of attributing the cause of an out-of-bounds measurement. The problem at hand is then one of classification and interpretation. Respectively, we want to attribute each defective measurement to one or multiple classes, and to relate the classes to failure mechanisms in order to implement the optimal maintenance action.

3.2 Description of the methodology

Our methodology aims to classify observations taken over time and at different positions of a product into a discrete set of possible root causes. The proposed approach is based on two key assumptions. First, we assume that observations of the product are discrete (e.g., OK when the measurement is within accepted bounds, and NOK when the measurement is out of bounds) and indirectly reflect the true state of the hardware, processes, and the product itself. Second, we assume to have knowledge of failure behaviour, i.e. root causes, and of the system design, such as component layout or process geometry.

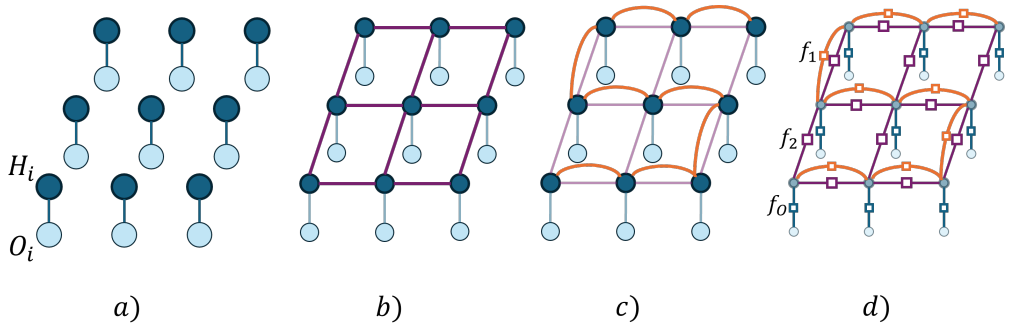


Figure 3.1: Construction of the CRF for a system with two representations. One starts with defining the variables and proceeds to add edges one type at a time. Light blue circles are observed variables and dark blue circles are hidden variables. Blue edges correspond to the observation factor f_O, purple edges correspond to one of the 2D-factors f_2 and orange edges correspond to 1D-factors f_1.

This classification task is reminiscent of the problem of image segmentation, where the goal is to label every pixel in a 2D image into a set of classes. Inspired by the literature on image segmentation [16], we use probabilistic graphical models (PGMs) in the form of Conditional Random Fields (CRFs) to model our system, and reason about it, in probabilistic terms. Our assumptions contain all the necessary ingredients to frame our problem in the language of PGMs [14]. In contrast with the square grids used in the image segmentation literature, we will make use of CRFs with arbitrary topology.

3.2.1 Mapping to a probabilistic graphical model

Let us take as an example a system where the observations are arrayed on the product in a 2D grid (think dies on a wafer, or points on a printed page) but also connected along a 1D path (e.g. exposure path, jetting order). Both the lithography and industrial printing processes mentioned in Chapter 1 are of this type. The construction of the PGM proposed by our framework is outlined in Figure 3.1.

First, we model each localized observation O_i as an observed random variable taking discrete values $\{OK, NOK\}$, see light blue nodes in Figure 3.1. As an example, these could be the points printed on a test page, which can be in the expected position (OK), or out of the acceptable bonds (NOK). The classification of observation O_i is then modelled as another discrete random variable H_i, this time hidden or unobserved, taking values in $\{c_1, ...c_k\}$, the set of root causes, of which we assume to have prior knowledge. These are depicted as dark blue nodes in the figure. In the printer example, the causes of a misprinted point could be, for instance, 'defective paper', or 'low ink'. Edges in the graph represent our belief that the two variables are directly correlated. We begin by adding the observation edges connecting each O_i and H_i pair, see blue edges in Figure 3.1-a. Next, we add edges between different H_i's. These edges encode proximity in one of the design-informed distance metrics. For instance, NOK observations caused by defective paper would be very correlated with their "neighbours" on the paper. We call the set of edges induced by each distance metric a representation of the observations. In Figure 3.1-b we add the purple edges induced by the grid-like 2D structure on the product, while in Figure 3.1-c we add the orange edges induced by the 1D path.

To make a conditional random field, we place factors f_j on the edges between variables. These factors are numerical arrays quantifying the strength of correlation between the discrete states in the variables they connect. These factors are depicted in Figure 3.1-d with empty squares. In this work we use design knowledge and domain expertise to determine

their values. Equation (3.1) gives an example of how we use one-hot encodings to represent discrete random variables.

$$\begin{aligned} O_i \in \{OK, NOK\} &\cong \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}, \\ H_i \in \{h = c_0, c_1, c_2\} &\cong \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\} \end{aligned} \tag{3.1}$$

Specifically, it shows random variable O_i with two states $\{OK, NOK\}$, represented as orthogonal unit vectors in a 2-dimensional space; together with a random variable H_i with three states $\{h, c_1, c_2\}$ corresponding respectively to healthy state and root causes c_1 and c_2, represented as orthogonal unit vectors in a 3-dimensional space.

Let $f_O(i) = f_O$ in Eq. (3.2) be the factor connecting the variables O_i and H_i in Figure 3.1-d for any index i.

$$f_O = \begin{matrix} \\ O_i = OK \\ O_i = NOK \end{matrix} \begin{matrix} \begin{matrix} H_i = h & H_i = c_1 & H_i = c_2 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \end{matrix} \tag{3.2}$$

Rows and columns of the factor correspond to the states of the variables O_i and H_i respectively. The entries of the array $f_O[j,k]$ indicate the strength of correlation we attribute to the combination of variable's states indicated by the j-th row and k-th column. Within each factor, only the relative differences matter, although it is good for the modeller to keep consistency across factors as much as possible. For the factor $f_O[j,k]$, the first row encodes the fact that $O_i = OK$ perfectly correlates with $H_i = h$. The second row encodes the fact that a NOK observation could be caused by either c_1 or c_2. The zeros in the array encode impossible combinations. If need be, false positives and negatives could be incorporated into the factor. Similarly, $f_1(i) = f_1$ shown in Eq. (3.3), is the factor connecting any two variables H_i, H_{i+1} consecutive along the 1D path.

$$f_1 = \begin{matrix} \\ H_i = h \\ H_i = c_1 \\ H_i = c_2 \end{matrix} \begin{matrix} \begin{matrix} H_{i+1} = h & H_{i+1} = c_1 & H_{i+1} = c_2 \end{matrix} \\ \begin{bmatrix} 1 & \epsilon & \epsilon \\ \epsilon & 2 & \epsilon \\ \epsilon & \epsilon & 1 \end{bmatrix} \end{matrix} \tag{3.3}$$

The rows of f_1 correspond to the different states of H_i and the columns correspond to the different states of H_{i+1}. As in the case of f_O, $f_1[j,k]$ is the weight given to the combination $H_i = c_j$ and $H_{i+1} = c_k$, for $j,k = 0,1,2$. Let us assume root-cause c_1 displays larger correlation along the 1D path than cause c_2. Factor f_1 enforces this by making $f_1[1,1]$ larger than the other entries. For simplicity, we have given a small weight of ϵ to all non-diagonal entries of f_1. Depending on the application, these entries can be given distinct values or be functions parametrized by, e.g., time, or the factor's location on the graph.

Let $\mathsf{O} = \{O_i\}, \mathsf{H} = \{H_i\}$ be the observed and hidden variables, and $\mathsf{V} = \mathsf{O} \cup \mathsf{H}$ be their union. Let $\mathsf{E} \subset \mathsf{V} \times \mathsf{V} \times \mathbb{N}$ be the set of (multi-)edges between variables where $\times$ denotes the Carthesian product. Let $\mathsf{F} = \{f_j(\vec{p})\}$ be the set of (possibly) parametrized factors, $\vec{p}$ the factor parameters, and $\Lambda : \mathsf{E} \to \mathsf{F}$ be a map assigning a factor to each element in E. Then, the probabilistic graphical model $\mathcal{M} = (\mathsf{V}, \mathsf{E}, \mathsf{F}, \vec{p}, \Lambda)$ defines a Conditional Random Field.

