Probabilistic Machine Learning for Uncertainty-Aware Diagnosis of Industrial Systems

Arman Mohammadi, Mattias Krysander, Daniel Jung, Erik Frisk

Abstract—Deep neural networks has been increasingly applied in fault diagnostics, where it uses historical data to capture systems behavior, bypassing the need for high-fidelity physical models. However, despite their competence in prediction tasks, these models often struggle with the evaluation of their confidence. This matter is particularly important in consistency-based diagnosis where decision logic is highly sensitive to false alarms. To address this challenge, this work presents a diagnostic framework that uses ensemble probabilistic machine learning to improve diagnostic characteristics of data driven consistency based diagnosis by quantifying and automating the prediction uncertainty. The proposed method is evaluated across several case studies using both ablation and comparative analyses, showing consistent improvements across a range of diagnostic metrics.

Index Terms—Neural networks for FDI, Uncertainty aware diagnostics, Ensemble probabilistic machine learning. Hybrid fault diagnosis, Structural analysis.

I. INTRODUCTION

A common strategy, known as Consistency-Based Diagnosis (CBD), frames fault detection and isolation as tests of whether observed behaviors are consistent with expectations. Faults are detected when observations deviate from the fault-free behavioral model, and isolated by checking inconsistencies across fault models. Traditionally, the developments relied on model-based approaches, using physical insight to generate residuals that indicate faults by comparing sensor measurements with expected outputs [1]. While effective, this approach requires detailed models and extensive expert knowledge, making it time-consuming and costly [2]. Machine learning, and specifically data-driven regression models, has been increasingly applied in diagnostics, where it uses historical data to capture nominal and faulty behaviors directly, bypassing the need for high-fidelity physical models [3]. Recent advances in deep learning have allowed data-driven regression models to identify subtle patterns and trends, making fault detection and isolation possible even in complex dynamic systems [4].

Despite their competence in achieving high accuracy in prediction tasks, deep neural networks often struggle in evaluation of their confidence in their estimation. A key factor is uncertainty, which comes in two primary forms. On one

hand, there is uncertainty intrinsic to the system, such as the inherent noise of measurement devices or the stochastic nature of system dynamics, referred to as *aleatoric uncertainty*. This type of uncertainty persists even with existence of infinite data; addressing it often requires additional information beyond the available features of the deep neural network. On the other hand, real-world environments frequently violate the assumption that training and testing data share the same underlying distribution. As a result, data-driven models often face difficulties when encountering conditions that differ from those seen during training. This leads to *epistemic uncertainty*, which reflects the model's lack of knowledge or exposure to certain scenarios. Unlike aleatoric uncertainty, epistemic uncertainty can be mitigated by collecting more diverse or representative data that better captures the operational space of the system [5].

In addressing the challenges of predictive uncertainty in data-driven models, [6] proposed a diagnostic framework that addresses the issue of overly confident predictions in neural networks, to avoid false alarms. This is especially critical in consistency-based diagnosis, where the decision logic is highly sensitive to false alarms. In the absence of alarms, all fault modes remain theoretical hypotheses, meaning the correct diagnosis still exists within the set of possible explanations. However, when a false alarm occurs, it eliminates valid hypotheses and reinforces incorrect conclusions. To manage uncertainty, the framework introduces separate, measurable parameters for each uncertainty type. Since aleatoric uncertainty represents intrinsic noise or chaos in the system output, it should not in itself trigger an alarm but can still allow for a reliable diagnosis. Thus, an adaptive threshold is applied to account for variations without overreacting. In contrast, epistemic uncertainty means the model is encountering unfamiliar conditions, so a diagnosis made under high epistemic uncertainty is unreliable. To address this, a One-Class Support Vector Machine (SVM) is used to detect out-of-distribution (OOD) samples and reject these residuals entirely. Figure 1 illustrates the proposed process for industrial applications.

A. Problem statement and contributions

While the existing frameworks handles predictive uncertainty in the examined case studies, it faces certain limitations. First, the use of One-Class SVM for anomaly detection may encounter difficulties in complex scenarios with high-dimensional data or dynamic relationships, due to its decision boundary definitions. Additionally, the identification of key features for addressing aleatoric uncertainty was performed empirically, which may limit the generalization of the proposed framework. To address these challenges, This work
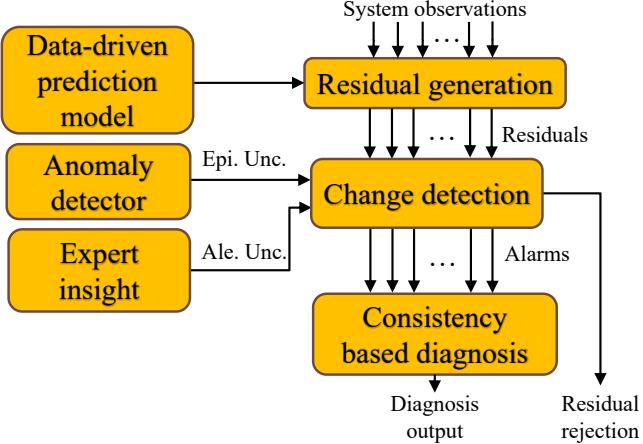
Fig. 1. The proposed process by [6] for developments made for industrial applications.

explores the advancements in predictive uncertainty of deep ensemble neural networks [7], to manage complex, high-dimensional data for gauging epistemic uncertainty while providing an automated measure for aleatoric uncertainty too. With respect to the previous works, this paper develops a diagnostic framework that accounts for uncertainty by combining deep ensemble neural networks with consistency-based decision logic. It provides practical guidance for designing and implementing the framework, including network architecture and training agenda. Finally, it introduces consistency-based evaluation metrics to assess performance across both simulation and industrial case studies.

B. Related research

Early advances have focused on adapting neural networks to incorporate uncertainty and probabilistic methods, primarily through a Bayesian approach [8]. This involves specifying a prior distribution over the network's parameters and, based on training data, computing a posterior distribution to quantify predictive uncertainty. Since exact Bayesian inference in neural networks is computationally intractable, various approximation techniques have been developed, including Laplace approximation [9], Markov Chain Monte Carlo (MCMC) methods [10], as well as advances in variational Bayesian methods [11]–[13], assumed density filtering [14], expectation propagation [15], and stochastic gradient MCMC approaches like Langevin dynamics [16] and Hamiltonian methods [17]. In practice, Bayesian neural networks can be more complex to implement and slower to train than non-Bayesian networks.

Gal and Ghahramani [18] proposed using Monte Carlo dropout (MC-dropout) to estimate predictive uncertainty by applying Dropout [19] at test time. This approach has been linked to an approximate Bayesian interpretation of dropout, and its simplicity has led to widespread practical use. Interestingly, dropout can also be interpreted as a form of ensemble model combination [19], where predictions are averaged across an ensemble of neural networks. This ensemble perspective may be more applicable, particularly in cases where

dropout rates are not tuned based on the training data, as any credible approximation to the true Bayesian posterior should be informed by the training data. This ensemble interpretation suggests that ensembles could serve as an alternative approach for estimating predictive uncertainty.

It has long been recognized that model ensembles improve predictive performance [20]. However, it is not always clear when and why an ensemble of neural networks can be expected to yield reliable uncertainty estimates. Bayesian model averaging (BMA) operates under the assumption that the true model lies within the prior's hypothesis class and performs a soft model selection to identify the single best model [21]. In contrast, ensembles combine multiple models to form a more robust predictor, which can be beneficial when the true model does not fall within the original hypothesis class. Related discussions on ensemble robustness and model selection can be found in [7], [22] and [21].

II. BACKGROUND

In this section, the principles of consistency-based diagnosis are summarized. Then, the design principles of data-driven residuals using a structural model of the system are presented. Finally, the theoretical background for uncertainty quantification using ensemble deep neural networks with adaptation from [23] is given.

A. Consistency based diagnosis

Fault diagnosis has traditionally been approached from two main directions: control theory, commonly referred to as Fault Detection and Isolation (FDI) [24], [25], and Artificial Intelligence (AI) [26]–[28]. FDI methods primarily focus on designing residual signals based on system models to detect deviations caused by faults, while development in AI, particularly consistency-based diagnosis, emphasize fault isolation by identifying conflicts between expected and observed behavior. A survey of the efforts to bridge these two communities, including hybrid approaches and integrative frameworks, can be found in [29]. In this work, we adopt a unified diagnostic framework proposed by [30], that uses diagnostic tests, such as residuals, designed to detect faults in specific parts of a system while remaining insensitive to faults in other areas.

By analyzing which residuals trigger alarms, a set of potential fault candidates can be identified, where each candidate represents a combination of faults explaining the observed inconsistencies [29]. Traditionally, model-based fault isolation relies on analyzing the model structure to determine which components are represented in each residual [31]. This is achieved by designing residuals with sensitivity to specific faults while being insensitive to others, often summarized in a fault signature matrix [32]. By integrating machine learning regression models into this consistency-based diagnostic framework, data-driven residuals can be generated that detect inconsistencies without requiring explicit physical models [33]. Furthermore, due to the high costs and challenges associated with collecting comprehensive fault data, particularly in the early stages of system development, when data from diverse fault scenarios is often limited and unlabeled [34], this

approach trains data-driven models on only nominal system behavior.

B. Residual design via structural analysis

Structural analysis is a useful tool for methodological design and analysis of diagnosis systems [32]. This method is particularly useful in early system design which allows for diagnosability analysis without requiring precise parameter values [1], [30]. A structural model can be represented as a bipartite graph $\mathcal{M} = (\mathcal{E}, \mathcal{X}, E)$ where $\mathcal{E}$ is the set of equations, $\mathcal{X}$ is the set of variables including all known and unknown variables and fault signals and E is a set of edges that encodes the correspondence between equations and variables [35]. In principle, achieving structural fault detectability depends on whether it is possible to design a residual generator that models the part of the system affected by the fault.

By utilizing a technique known as Dulmage-Mendelsohn (DM) decomposition on the structural model, it becomes feasible to perform various analyses, including fault detectability and isolability assessments, as well as the identification of redundant equation sets for residual generation [36]. The DM decomposition partitions the structural model into an under-determined $\mathcal{M}^-$, exactly determined $\mathcal{M}^0$ and an over-determined part $\mathcal{M}^+$. Different residual candidates can be constructed by identifying various subsets of the over-determined part of the model $\mathcal{M}^+$ that remains over-determined [36]. From a diagnosis perspective, the minimally over-determined equation sets, so-called minimally (structurally) over-determined (MSO) equation sets, are of special interest since these correspond to the smallest parts of the system that can be monitored separately [36].