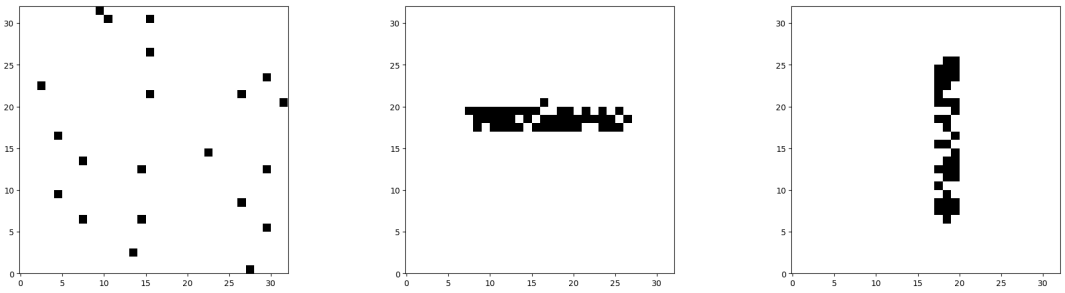
A CRF encodes the conditional probability distribution $p(\mathsf{H}|\mathsf{O})$, meaning that it encodes the joint posterior probability of the hidden variables H conditioned on the observed variables O.

In plain English, this captures what is likely to be going on in the system given the observations. We then obtain the marginal posterior probability $p(H_i|\mathbf{O})$ for every hidden variable H_i using loopy belief propagation [14] as our inference algorithms, although others exist, see [19]. This marginal posterior probability conditioned on the observations becomes our classification, i.e. our diagnosis for the root cause of the observation.

The strength of the CRF, and more generically of PGMs, as a framework for modelling defectivity patterns resides in its ability to contain arbitrary graph topologies, and to perform inference, i.e., reasoning, across all representations forming the graph. This empowers the modeller to adapt the model to the system at hand, including domain specific knowledge that would be hard to account for using more data-driven forms of image segmentation. Furthermore, we remark that our methodology is not limited to 1 or 2-dimensional lattices embedded in a graph. Indeed, one could conceive, e.g., of fan-like representations of electrical interconnections on solar panels, or completely irregular structures like the overlapping high-voltage grid and communication networks forming a modern electricity grid.

3.2.2 Multi-Resolution Spatial and Temporal Analysis

The factors in Figure 3.1 are all binary in the sense that they connect two variables. It is also possible to use unary (affecting just one variable) or higher order factors. Unary factors are useful because they act as weights on variables. In Bayesian terms, we can think of them as priors. These can come from historical data or from previous computations on the same data. In this work we will use unary factors to extend the models in two directions: multi-resolution and temporal analysis. We will first motivate the need for them and then explain how to construct the corresponding MRF.



a) Singlets pattern b) Horizontal pattern. c) Vertical pattern

Figure 3.2: Examples of all three considered features from simulated data of a system with a 2D representation.

Figure 3.2 depicts example measurements of a system with a 2D row-column representation with clear singlet, horizontal and vertical patterns. The horizontal and vertical patterns are examples of patterns that would be missed by the graph architecture in Figure 3.1-d. Looking at Figure 3.2-c we can see that the vertical pattern is broken into sections where NOK observations are diagonally connected with other NOK observations. However, the MRF only has vertical and horizontal connections along the 2D representation. The conclusion is that each one of these sections would be diagnosed separately, and given their shape, many of them would be classified as something other than part of a vertical pattern.

Overcoming some of these issues could be achieved by adding connections between non-contiguous observations, or by adding diagonal edges. These solutions add considerable computational complexity to the model while not fully solving the problem of identifying global patterns that are not contiguous. In this work we have opted for modelling the sys-

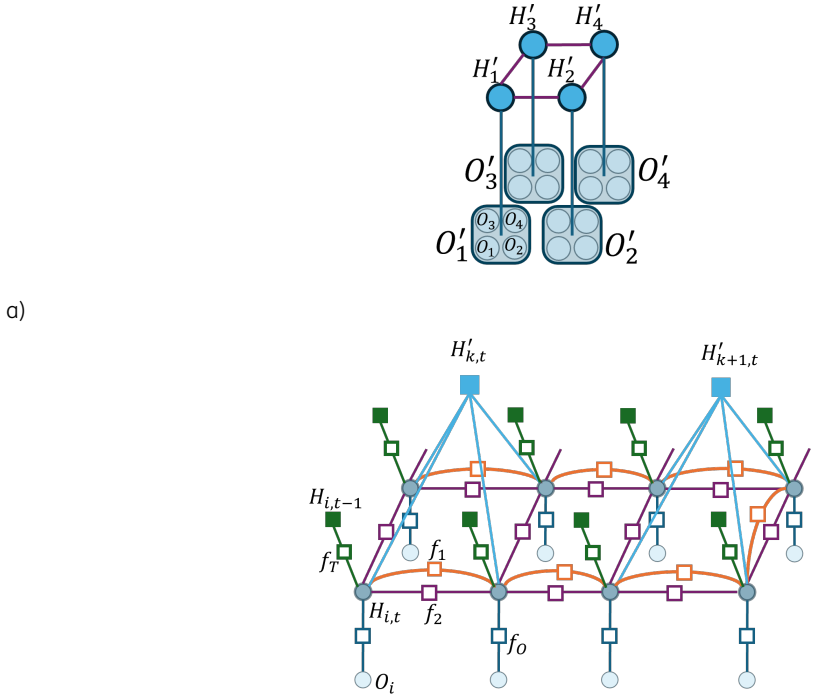


Figure 3.3: Construction of a low-resolution model for the system in Figure 3.1. a) Observations are pooled into blocks of size 2×2 and the new hidden states H'_k are connected. b) Depicts the full model with multi-resolution and temporal information. The inference results of the low-resolution model inserted as unary factors $H'_{k,t}$ (solid blue) and therefore not connected to each other, and the posteriors of the previous time-step inserted as unary factors $H_{i,t-1}$ (solid green), connected to the current variables $H_{i,t}$ via another factor accounting for time evolution of the states.

tem at a lower resolution scale. At low resolutions, patterns that are too discontinuous for the high-resolution model with only nearest neighbour interactions are smoothed over and become once again continuous, and thus, visible to the model. While multi-resolution analysis is not limited to 2D lattices, or indeed, lattices of any dimension [20], [21], in this work we limit ourselves to the 2-dimensional case since that is the main representation in the CPP use-case described in Section 4.

The process starts with grouping and fusing the 2D observations. In this work we group them into 2-dimensional blocks of $l \times l$. We use $MaxPool(O_i, \dots, O_{i+l^2-1})$ to determine the value of the effective observation O'_k representing the block, and assign also an effective hidden variable to each block H'_k. Then we create a model by connecting the pooled observations to the new hidden root causes for the blocks and connect these to each other following the 2D grid pattern inherited from the base model. This is shown in Figure 3.3-a. Although some features are shared, low- and high-resolution models might have different topologies and might have different factors. Upon usage, we use the low-resolution model to infer the marginal distributions of the blocks' hidden variables and subsequently we insert them as priors on the full-resolution model (solid light-blue squares), see Figure 3.3-b, [16].

The introduction of multi-resolution models, however, creates its own specific challenges. The process of reducing the resolution of the data cannot usually be done simultaneously and consistently across different representations. For example, in the toy model of Figure 3.1, it is not possible to pool the variables into groups of four that are contiguous in both the 2D and 1D representations, see the two columns on the left of that figure.

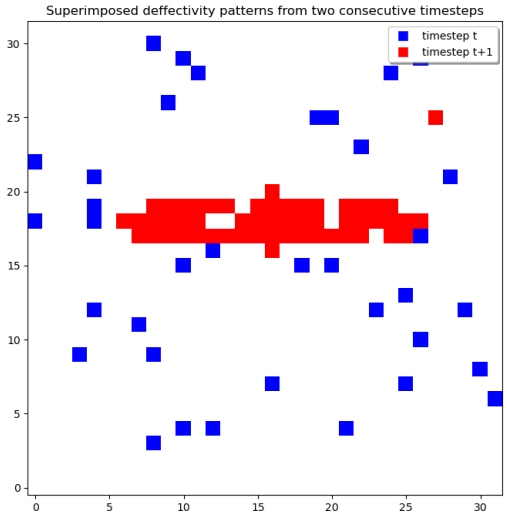


Figure 3.4: Measurements of defectivity in two consecutive timesteps.