Example II.1. *To illustrate the development and analysis of a structural model for fault diagnosis, this example uses the three-tank system modeling case presented in [32]. The system is described by the following set of equations:*

$$
\begin{aligned}
&e_1 : q_1 = \frac{1}{R_{V1}}(p_1 - p_2) + f_{V1}, && e_7 : y_1 = p_1, \\
&e_2 : q_2 = \frac{1}{R_{V2}}(p_2 - p_3) + f_{V2}, && e_8 : y_2 = q_2, \\
&e_3 : q_3 = \frac{1}{R_{V3}}(p_3) + f_{V3}, && e_9 : y_3 = q_0, \\
&e_4 : \dot{p}_1 = \frac{1}{C_{T1}}(q_0 - q_1) + f_{T1}, && e_{10} : \dot{p}_1 = \frac{dp_1}{dt}, \\
&e_5 : \dot{p}_2 = \frac{1}{C_{T2}}(q_1 - q_2) + f_{T2}, && e_{11} : \dot{p}_2 = \frac{dp_2}{dt}, \\
&e_6 : \dot{p}_3 = \frac{1}{C_{T3}}(q_2 - q_3) + f_{T3}, && e_{12} : \dot{p}_3 = \frac{dp_3}{dt},
\end{aligned}
\tag{1}
$$

where f_i are fault signals, y_i are known signals, C_i and R_i are known fixed parameters, and the remaining are unknown variables. Figure 2 shows the resulting structural model, constructed based on the component equations in (1). The structural model consists of 12 equations and 10 unknown variables, including three dynamic state variables (marked I) and their corresponding derivatives (marked D). Additionally, the model includes six fault variables and three known variables. Using the Fault Diagnosis Toolbox [32], three MSO
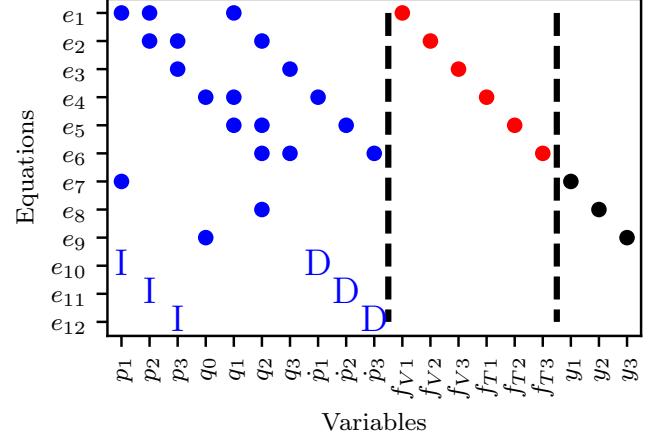

Fig. 2. Structural model of the three-tank system. Equations and variables are arranged to reflect causal and dependency relationships. Dynamic variables are marked as **I** (integral states) and **D** (their time derivatives).
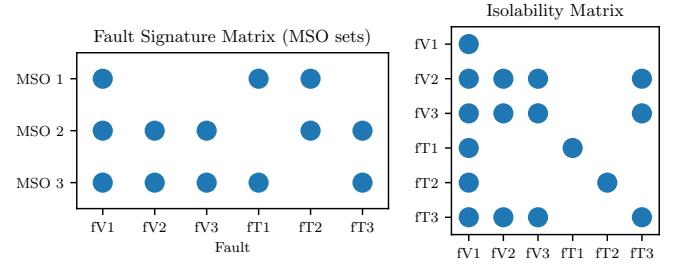

Fig. 3. Fault signature matrix and isolability matrix corresponding to three MSO sets.

sets that achieve maximum isolability are extracted. Figure 3 presents the corresponding fault signature matrix and fault isolability matrix derived from the given MSO sets. Each row in the fault signature matrix indicates which faults influence the equations in an MSO set. A dot in position (i, j) of the isolability matrix indicates that fault j is a diagnosis if fault i is the true fault.

Several approaches exist for constructing residual generators from a given MSO set. By removing one equation from an MSO set, what is left has an equal number of unknown variables and equations, creating an exactly determined set. This set can be used to determine how to compute all unknown variables from the known ones using a matching algorithm [30]. Then, the remaining equation, called the residual equation, can be used to compute unknown variables.

The creation of neural network-based residuals from MSO sets involves using neural networks to model discrete-time nonlinear representations of the states. In this model, the network structure is designed using the computational graph attained from the matching algorithm. The dynamic system is formulated in a state-space model, where unknown functions governing state transition computations are approximated using Multilayer Perceptron (MLP) blocks. By training residuals in an autoregressive manner on nominal system behavior, it is

possible to generate residual signals that are ideally sensitive to specific faults, as identified through structural analysis [6].

C. Uncertainty quantification of neural networks using ensemble models

Given a set of pairs of inputs and targets $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, a probabilistic predictive model approximates the true conditional probability distribution $p(y \mid x, \mathcal{D})$ with $q(y; f_\theta(x))$, where q belongs to some family of distributions parameterized by f_θ. In this paper, under the assumption that q is a Gaussian distribution, for each input x, the neural network outputs a parameter vector $z = f_\theta(x)$, where $z = (\hat{\mu}, \hat{\sigma}^2)$ represents the predicted mean and variance of the normal distribution. The network parameters θ are optimized in order to maximize the likelihood of the data with respect to $q(y; f_\theta(x))$.

The uncertainty in a model's prediction can be characterized using the estimated conditional probability. For a fixed value of θ, the model $q(y; f_\theta(x))$ will only capture aleatoric uncertainty. Conceptually, we can address this limitation with a Bayesian approach, learning a posterior distribution over the model parameters $p(\theta \mid \mathcal{D})$ and expressing the predictive distribution for a data point x^* as:

$$p(y^* \mid x^*, \mathcal{D}) = \int \underbrace{p(y^* \mid x^*, \theta)}_{\text{aleatoric}} \underbrace{p(\theta \mid \mathcal{D})}_{\text{epistemic}} \, d\theta. \tag{2}$$

More specifically, we can use this approach to define the different types of uncertainty:

$$\begin{aligned} U_{\text{tot}} &= \mathcal{I}\left[p(y \mid x, \mathcal{D})\right], \\ U_{\text{ale}} &= \mathbb{E}_{p(\theta \mid \mathcal{D})}\left[\mathcal{I}\left[p(y \mid x, \theta)\right]\right], \\ U_{\text{epi}} &= U_{\text{tot}} - U_{\text{ale}}, \end{aligned} \tag{3}$$

where $\mathcal{I}$ is some uncertainty measure, such as variance, entropy, or differential entropy [23]. Computing the posterior distribution $p(\theta \mid \mathcal{D})$ is intractable when f_θ is given by a deep neural network. A simple approach is training an ensemble of M independent models $\{f_{\theta_m}\}_{m=1}^M$ to provide a natural approximation of the posterior $p(\theta \mid \mathcal{D})$ to compute different types of uncertainty using (3), an idea that has been previously explored by [7], [37], [38].

III. PROPOSED FRAMEWORK

First, the derivation of formulation for quantification of epistemic and aleatoric uncertainty is shown. Then, the diagnostic framework that integrates the ensemble uncertainty in CBD decision logic is illustrated.

A. Uncertainty quantification for probabilistic regression models

By treating an ensemble of probabilistic neural networks as an independent and uniformly weighted mixture, we approximate the aggregated predictive distribution by matching its first and second moments. The ensemble estimate of the predictive mean $\hat{\mu}_*$ and variance $\hat{\sigma}_*^2$ are calculated as (See Appendix A):

$$\hat{\mu}_* = \frac{1}{M} \sum_{m=1}^M \hat{\mu}_m, \tag{4a}$$

$$\hat{\sigma}_*^2 = \frac{1}{M} \sum_{m=1}^M \hat{\sigma}_m^2 + \frac{1}{M} \sum_{m=1}^M (\hat{\mu}_m - \hat{\mu}_*)^2 \tag{4b}$$

where $\hat{\mu}_m$ and $\hat{\sigma}_m^2$ denote the predicted mean and variance, respectively, from the m-th model in the ensemble for a given input. Considering the variance representation of uncertainty, each model individually estimates the inherent variability of the output by predicting a variance $\hat{\sigma}_m^2$. Since the models are treated as independent samples from the posterior over parameters, the overall aleatoric uncertainty is approximated by the average of the predicted variances across all ensemble members. This corresponds to the Monte Carlo approximation of the expected predictive variance over the posterior distribution, as defined in (3). The total predictive uncertainty is represented by the variance of the aggregated predictive distribution $\hat{\sigma}_*^2$. Thus, the aleatoric and epistemic uncertainties can be approximated as:

$$U_{\text{ale}} \approx \frac{1}{M} \sum_{m=1}^M \hat{\sigma}_m^2, \quad U_{\text{epi}} \approx \frac{1}{M} \sum_{m=1}^M (\hat{\mu}_m - \hat{\mu}_*)^2 \tag{5}$$

Example III.1. *To illustrate the intuition behind this approach, consider a dataset where the training samples are drawn from the function $y = x^3 + \xi(x)$, where $\xi(x)$ represents noise as a function of the input x, simulating aleatoric uncertainty. The input x for the training data is sampled within the range $x \in [-2, 2]$, whereas the test set extends beyond this range, with $x \in [-3, 3]$. This setup introduces epistemic uncertainty in the regions $x \in [-3, -2]$ and $x \in [2, 3]$, as these input values lie outside the training distribution. Figure 4 illustrates this setup using (4) and (5) on 10 stacks of probabilistic predictive models of neural networks (referred to as PNNs throughout this paper). As it can be seen, aleatoric uncertainty is illustrated by the variance in the prediction within the training range, where noise $\xi(x)$ affects the data. Epistemic uncertainty exists in the extrapolated regions where the model's confidence decreases due to a lack of training data.*