The output of a lower-resolution 2D model thus cannot be interpreted as a classification of observations into proper root-causes. Rather, we refer to these classes as *features*. In this work we classify 2D observations into three different kinds of distinct 2D features: *singlets*, *vertical* patterns and *horizontal* patterns. Models that classify observations into these classes are referred to as *feature models*, in contrast to root-cause models.

The second use of unary factors or priors in this work is in modelling the evolution of the hidden states over time, be they root-causes or features. Let us motivate the need for this.

Consider a toy model consisting of 32×32 observations arrayed in a 2D grid. In Figure 3.4 we see the NOK measurements of appearing in two consecutive timesteps t and $t+1$, represented in blue and red respectively. In this example, NOK measurements remain once appeared, leading to growing patterns over time. At time t we observe a pattern that should be classified as singlets. At time $t+1$ an horizontal pattern is added which happens to be contiguous to some measurements that were previously NOK. Without taking time into account, those measurements would now be reclassified as belonging to a horizontal pattern, even though we know that they were previously classified as singlets and predate the appearance of that horizontal pattern, and therefore should not be classified as such.

The solution is to carry the outcome of inference from one timestep to the next. We assume the data consists of a time series of 2D observations. Our models so far have been geared towards representing and reasoning on one timestep of the series of observations. To allow the inferred hidden states in the previous timestep $H_{i,t}$ to affect the current timestep $H_{i,t+1}$, we insert them as unary factors containing the posterior distribution of $H_{i,t}$ computed in the previous iteration (solid green squares in Figure 3.3-b) and connect them to the variables $H_{i,t+1}$ via a factor f_T encoding the time evolution of the hidden states (empty green squares). This carrying of information forwards in time is a form of Bayesian filtering [22].

Adding time to our models (in the form of posteriors for previous time steps as priors for the current) allows us to keep our classifications consistent across time. This is important because, even though the features may seem to change from one moment to another, the root causes do are consistent in time. Keeping the inferred features also consistent allows us to connect features to root causes.

3.2.3 Multistep reasoning

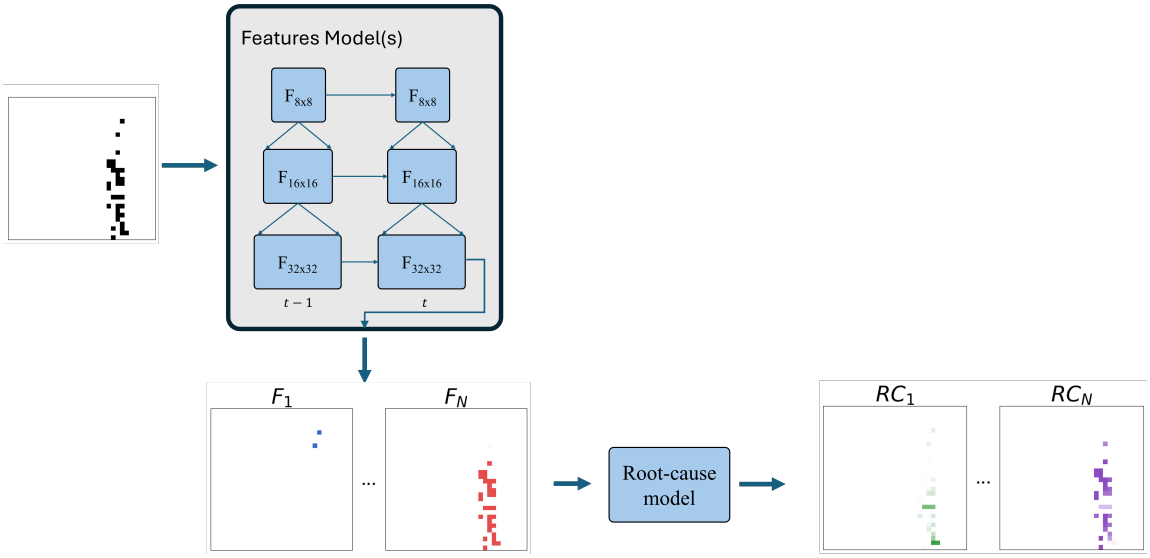


Figure 3.5: Simplified view of the multistep reasoning workflow, with feature models followed up by root-cause models

In the previous section we introduced multi-resolution and temporal analysis to create a *features* model which outputs classes that are "global" in the 2D representation and consistent in time. These features, or classes, do not map one-to-one to our root causes. In order to further refine our classification, we use the output of the features model as input for another *root-cause* model. This model differs from the features model in that:

1. it includes all the other representations, and is not limited to a 2D grid.
2. It uses the output feature classification of the features model as the observed states instead of the OK, NOK measurements used as evidence on the features model.
3. The states of its hidden variables correspond to root-causes, rather than features.

We depict this workflow graphically in figure 3.5. We will revisit this multistep flow in depth in Chapter 4, where we apply it to a Canon Production Printing example.

3.2.4 Key aspects

The key aspects of the methodology are its ability to allow for reasoning across different representations, resolutions and timesteps. The arbitrary topology of graphical models allows the modeller to incorporate bespoke geometric and process knowledge in the location of the factors. At the same time, expert knowledge about the behavior of root causes can be incorporated in the parameters determining the factors. These parameters can be either chosen from first principles or learned. Probabilistic graphical models are also generative models, and can therefore be used both for inference and simulation, assuming the model topology and parameters accurately capture reality.

3.2.5 Assumptions and limitations

In the development of the methodology we have made a few assumptions and identified a few limitations. As per the assumptions:

1. Discrete variables: We have assumed that all variables are discrete. This includes measurements (OK, NOK) and hidden states, either reflecting features or root causes. Discretization is natural in root cause analysis, but does mean that performance issues due to degradation, as well as prediction and prognosis, are hard to accommodate to the formalism.
2. Accurate measurements: We have assumed that measurements are accurate (no false positive/negative rates). This could easily be changed by altering the measurement factors connecting hidden and observed variables. In the systems we have studied this has not been an issue. This seems a reasonable assumption given that precision metrology is an enabler of precision manufacturing.
3. Known root causes: We have assumed contributors (root causes) are known and independent. The methodology cannot identify patterns arising from root-causes not accounted for in the modelling process. In other words, the methodology does not know how to say "I don't know". If the system in question follows the Pareto principle, meaning that a handful of the root causes will typically explain the majority of the abnormal cases, one would expect the impact of this assumption to be limited.
4. Known topology and factors: We have assumed complete knowledge of the different representations in which to view the different failure mechanisms. We have also opted for manually specifying the factors. The first assumption is quite natural, given that these are engineered systems. The second is a limitation, as discussed below.

We have also encountered a number of limitations:

1. Specification: Manually specifying good parameters for the factors requires skill at translating root-cause behaviour into probabilistic weights.
2. Scalability: The number of parameters to tune grows linearly with the number of representations and quadratically with the number of root causes. If the factors are not constant but vary over the layout, the number of parameters could grow even faster, although that would depend on the specific way factors depend on the layout.
3. Model Complexity: Creating models that are capable of reasoning across time, multiple resolutions and multiple representations all at once is challenging. Our solution has been to split these stages of reasoning into multiple models that form a chain of reasoning. This limits the potential of the reasoning, as not all information can be used all at once.
4. Computational limitations: The computational complexity of loopy belief propagation, the algorithm chosen for inference, is $O(I \cdot E \cdot D^2)$, where I is the total number of iterations, E is the number of edges, in this case $E = O(N)$, and $D = O(1)$ is the number of distinct root causes, the algorithm is very space efficient, requiring only $O(E \cdot D)$ bits of memory. Although the computation time varies per case, it stays around a second on a standard laptop for models with 10^3 hidden variables out of ca. 5000, acceptable for the scope of this study. Loopy belief propagation is not guaranteed to converge to the true posteriors [19]. Scaling up to thousands or tens of thousands might require changes to the inference algorithm.

4 Industrial application: print quality

This chapter is adapted from [19].

4.1 Description of the case

In this section, we discuss the application of the methodology to an industrial use case: the diagnosis of print quality artifacts in an industrial inkjet printer manufactured by Canon Production Printing.