B. Uncertainty-aware diagnostic framework

The proposed framework builds upon the structural analysis-based residual generation discussed earlier and an ensemble of probabilistic neural networks (PNNs) for regression tasks. The foundation of the approach lies in maintaining the MSO-based causal structure for each residual, with the expectation that each trained residual will exhibit the fault sensitivity pattern assumed in the structural analysis. Figure 5 provides a schematic view of the uncertainty-aware diagnostic framework. Each residual $r = y - \hat{\mu}_* \sim \mathcal{N}(0, \hat{\sigma}_*^2)$, is modeled using an ensemble PNN, which also quantifies aleatoric and epis-
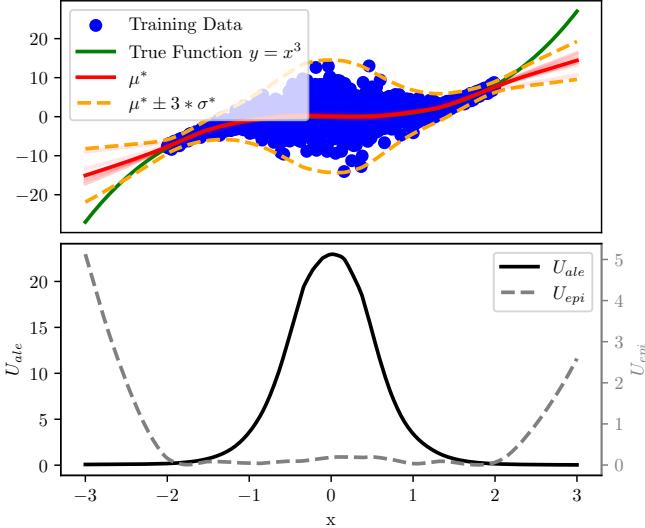
Fig. 4. Illustration of the training dataset, the true function $y = x^3$, ensemble predictions $\hat{\mu}_*$ and $\hat{\sigma}_*$ of 10 stacks of neural network, and the corresponding aleatoric and epistemic uncertainty.

Fig. 5. Overview of the proposed uncertainty-aware diagnostic framework. Diagnostic decisions are made based on the logic defined in (6).

temic uncertainty. The decision-making process is formalized as:

$$
\begin{aligned}
U_{\text{epi}} > \epsilon & \quad \text{out of range} \\
|r| \le J \quad \& \quad U_{\text{epi}} \le \epsilon & \quad \text{no conclusion} \\
|r| > J \quad \& \quad U_{\text{epi}} \le \epsilon & \quad \text{fault detected}
\end{aligned} \tag{6}
$$

Since uncertainty approximates the variance of the prediction, under the assumption of a Gaussian normal distribution with zero mean for the residual, we formulate the threshold design as a two-sided statistical detection problem. For a desired false alarm rate P_{fa}, the threshold is given by $J = \hat{\sigma}_* \Phi^{-1}(1 - P_{\text{fa}}/2)$, where Φ^{-1} is the inverse cumulative distribution function of the standard normal distribution. We define a threshold parameter $\alpha = \Phi^{-1}(1 - P_{\text{fa}}/2)$ such that $J = \alpha \hat{\sigma}_*$. For a 1% false alarm rate across the training set, this yields $\alpha = \Phi^{-1}(0.995) \approx 2.576$. The parameter ϵ determines when an out-of-distribution warning should be issued. By normalizing U_{epi} with respect to its maximum value, excluding the top 1% of anomalies in the training data, we set $\epsilon = 1$.

IV. EVALUATION AND RESULTS

This section presents the evaluation of the proposed diagnostic approach. It begins with a description of the datasets and case studies, followed by implementation and training agenda of the ensemble probabilistic neural network. The diagnostic performance is then analyzed through a set of defined evaluation metrics. Finally, an ablation and comparative study is conducted to further assess the contribution of each component within the diagnostic framework and compare its performance against existing methods.

A. Data sets and case study description

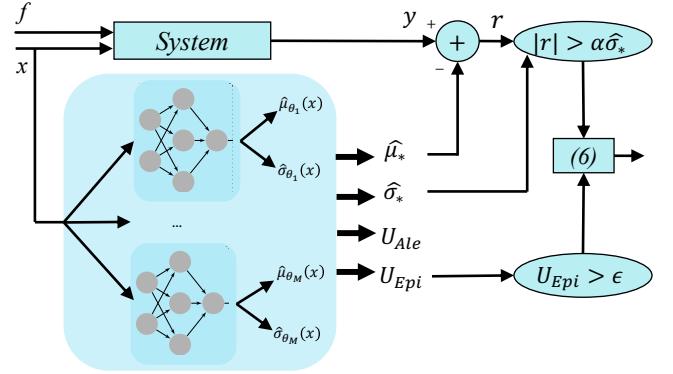Several diagnostic benchmarks are included in this study. The requirements for using them are the availability of a structural model, training data representing nominal system conditions, and test data from various fault scenarios.

1) Simulation of a two-tank system: Data and faults are generated in Simulink. The benchmark consists of a classic two-tank water system, where water is pumped into the upper tank and flows into the lower tank through interconnected pipelines. Fault candidates include actuator faults in the pump, sensor faults in level and flow measurements, leakages at various points, and partial obstructions in the connecting pipes. This benchmark is recognized and has been used extensively in the fault diagnosis literature, making it a relevant baseline for comparison. Its simulation-based nature allows controlled generation of data with desirable properties for research investigations [39]. A detailed schematic and fault descriptions are provided in Appendix B.

2) Aftertreatment in a heavy truck: This industrial case study is the Selective Catalytic Reduction (SCR) system used to reduce NOx emissions in diesel engines. The focus is on the subsystem responsible for dosing urea into the exhaust stream. Data is collected in a workshop environment under various drive cycles and fault conditions, primarily targeting clogging faults. The setup includes signals from pressure sensors and the Electronic Control Unit (ECU). Data from different fault scenarios are generated by replacing faulty components in the experimental setup. The collected fault scenarios considered in this study are motivated by previous workshop experiences. This benchmark is included due to its experimental nature, serving as a realistic setting to evaluate the diagnostic algorithms. Since the dataset is pre-logged, it poses practical challenges typical of real-world applications, such as uncontrolled conditions and limited ability to influence fault scenarios [6]. Details including the system schematic, fault types and data collection methodology are provided in Appendix C.

3) Air-path of an automotive gasoline engine: This case study focuses on the air path of a turbocharged gasoline engine, a critical subsystem for fuel injection, emissions control, catalyst protection, and efficiency optimization. Due to strong interactions via the exhaust turbo and intake compressor, faults in almost any part of the air path affect multiple measured variables. The benchmark dataset, introduced in [40], was

collected from an engine test bench and includes data from nominal operations and various fault scenarios, such as sensor faults and intake manifold leakage. This dataset is publicly available and has been established as a benchmark, making it suitable for reproducible evaluation. Moreover, the system's complexity and high-dimensional input space, along with its erratic behavior under fault conditions, pose a challenge to diagnostic algorithms. Detailed descriptions, system schematics and fault scenarios are provided in Appendix D.

B. Training and implementation details of the ensemble probabilistic neural network

The core model architecture employed in this work is a recurrent neural network designed for probabilistic sequence modeling. The model consists of a single-layer Long Short-Term Memory (LSTM) network followed by two separate fully connected layers: one for estimating the mean and another for the standard deviation of the predictive distribution. Three different data sets mentioned in previous section were used for training, validation and testing, each representative of a different diagnostic benchmark. The nominal and fault free operation is used during training and validation, while fault data and other set of nominal data is used for testing of the diagnostic performance. All implementations are performed in Python using the PyTorch library. The optimizer used is Adam. Batch size, learning rate, weight decay rate, hidden dimensionality of network and total number of epochs are tuned independently for each experiment.

1) Scheduling of the training objective: The overall training procedure is summarized in Algorithm 1. In this work, the training objective is scheduled to change from Mean Squared Error (MSE) to Negative Log-Likelihood (NLL) under a Gaussian distribution assumption. This scheduling helps to warm up the model for the regression task before transitioning to likelihood-based training. Additionally, during the warm-up phase the prediction horizon is increased gradually, to allow the model to have a stable autoregressive prediction for full sequence length provided in the training set. Let H denote the full prediction horizon, H_{init} the initial horizon, and ΔH the step size by which the horizon is increased during training. The model parameters are denoted as $\theta_m = \theta_m^{(\mu)} \cup \theta_m^{(\sigma)}$, where $\theta_m^{(\mu)}$ includes the parameters of the LSTM and the feedforward layer responsible for predicting the mean of the output distribution $\hat{\mu}$. The remaining parameters, $\theta_m^{(\sigma)}$, correspond to the feedforward layer that models the standard deviation of the distribution $\hat{\sigma}$. The warm-up phase lasts for τ_{w} epochs, during which the model is trained with $\mathcal{L}_{\text{MSE}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{\mu}_i)^2$ in which $\theta_m^{(\sigma)}$ are frozen. After the warm-up phase, the training objective is switched to,

$$\mathcal{L}_{\text{NLL}} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{(y_i - \hat{\mu}_i)^2}{2\hat{\sigma}_i^2} + \frac{\log \hat{\sigma}_i^2}{2}\right) + \text{const.}$$

where $\theta_m^{(\mu)}$ are frozen and $\theta_m^{(\sigma)}$ are trained for τ epochs.

C. Results

The results presented in this section are derived based on the following stages. First, for each case study, a structural

---

Algorithm 1 Two-phase training with horizon increase
***
Ensure: Trained parameters $\{\theta_m\}_{m=1}^{M}$
1: Initialize each θ_m
2: $H_{\text{curr}} \leftarrow H_{\text{init}}$
3: **for** $e = 1$ to τ_w/n **do**
4:   Train each $\theta_m^{(\mu)}$ ($\mathcal{L}_{\text{MSE}}$) over horizon H_{curr}
5:   $H_{\text{curr}} \leftarrow \min(H_{\text{curr}} + \Delta H, H)$
6: **end for**
7: **for** $e = 1$ to τ **do**
8:   Train each $\theta_m^{(\sigma)}$ ($\mathcal{L}_{\text{NLL}}$) over full horizon H
9: **end for**
10: **return** $\{\theta_m\}_{m=1}^{M}$

---

analysis is conducted to identify a set of candidate residual configurations. Subsequently, each network is trained using Algorithm 1 on a nominal dataset, which represents the corresponding subsystem under normal operating conditions, free of faults. The results shown in this section are obtained by evaluating the trained models on new datasets containing both nominal and faulty system behavior. Due to the nature of consistency-based diagnosis, which relies on residuals in an autoregressive fashion, all evaluations are performed on the full sequence of data.

Example IV.1. *Figure 6 illustrates a representative output of the diagnostic framework using one example residual from the aftertreatment system. The figure consists of a 3×3 grid, where each column corresponds to a different operating mode. The left column represents nominal behavior, the middle column a fault that is sensitive to the selected residual, and the right column a fault that is structurally decoupled from the residual. The first row shows the residual signals and their associated adaptive thresholds. The second row presents the epistemic uncertainty estimated by the model, with a black dashed line as the OOD detection threshold. The third row visualizes the resulting decision according to* (6). *The left column shows minimal anomalies and false alarms, as the nominal data aligns well with the training distribution. The middle column, a sensitive fault, shows both alarms and anomalies, since the fault causes a shift outside the valid range. The right column, a decoupled fault, mainly triggers OOD rather than alarms.*