Print quality is a key performance criterion in industrial inkjet printing, as it directly affects the overall quality of the product. Even minor mechanical or operational deviations in the printing system can lead to visible artifacts on the print, such as streaks or color banding. In industrial printers, a **printhead** is the component that deposits ink onto the substrate. Each printhead is dedicated to a single color — **yellow (Y)**, **magenta (M)**, **cyan (C)**, and **black (K)**. These printheads are arranged in a **printhead array** and aligned along the direction in which the paper moves through the printer. Each printhead contains thousands of **nozzles**, each only a few micrometers in diameter, which collectively jet billions of ink droplets per second to ensure continuous, high-quality printing.

To monitor nozzle performance, the printer periodically prints, scans, and analyses a **test page**. This process provides an indirect measurement of each nozzle's printing accuracy, indicating which nozzles are jetting correctly and which ones do not. The resulting measurements are logged in a database and can be used for downstream analyses. These measurements are also used by the printer to assess nozzle health and determine when maintenance actions are required. A high-level schematic of the printing and scanning process is shown in Figure 4.1a.

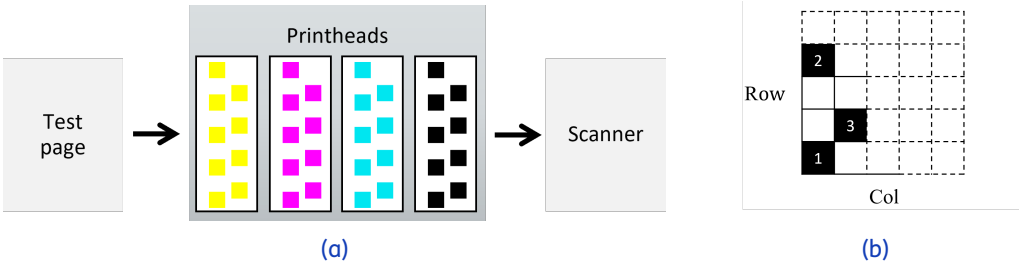


Figure 4.1: (a) High-level schematic of the inkjet printing process, where only test pages are scanned; (b) example positions of three nozzles on the printhead, indicated in black. The number assigned to each nozzle (1, 2, 3) corresponds to the jetting order.

4.1.1 Nozzle layout and jetting order

Nozzles on the printhead are arranged in a two-dimensional grid, where each nozzle can be identified by its row and column position, as shown in Figure 4.1b. We refer to this configuration as the **printhead layout**. To ensure that the nozzles print along a straight line as the

medium moves beneath the printhead array, each nozzle fires (jets ink) at a slightly different time. The timing differences are on the order of microseconds. We refer to the sequence in which the nozzles fire as the **jetting order**, which follows the nozzle numbering shown in Figure 4.1b. Notice that nozzles in close spatial proximity on the printhead, such as nozzles 1 and 3, are not consecutive in the jetting order.

4.2 Specific challenges

The main challenge in diagnosing failing nozzles is inferring the (hidden) root causes from the measured states of the nozzles. As described in the previous section, nozzle performance is evaluated by printing and scanning dedicated test sheets. From the scans, the jetting angle for each nozzle is computed. Each nozzle's health is then assessed by comparing its measured jetting angle against a set threshold. If the angle remains within this threshold, the nozzle is reported as *OK*; otherwise, it is reported as *NOK*. For example, a nozzle that is permanently broken and a nozzle that is clogged, will both be reported as *NOK*. Since the true underlying state of each nozzle cannot be observed directly, diagnosis must rely on temporal dynamics and spatial correlations to infer the most likely root cause.

Furthermore, knowledge on the root causes is mostly tacit. Unlike system-down events, which trigger service calls, print quality issues are typically intermittent, gradual, or resolved after routine maintenance actions. Hence, the underlying failure mechanisms are seldom observed directly by service engineers and even more rarely annotated. When interventions do occur, the affected printhead is often replaced in its entirety, meaning that the true cause remains unknown. As a result, knowledge about failing nozzle patterns resides primarily with a small group of domain experts and is not systematically captured. Moreover, interpretations of these patterns are not always consistent across experts. This leads to a lack of ground truth. In some instances, controlled experiments can provide ground truth, but these may be operationally costly and not always feasible for all root causes.

4.3 Scoping

The space of nozzle failure patterns is large and highly unbalanced. While a small number of patterns occur frequently in the field, many others appear only sporadically, resulting in a long tail of rarely observed cases. Given this distribution, it is infeasible to build a diagnostic model that covers the full range of potential root causes.

According to domain experts from Canon Production Printing, the majority of recurring nozzle failure patterns occur at the maintenance side of the printhead array. This region comprises the nozzles located closest to the edge of the paper. Failures in this region often manifest as vertical patterns of non-jetting nozzles. For this reason, the application of our methodology focuses on diagnosing the root causes, four in total, of vertical non-jetting patterns occurring at the maintenance side of the printhead array. These root causes were selected based on their high recurrence in field data and their practical relevance, as they correspond to failure mechanisms that can be addressed through routine maintenance actions.

For confidentiality reasons, the physical and operational details of the root causes cannot be disclosed and are therefore denoted as RC_1 – RC_4. These labels correspond to distinct failure mechanisms recognized by domain experts.

4.4 Methodology application and validation

4.4.1 Data collection and preparation

The validation dataset used consisted of test sheets showing vertical patterns of non-jetting nozzles. The selected cases were chosen to reflect patterns associated with the four root causes in scope. Not all cases could be assigned to one of these causes; this is addressed explicitly in Section 4.4.3.1.

4.4.1.1 Data collection

Nozzle performance data was retrieved from an internal database that logs the binary nozzle states for each printed test sheet from multiple printers operating in the field. Data extraction was semi-automated and limited to a fixed operational period (1 May 2024 to 31 July 2024).

The root causes considered in this study manifest as vertical patterns involving between 10 and 100 non-jetting nozzles. According to domain experts, this range corresponds to the typical pattern size observed for the defined set of root causes. Based on this domain knowledge, records were filtered to retain only test sheets within this range and for which nozzle data was available for all printheads in the printhead array.

As the resulting records were unevenly distributed across printers, stratified sampling by printer identifier was applied to mitigate skewed data and prevent over-representation of patterns from a small number of printers. Remaining patterns were then manually reviewed to obtain a representative set of 100 vertical patterns.

Maintenance context. According to domain experts, the temporal behavior of nozzle patterns across maintenance events is critical for distinguishing between different root causes. Some failure mechanisms persist across maintenance actions whereas others are mitigated or resolved by them.

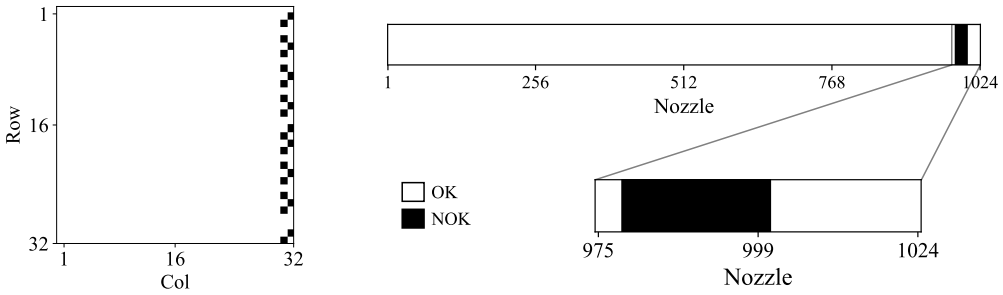
For this reason, and to support both data labeling and model validation, additional nozzle data was retrieved for each test sheet within a temporal window around maintenance events. This temporal context is explicitly used by the model to reason across successive test sheets. The window includes the ten test sheets preceding the last maintenance action before the observed pattern and the ten test sheets following the first maintenance action after it.

4.4.1.2 Data preparation

Because this application focuses on defectivity patterns located at the maintenance side of the printhead array, only the last 32 columns of nozzles closest to the paper edge are considered. Combined with the 32 nozzle rows on each printhead, this results in a binary 32×32 image per printhead:

$$O^{p}_{32\times32}(t_n) \in \{0,1\}^{32\times32},$$

where 1 denotes a non-jetting (*NOK*) nozzle and 0 a jetting (*OK*) nozzle. One such 32×32 image is obtained per printhead, $p \in \{Y, M, C, K\}$ — corresponding to the yellow, magenta, cyan, and black printheads — and these four images serve as input to the diagnostic workflow. An example of one image is shown in Figure 4.2a.



(a) (b)

Figure 4.2: Example of binary nozzle data for a single printhead as shown in (a) the printhead layout (2D) representation and (b) the jetting order (1D) representation, obtained by flattening the 2D representation according to the jetting order (see Figure 4.1b).

4.4.1.3 Labeling

Two micro print processing experts, with expertise in analyzing nozzle defects and nozzle-level processes, annotated the 100 patterns at nozzle level using an internally developed prototype tool. Each nozzle was labeled as one of the predefined root causes, assigned as *Other* if the root cause was determined to be something else, or marked as *Unknown* if the expert could not confidently identify the cause.

4.4.2 Implementations of the methodology

The complete workflow for diagnosing print quality issues at multiple spatial resolutions and across time is shown in Figure 4.3. The workflow consists of two stages: (i) a cascade of *feature models* that extract global 2D patterns as described in Section 3.2.3, and (ii) a two-stage *root cause classification*, performed first at low resolution and finally at nozzle level.

4.4.2.1 Feature models

Each image $O^p_{32\times32}(t_n)$ first passes through a sequence of three printhead-independent feature models,

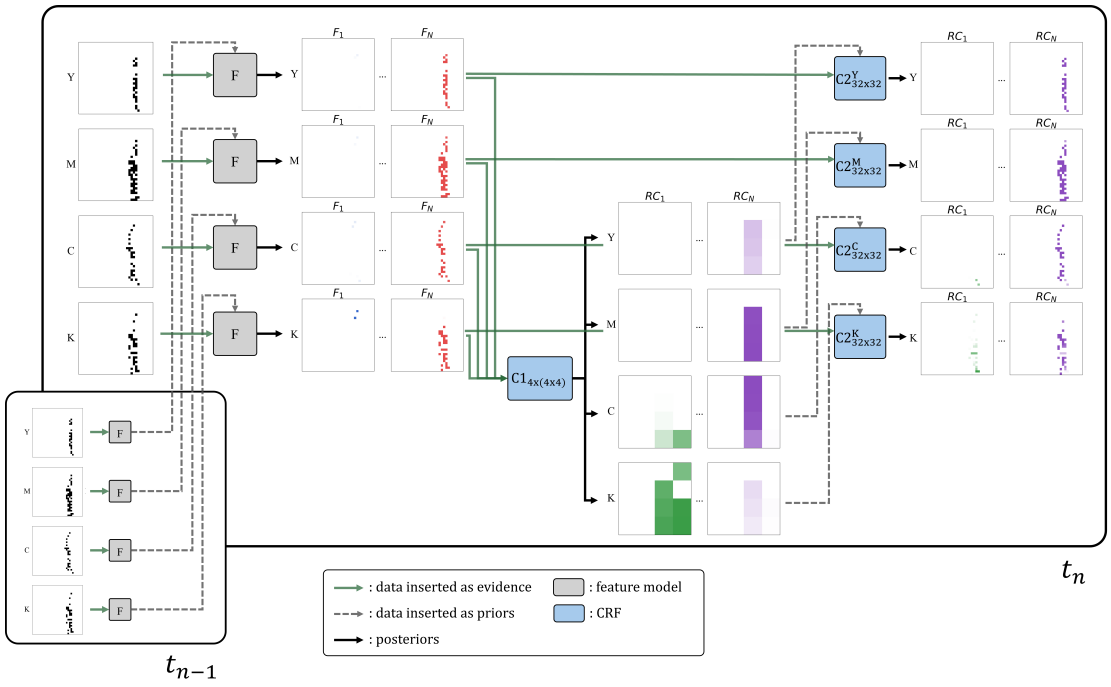
$$F_{8\times8}, \qquad F_{16\times16}, \qquad F_{32\times32},$$

each implemented as a CRF with hidden states $f \in \{\mathrm{N}, \mathrm{S}, \mathrm{H}, \mathrm{V}\}$ corresponding to the *Normal*, *Singlet*, *Horizontal*, and *Vertical* classes. The resolutions $\{8, 16, 32\}$ were selected empirically. A resolution of 8×8 was considered coarse enough to capture horizontal and vertical patterns of failing nozzles.

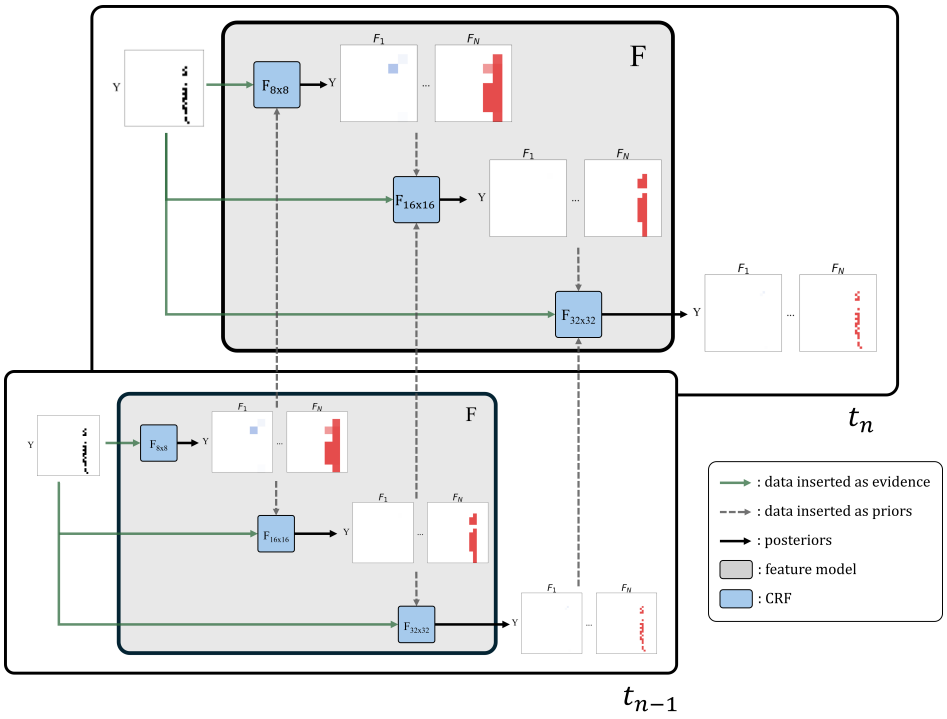
Unlike the generic CRF as shown in Figure 3.1, the feature models only use the 2D grid representation and do not include the f_1 (1D path) factors. The goal of the feature models is to extract global 2D patterns that are later used for root cause classification. A visual representation of the feature models workflow is shown in Figure 4.3b.

Model inputs. The model $F_{l\times l}$ at resolution $l \in \{8, 16, 32\}$ receives the following inputs:

1. The downsampled binary image: $O^p_{l\times l}(t_n) = \mathrm{MaxPool}_{32\to l}(O^p_{32\times32}(t_n))$.
2. The posteriors from the previous time step, inserted as priors: $F^p_{l\times l}(t_{n-1})$.
3. For $l > 8$, the posteriors of the preceding lower-resolution model, inserted as multi-resolution priors: $F^p_{(l/2)\times(l/2)}(t_n)$.



(a)



(b)

Figure 4.3: Overview of the workflow of the probabilistic features and root cause models for the print quality application. (a) Overall diagnostic workflow, with the feature model cascade indicated as gray blocks labeled F. (b) Internal structure of the feature model cascade, showing the multi-resolution and temporal inference.

Model outputs. Each $F_{l\times l}$ outputs a 3D tensor $F^p_{l\times l}(t_n) \in [0,1]^{l\times l\times 4}$, where the last dimension indexes the feature set $\{\mathrm{N},\mathrm{S},\mathrm{H},\mathrm{V}\}$. The output of the final feature model $F_{32\times 32}$ is used as evidence for the subsequent root cause models.

4.4.2.2 Low-resolution root cause model

The output of the final feature model $F_{32\times 32}$ serves as input to a single CRF,

$$C1_{4\times(4\times 4)}.$$

Structurally, the low-resolution model is equivalent to four 4×4 CRFs, one per printhead. These are then connected into a single model by linking the hidden variables on the bottom row of one printhead to the top row of the next, in the order $Y \to M \to C \to K$. This linking enables the model to reason on patterns that extend across printheads and enforces consistent classification of such patterns as a single root cause. Moreover, whether a pattern extends across multiple printheads also provides discriminative information for identifying certain root causes.