To get an overall perspective, two performance matrices are computed. The first summarizes the residual-level alarm behavior (corresponding to the detection of faults according to fault signature matrix) and the second matrix captures the diagnostic-level performance of the overall system after applying the consistency-based decision logic considering all the residuals (corresponding to the isolability matrix). Figure 7 represents the residual-level alarm behavior applied to the two-tank system. The number in position (i, j) is the probability s_{ij} in percent that fault f_j enriches the diagnosis in r_i, e.g. in case of using the framework decision logic,

$$s_{ij} = P(|r_i| > J_i \quad \& \quad U_{\text{epi,i}} \leq \epsilon \mid f_j \text{ true fault}).$$

The fault sensitivity of each residual, represented by the fault signature matrix, is encoded in the color of the numbers. Black
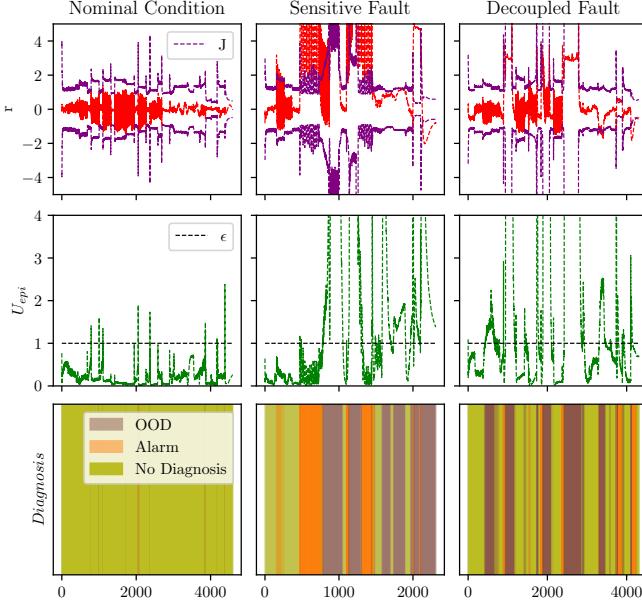
Fig. 6. Illustration of an example of a residual signals, adaptive thresholds, and epistemic uncertainty across different fault scenarios from the aftertreatment system.

Fig. 7. Probability of alarm for each residual with respect to different data sets used in the analysis for the simulation study. Black numbers indicate decoupled and red numbers indicate sensitive faults according to the decision structure. The number in parentheses corresponds to the change compared to the traditional alarm approach.

numbers indicate faults that are expected to be decoupled, meaning the alarm percentage should be close to zero, while red numbers indicate faults that should be detected by the residuals. In order to provide a sense of comparison, each cell also includes the difference in probability (in parentheses) when using the proposed decision framework, compared to traditional alarm generation based on a prediction model with a fixed threshold corresponding to a 1% false alarm rate across the training set, without any anomaly detection mechanism. As shown in Figure 7, a clear reduction in alarm rates is observed across both sensitive and non-sensitive faults. It is important to note that the primary objective of the framework is to reduce false alarms while preserving the true detection capability of each residual for its associated fault cases. This behavior is observed across several instances for example, false alarms in faults f_a, f_{c1} and f_{l1} in residuals r_0, r_1, r_2, r_3 and r_5 are largely reduced. The only notable instance of low performance is observed in the case of fault f_{c2} in residual r_3, where the method fails to suppress false alarms. This is due to the neural network not capturing the true dynamics between the signals, instead learning statistical correlations that do not reflect the actual system behavior. This leads to persistent false alarms, an aspect that is discussed in detail in Section V.

In order to quantify diagnosis performance, all residual generators have been run on all data, minimal diagnoses have been computed for each sample. The diagnosis results for the two tank system are summarized in a fault isolation performance matrix shown in Figure 8. The number in position (i, j) is the probability p_{ij} in percent that fault f_j is a diagnosis given that the true fault is f_i, i.e.,

$$p_{ij} = P(f_j \text{ diagnosis} \mid f_i \text{ true fault}).$$

The evaluation results in Figure 8 can then be considered as the performance of the diagnosis system design using CBD decision logic. The values in parentheses indicate the difference in performance between the proposed framework and the traditional approach based on fixed-threshold alarms without anomaly detection. The structural isolation property of the system, as derived from the structural model, is encoded using colors with red indicates fault modes that are structurally sensitive and black for isolated faults. Note that consistency-based diagnosis is not a classification approach computing exactly one diagnosis. In consistency-based diagnosis, modes are rejected if there is enough evidence to do so. This means that the row sum in the matrix can be greater than 1, even though the ideal scenario is that the matrix is diagonal. The results show a clear improvement in the detection of faults f_a, f_{c1}, and f_{l1}, due to a reduction in false alarms.

A similar evaluation is performed on the other two case studies, and the corresponding result tables can be found in Appendix C and D. Given the extensive number of evaluations and results presented in this study, a set of comparative metrics is introduced to provide a systematic assessment of diagnostic model performance.

D. Comparison metrics

Selecting an appropriate diagnostic metric extends beyond simply evaluating loss function values. Although achieving a low training and validation loss provides an initial indication of effective model training, diagnostic performance must also encompass reliable fault detection and isolation. In the consistency-based framework adopted here, the model's performance is therefore assessed by examining its performance along several diagnostic metrics.