The low-resolution root cause model excludes the f_1 factors and does not incorporate temporal priors. Multi-resolution priors are not applicable at this stage.

Model inputs. For each printhead p, the feature posteriors $F^p_{32\times 32}(t_n)$ are divided into sixteen non-overlapping 8×8 regions. Within each region, feature posteriors are pooled using a threshold-based operation, yielding a single probability vector per region. This results in a 4×4 grid of observed variables $\widetilde{F}^p_{4\times 4}(t_n)$.

Each region is associated with a hidden random variable taking values in the set of possible root causes $c \in \{\mathrm{N},\mathrm{RC}_1,\mathrm{RC}_2,\mathrm{RC}_3,\mathrm{RC}_4\}$, where N denotes the *Normal* state.

Model outputs. The model outputs root cause posteriors for each of the sixteen regions per printhead: $C1^p_{4\times 4}(t_n) \in [0,1]^{4\times 4\times 5}$. These posteriors serve as priors for the high-resolution root cause models.

4.4.2.3 High-resolution root cause model

In the final stage of the workflow, four printhead-dependent root cause models perform root cause classification at nozzle level:

$$C2^Y_{32\times 32},\quad C2^M_{32\times 32},\quad C2^C_{32\times 32},\quad C2^K_{32\times 32}.$$

These models follow the multi-representation structure as shown in Figure 3.1, but exclude the temporal priors. Here, the 2D representation corresponds to the printhead layout and the 1D representation corresponds to the jetting order, see Figure 4.2.

Model inputs. Each high-resolution model $C2^p_{32\times 32}$ receives the following inputs:

1. The posteriors from the $F_{32\times 32}$ feature model: $F^p_{32\times 32}(t_n)$.
2. The posteriors from the low-resolution root cause model: $C1^p_{4\times 4}(t_n)$.

Although the root cause models do not directly observe the original nozzle states O, *OK* nozzles are not mapped to a root cause. In the feature models, only *NOK* observations produce non-zero posteriors for the *Singlet*, *Horizontal*, and *Vertical* states, while jetting nozzles are mapped to the *Normal* state. Similarly, in the root cause models, this *Normal* state is preserved and not mapped to any of the four root causes.

Model outputs. Each CRF outputs nozzle level posteriors over all root causes, representing the inferred state of each nozzle at time t_n: $C2^p_{32\times32}(t_n) \in [0,1]^{32\times32\times5}$.

In summary, the feature models perform multi-resolution and temporal reasoning within the 2D representation, while the low-resolution root cause model performs cross-printhead reasoning without temporal or multi-resolution priors. The high-resolution root cause models then refine the diagnosis at nozzle level using the outputs of both.

All factors in the CRFs used in this application are specified from first principles using domain expert knowledge and were refined after reviewing model outputs with print quality experts.

4.4.3 Evaluation

The diagnostic performance of the workflow was evaluated by comparing inferred root causes from the high-resolution models against expert annotations. Although inference was performed at nozzle level, evaluation was conducted at the level of an entire test sheet.

Model performance was quantified using overall **accuracy** and the **macro-averaged F1 score**. Accuracy reflects the proportion of correctly classified patterns, while the macro-averaged F1 score evaluates the performance of a model by treating all classes as equally important, regardless of how many samples each class contains. Both metrics take values between 0% and 100%, where higher values indicate better performance; a value of 100% corresponds to perfect classification.

4.4.3.1 Ground truth

As the ground truth on the underlying failure mechanisms is unavailable, expert judgment was used as a proxy. Two print quality experts jointly reviewed all cases, providing nozzle-level annotations, and reached consensus on the assigned root cause for each case. The nozzle-level annotations were aggregated by counting the number of nozzles assigned to each root cause, and the dominant [3] root cause was taken as the case label. The aggregated labels were then reviewed with the experts and confirmed to reflect the intended dominant root cause. Only patterns labeled as $\{\mathrm{RC}_1, \mathrm{RC}_2, \mathrm{RC}_3, \mathrm{RC}_4\}$ were considered; patterns labeled as *Other* or *Unknown* were excluded. This resulted in a total of 81 patterns.

4.4.3.2 Model prediction

For each pattern, nozzle-level posterior probabilities were aggregated by summing the probabilities per root cause across all nozzles. The predicted root cause was defined as the class with the highest aggregated posterior and compared against the expert-assigned label.

4.4.4 Results

On the validation set of 81 patterns, the workflow achieved an overall accuracy of **86%** (70 correct classifications). The macro-averaged F1 score was **86%**. The confusion matrix summarizing the classification results is shown in Figure 4.4. The evaluation set contained 17, 21, 21, and 22 patterns for RC_1 through RC_4, respectively.

Figure 4.5 and Figure 4.6 show the inferred posterior distributions for two patterns from different printheads and test sheets. Both patterns exhibit a similar vertical band of non-jetting

[3] This aggregation step allowed us to assign a single label to cases containing multiple root-causes.

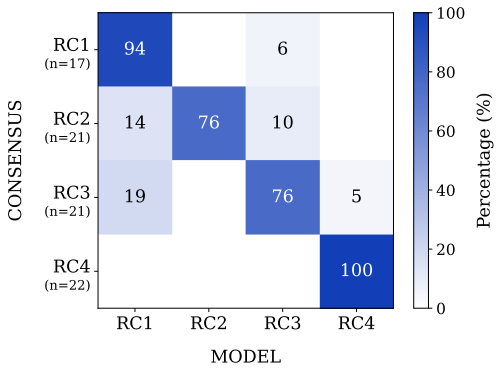


Figure 4.4: Row-normalized confusion matrix comparing the predicted classes of the CRF-workflow with the corresponding labels on the validation set. To illustrate, of the patterns labeled as RC_2, three (14%) are misclassified as RC_1 and two (10%) as RC_3. Overall accuracy on the validation set is **86%** (70 correct classifications).

nozzles at the edge of the printhead, as captured by the feature models. When downsampling the output of the feature models, the non-jetting nozzles at the edge of the printhead merge into a continuous vertical band, enabling the low resolution root cause model to capture broad, global patterns. RC_1, RC_2, and RC_3 are associated with root causes that appear as vertical structures in the spatial domain. Whereas RC_4 is associated with isolated underperforming nozzles, which can appear as clusters of any shape when the input is downsampled.

The definitive diagnosis is performed at full resolution, where the model reasons jointly over both the jetting order and the printhead representations. In Figure 4.6, the inferred posteriors align with the continuous sequence of non-jetting nozzles along the jetting order, leading to a diagnosis consistent with a jetting-order-related root cause (RC_2). In Figure 4.5, the model assigns higher posteriors to a root cause that manifests primarily in the spatial domain (RC_1). In addition, a small number of isolated non-jetting nozzles are present in both patterns. These do not form part of the dominant vertical structure and are therefore not clearly identified in the output of the low-resolution root cause model. When considered by the high-resolution model, however, these isolated observations are correctly attributed to RC_4, a root cause related to single failing nozzles.

By incorporating and reasoning over both the printhead layout and the jetting order representations, the CRFs are able to distinguish between different root causes that give rise to superficially similar vertical defectivity patterns. The resulting diagnoses were reviewed with domain experts at Canon Production Printing and were found to be consistent with their understanding of the underlying failure mechanisms.

4.5 Lessons learnt

The application of the proposed methodology to this industrial use case yielded several insights that are relevant beyond the specific context of print quality. These lessons provide guidance for developing (probabilistic) models for performance diagnostics in industrial settings:

- **Scoping of diagnostic models.** In operational environments, the space of possible defectivity patterns and root causes is typically large, highly unbalanced, and only partially understood. Attempting to model this space exhaustively is therefore impractical.

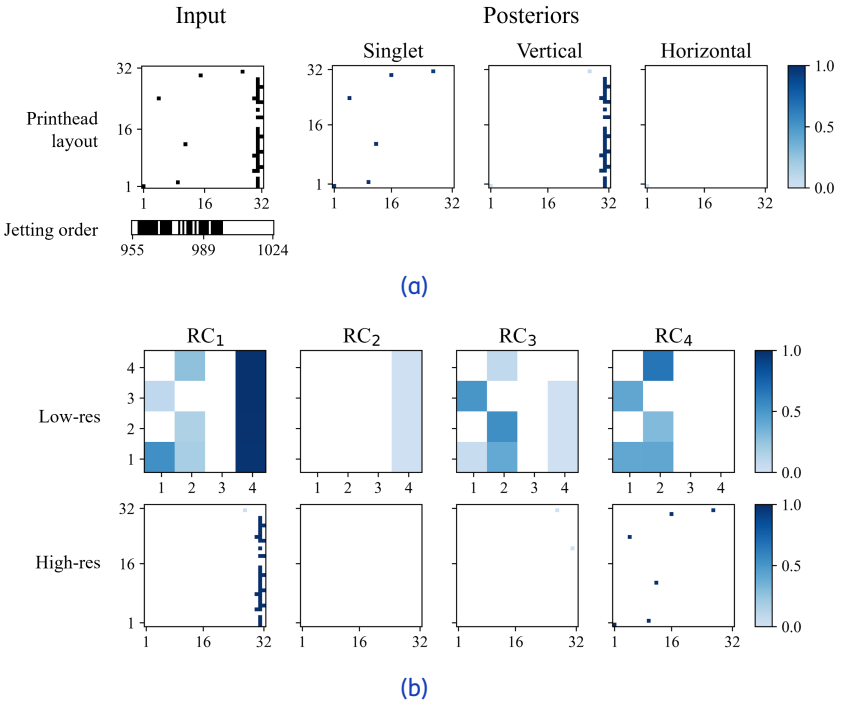


Figure 4.5: Inference results for a defectivity pattern on a single printhead. (a) Binary nozzle states used as input and posteriors of feature model. The nozzle states used as input to the workflow are visualized in both the printhead layout (2D) and jetting order (1D) representations. For the jetting order representation, only the last 70 nozzles are shown for clarity. (b) Root cause model posteriors. The columns labeled RC_1–RC_4 show the inferred posterior probabilities for each root cause for both the low-resolution and high-resolution root cause model.

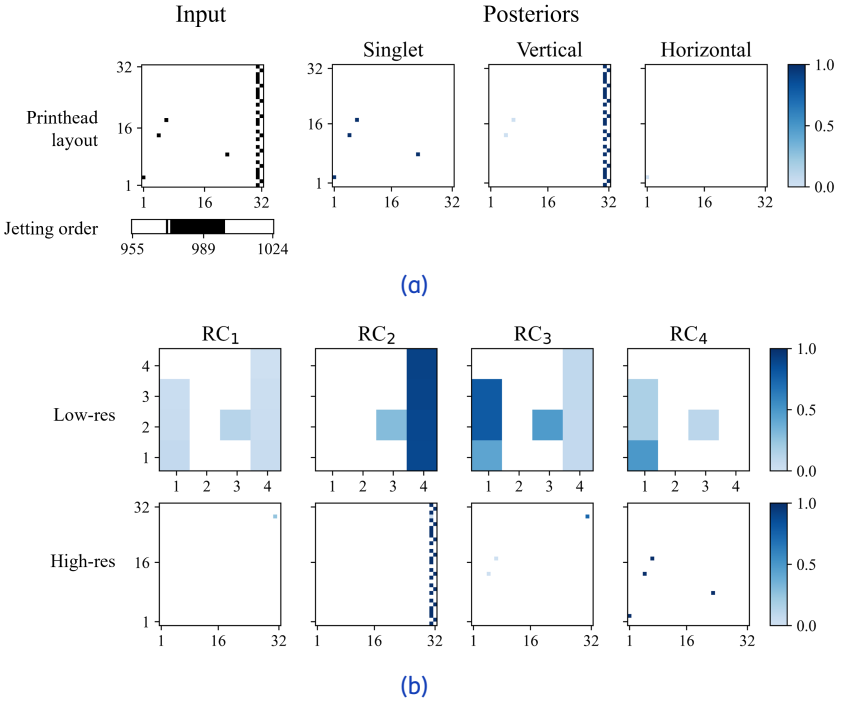


Figure 4.6: Inference results for a second vertical defectivity pattern. The layout and visualization are identical to Figure 4.5, showing model input and feature posteriors (a) and root cause model posteriors for both the low-resolution and high-resolution root cause model (b).

In this application, restricting the scope to a limited number of recurring and actionable root causes, and to a specific region of the printhead array, proved to be an important precondition for successful modeling. This illustrates that, in industrial diagnostics, completeness often needs to be traded for robustness and interpretability.

- **Use of multiple data representations.** While the two-dimensional printhead layout provides a natural representation of nozzle data, it was insufficient to distinguish between root causes producing similar spatial patterns. Incorporating the jetting order as an additional one-dimensional representation was needed to disambiguate different failure mechanisms, as shown in Figure 4.5 and Figure 4.6. Expert feedback further indicated that additional representations, such as the appearance of the pattern on the printed test sheet, could provide complementary information.
- **Availability of labeled data.** As is common in performance diagnostics, labeled data on root causes were scarce, since failures are rarely observed directly and expert knowledge is largely tacit. A practical lesson is that building and maintaining a library of labeled defectivity archetypes is highly beneficial. Such a library consolidates expert knowledge, supports consistent labeling, provides a reference for model validation, and facilitates the training of new service engineers. Importantly, this should be treated as a long-term investment rather than a one-off effort.
- **Validation and expert involvement.** Quantitative performance metrics are necessary but insufficient to establish trust in an industrial setting. Reviewing inferred diagnoses with domain experts provided qualitative validation and informed subsequent refinements of the model, while also building confidence in the methodology among the experts.

5 Conclusions

This report has presented a scalable and robust methodology for the performance diagnostics of high-tech production systems, specifically addressing defectivity patterns that manifest on manufactured products. By leveraging Probabilistic Graphical Models (PGMs) in the form of Conditional Random Fields (CRFs), the proposed framework proved to successfully bridge the gap between data-driven and knowledge-based diagnostic approaches. The methodology is characterized by the following key features:

- Multi-representation reasoning: the use of arbitrary graph topologies in the PMG allows for the modeling of complex spatial dependencies that reflect the underlying physical layout and processes of the system.
- Multi-resolution analysis: by incorporating hierarchical levels of (modeling) granularity, the approach can identify both local and global patterns that might be not identifiable at a single resolution.
- Temporal reasoning: the integration of Bayesian filtering techniques enables the system to carry information forward in time, ensuring diagnostic consistency and the ability to distinguish between persistent and transient issues.
- Multistep reasoning: the methodology employs a structured approach that first classifies observations into intermediate features before performing a final root-cause classification.

Preliminary validation of the methodology demonstrated its effectiveness in correctly identifying the root cause in 70 out of 81 cases involving an industrial printer from CPP, as validated against a two-expert consensus. These results suggest the methodology can serve as an effective preliminary triage tool. By offering initial diagnostic suggestions, it can streamline the troubleshooting process and reduce the mean time to resolution of performance diagnostic cases.

5.1 Main findings

Below we list the main findings resulting from our investigations. These findings serve as answer to the research question detailed in Section 1:

- Knowledge about performance-related failures is not always available during the design phase of a new system or component, mostly because these failures cannot be known or predicted beforehand and only manifest during actual operation. This poses a challenge for a model-based approach, as it is difficult to acquire the knowledge required to specify and calibrate the models before the system is fully operational.
- Limited efforts are undertaken by industrial organizations to systematically label large amounts of performance related issues from systems in the field. This poses a challenge for a data-driven approach due to the lack of training datasets.

- Probabilistic graphical models (PGMs), similar to those used in hard-down diagnostics and involving solely discrete random variables [4], can be effectively applied also to performance diagnostics problems. These models allow for structured reasoning under uncertainty and can support the root cause analysis of defectivity patterns.
- Performance diagnostics of defectivity patterns requires reasoning across both time and multiple resolution scales. To be effective, the PGM must therefore incorporate temporal dynamics and multiscale dependencies, capturing how system behavior evolves over time and across (spatial) levels of granularity.
- On-product defectivity patterns are the result of root-causes acting on the various processes performed by high-tech production systems. Therefore, their diagnosis requires both local reasoning (examining a defect in the context of the specific process where it likely originated) and global reasoning (considering how interconnected processes might interact to cause a given defect pattern).
- For some performance diagnostic problems, discrete PGMs are insufficient. These cases require PGMs that include also continuous random variables. Developing and applying such models remains an area for further research, with probabilistic programming identified as a promising approach to address this next level of complexity [23, 24].

5.2 Future research

Based on the methodology and findings presented in the document, below we summarize key directions for future research:

- Extension to continuous random variables: while the current approach allows only of random variables with discrete states (e.g., OK, NOK), future research should allow incorporating continuous random variables. In industrial applications, this would allow the model to reason on varying ranges of performance degradation, such as the degrees of jet-angle deviation in nozzles of inkjet printers, or gradually shifting environmental factors like temperature and pollution. This extension therefore will expand the methodology's applicability to a wider range of diagnostic use cases, both within the high-tech industrial sector and across other industries.
- Confidence metrics: The current approach cannot account for root causes that it does not know, or return a measure of the confidence the user should have in the computed diagnosis. We are currently investigating the use of measures of computational hardness, e.g. number of iterations of belief propagation, together with the likelihood of the model given data, to characterize our confidence in the model's output. Another approach would be to use expert-labelled data to obtain a set of cases where the model is wrong, and use deep learning to learn an embedding of the cases where the model struggles. In this setting one would not even need to run the model to obtain an estimate of the hardness of the case.
- Factor learning: a current limitation of the methodology is the reliance on manual parameter specification, which requires significant domain expertise, multiple iteration for optimization of the factors, and eventually can be difficult to scale. Shifting from manual tuning to learning the factors directly from operational data would reduce the modeling burden on human experts and allow the system to adapt more dynamically to the many interacting aspects found in high-tech systems. Future research should investigate the computational feasibility of such factor learning, specifically focusing on methods that can operate effectively despite the scarcity of labeled data.

- Lightweight alternatives to PGMs: to increase computational performance and scale the modeling effort to more complex use cases, alternatives to traditional PGMs might be investigated; one promising alternative we identified in this context is graph signal processing [25]. In this approach, the graph structure (which could be inherited from a PGM) still encodes the essential interdependencies, but it eliminates the need for detailed knowledge for the specification of factors. This results in an approach with lower inference capabilities, but provides a significant reduction in modeling complexity and the computational resources required for inference.
- Ablation study: to evaluate the contribution of the key components of the methodology, such as multi-representation reasoning, multi-resolution analysis, etc., would be interesting to conduct an ablation study. For example by systematically removing individual features to assess their specific impact on the CRF model's inference capabilities.

5.3 Recommendations for industrial implementation

In our investigation we observed that while purely data-driven models are often constrained by the scarcity of labeled industrial datasets, PGMs provide a robust alternative, by integrating domain expertise directly into the diagnostic approach. Based on this observation we propose a twofold strategy for industrial implementation:

- Improvement of data labeling: to enable the use of advanced data-driven methodologies, organizations must bridge the "labeling gap". This should be executed via the deployment of specialized labeling dashboards and by implementing software interfaces designed to streamline the annotation process for site engineers. These efforts will support the development of a failure archetype library, establishing a centralized repository of failure modes. This library should be iteratively expanded using cross-fleet data to capture a diverse range of system behaviors and edge cases.
- Knowledge formalization: in scenarios where data-driven methods, even those trained on labeled sets, fail to achieve the required diagnostic accuracy, expert knowledge must be structurally harvested. This should happen by capturing the heuristics and causal reasoning used by domain experts during troubleshooting. This consolidated knowledge can then be used to define the structure and parameters of model based diagnostic approaches, such as the PGMs proposed in this study, to enhance diagnostic accuracy.

References

[1] P. van Kappen et al. *High-tech industry in 2040*. Den Haag: TNO, 2023. URL: https://publications.tno.nl/publication/34640945/W2sLVo/kappen-2023-hightechindustry.pdf (visited on 01/09/2026).

[2] L. Barbini, A. Piedrafita, and T. C. Nägele. *Vision and Outlook on High-Tech Equipment Diagnostics*. Eindhoven: TNO, 2024. URL: https://publications.tno.nl/publication/34643646/2mUzs9JC/TNO-2024-R12778.pdf (visited on 01/09/2026).

[3] T. C. Nägele and P. America. *Model-Based System Engineering for Diagnostics*. Eindhoven: TNO, 2025.

[4] T. C. Nägele et al. *SD2Act 2024: Guided Diagnosis of Functional Failures in Cyber-Physical Systems*. Eindhoven: TNO, 2025. URL: https://resolver.tno.nl/uuid:52f484ea-2638-4403-88a4-65305b245697 (visited on 12/01/2025).

[5] Breathing Color. *Bad Prints, Icc Profiles, Head Strikes, Inaccurate Color, Banding, Ink Artifacts*. https://www.breathingcolor.com/blogs/news/top-5-printmaking-tipstrickstechniques-june-11 [Accessed: 2025-11]. 2025.

[6] J. Ma et al. "Review of wafer surface defect detection methods". In: *Electronics* 12.8 (2023), p. 1787.

[7] TKI. *CareFree project website*. https://hollandhightech.nl/en/programmes-and-projects/projects/carefree-2025 [Accessed: 2025-11]. 2025.

[8] J. Bai et al. "A Comprehensive Survey on Machine Learning Driven Material Defect Detection". In: *ACM Computing Surveys* 57.11 (2025), pp. 1–36.

[9] Kaggle. *WM-811K wafer map dataset*. https://www.kaggle.com/datasets/qingyi/wm811k-wafer-map [Accessed: 2025-11]. 2025.

[10] J. Zhu, M. Jiang, and Z. Liu. "Fault detection and diagnosis in industrial processes with variational autoencoder: A comprehensive study". In: *Sensors* 22.1 (2021), p. 227.

[11] D. Mehta and N. Klarmann. "Autoencoder-based visual anomaly localization for manufacturing quality control". In: *Machine Learning and Knowledge Extraction* 6.1 (2023), pp. 1–17.

[12] E. van Gerwen. *Guided root cause analysis of machine failures - Status 2023*. Eindhoven: TNO, 2025. URL: https://publications.tno.nl/publication/34641994/nhUi7q/TNO-2024-R10109.pdf (visited on 12/01/2025).

[13] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., 1988. ISBN: 1558604790.

[14] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. Adaptive computation and machine learning. MIT Press, 2009. ISBN: 9780262013192.

[15] A. Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, 2009. ISBN: 0521884381.

[16] Z. Kato and J. Zerubia. *Markov Random Fields in Image Segmentation*. Foundation and Trends in Signal Processing. Sept. 2012, p. 164. URL: https://inria.hal.science/hal-00737058.

[17] M. Baron, C.K. Lakshminarayan, and Z. Chen. "Markov random fields in pattern recognition for semiconductor manufacturing". In: *Technometrics* 43.1 (2001), pp. 66–72.

[18] Z. Nadrich. *Statistical Inference and Applications of a Spatial-Temporal Markov Random Field*. American University, 2020.

[19] L. Barbini et al. In: *PHM Society Asia-Pacific Conference*. Vol. 5. 1. 2025. DOI: https://doi.org/10.36001/phmap.2025.v5i1.4608.

[20] Z. Zhang et al. “Hierarchical Multi-View Graph Pooling With Structure Learning”. In: *IEEE Transactions on Knowledge and Data Engineering* 35.1 (2023), pp. 545–559. DOI: 10.1109/TKDE.2021.3090664.

[21] S. Zhang et al. “Graph convolutional networks: a comprehensive review”. In: *Computational Social Networks* 6 (Nov. 2019). DOI: 10.1186/s40649-019-0069-y.

[22] S. Särkkä. *Bayesian Filtering and Smoothing*. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2013.

[23] A. Piedrafita and L. Barbini. “Leveraging Generative and Probabilistic Models for Diagnostics of Cyber-Physical Systems”. In: *PHM Society European Conference*. Vol. 8. 1. 2024, p. 7.

[24] A. Piedrafita, G.J. van den Braak, and L. Barbini. *Investigations on Probabilistic Programming Applications in Engineering*. Eindhoven: TNO, 2024. URL: https://publications.tno.nl/publication/34643165/dR4Z0zhu/TNO-2024-R11725.pdf (visited on 12/01/2025).

[25] Antonio Ortega et al. “Graph signal processing: Overview, challenges, and applications”. In: *Proceedings of the IEEE* 106.5 (2018), pp. 808–828.

ICT, Strategy & Policy

High Tech Campus 25
5656 AE Eindhoven
www.tno.nl

TNO innovation for life