Consider fault signature matrix T_{ij} which corresponds to the sensitivity for each residual i to fault data label j (e.g. Figure 3). Let N_0 denote the number of pairs (i, j) for which

Injected fault \ Diagnosed fault	nf	fa	fc1	fc2	fl1	fl2	fl3	fh1	fh2	ff1	ff2
nf	98.8 (3.1)	0.3 (-1.3)	0.2 (-1.3)	0.3 (-1.4)	0.7 (-0.9)	0.1 (-1.7)	0.2 (-1.4)	0.9 (-0.8)	0.3 (-1.3)	0.6 (-1.2)	0.2 (-1.4)
fa	15.9 (1.1)	83.9 (80.5)	83.6 (80.7)	1.0 (0.6)	1.1 (0.8)	1.0 (0.6)	0.0 (-0.2)	5.1 (2.2)	1.0 (0.8)	0.1 (-0.2)	0.0 (-0.2)
fc1	15.7 (1.7)	33.6 (32.9)	74.3 (73.9)	2.6 (2.2)	5.0 (3.5)	2.6 (2.2)	0.0 (-0.2)	2.1 (0.6)	2.6 (2.4)	1.6 (0.2)	0.0 (-0.2)
fc2	14.4 (0.5)	0.1 (-0.6)	0.0 (-0.4)	8.1 (7.7)	0.1 (-0.2)	17.7 (17.4)	3.1 (1.1)	0.3 (-0.2)	1.0 (0.8)	1.9 (1.7)	3.1 (1.1)
fl1	16.0 (2.0)	67.2 (66.5)	81.5 (81.1)	67.1 (66.7)	82.4 (80.9)	67.8 (67.4)	0.0 (-0.2)	1.8 (0.3)	68.3 (68.1)	1.6 (0.2)	0.0 (-0.2)
fl2	17.1 (2.0)	1.9 (0.3)	1.8 (0.5)	81.9 (0.3)	2.0 (0.9)	81.5 (3.3)	0.4 (0.2)	0.3 (-0.3)	1.8 (0.8)	0.5 (0.2)	0.4 (0.2)
fl3	14.4 (0.5)	0.1 (-0.6)	0.0 (-0.4)	54.2 (53.8)	0.1 (-0.2)	83.4 (83.1)	84.6 (1.2)	0.3 (-0.2)	1.0 (0.8)	83.9 (83.7)	84.6 (1.2)
fh1	15.8 (1.1)	4.2 (0.7)	2.8 (0.8)	0.0 (-0.4)	1.1 (0.8)	0.0 (-0.4)	0.0 (-0.2)	83.9 (1.2)	0.0 (-0.2)	1.1 (0.8)	0.0 (-0.2)
fh2	14.7 (0.7)	0.2 (-0.5)	0.1 (-0.3)	1.1 (0.7)	0.2 (-0.1)	1.0 (0.7)	0.4 (0.2)	0.3 (-0.2)	84.4 (1.1)	0.1 (-0.1)	0.4 (0.2)
ff1	28.1 (14.1)	1.9 (1.1)	8.1 (7.6)	1.3 (0.9)	56.2 (54.3)	1.3 (0.9)	1.2 (1.0)	56.4 (54.2)	0.1 (-0.1)	71.3 (-12.2)	1.2 (1.0)
ff2	15.6 (1.4)	0.1 (-0.6)	0.0 (-0.4)	69.0 (68.7)	0.2 (-0.1)	68.0 (67.7)	83.4 (0.3)	0.4 (-0.1)	0.2 (0.0)	68.4 (68.2)	83.4 (0.3)

Fig. 8. The fault isolation performance matrix for the simulation study. Black numbers indicate decoupled and red numbers indicate sensitive faults according to the isolability matrix. The number in parentheses corresponds to the change compared to the traditional alarm apprach.

$T_{ij} = 0$ and N_1 the number of pairs (i, j) for which $T_{ij} = 1$. The average false alarm rate is denoted as

$$S_{\text{FA}} = \frac{1}{N_0} \sum_{i,j:T_{ij}=0} s_{ij} \qquad (7)$$

and the average missed detection denoted rate is denoted as

$$S_{\text{MD}} = 1 - \frac{1}{N_1} \sum_{i,j:T_{ij}=1} s_{ij}. \qquad (8)$$

These metrics are evaluating the deviation of residuals in behaving corresponding to their decision table. The next metrics are inspired by CBD interpretation of alarms proposed by [41]. Let D denote the set of diagnoses, n_f the number of faults, $\tilde{F}$ the set of fault modes without the no-fault mode, and I_{ij} the value in the structural isolability matrix in position (i, j) (e.g, Figure 3). The diagnosis performance measures considered here are the false alarm probability

$$p_{\text{FA}} = 1 - P(NF \in D|NF), \qquad (9)$$

the mean missed detection probability

$$p_{\text{MD}} = \frac{1}{n_f} \sum_{f_i \in \tilde{F}} P(NF \in D|f_i), \qquad (10)$$

and aggregated detection error

$$p_{\text{D}} = \frac{1}{n_f^2} \sum_{f_i \in \tilde{F}} P(NF \notin D|f_i) \sum_{f_j:I_{ij}=1} |P(f_j \in D|f_i) - I_{ij}| \qquad (11)$$

that ideally should be 0. While additional diagnostic metrics such as time to detection and isolation may offer further insights, these are not emphasized as primary comparison metrics, given their heavy dependence on the particular detection scheme employed in post-alarm processing stages.

E. Ablation and comparative study

To gain insight into the contributions of different components within our work, we have conducted an ablation study. The objective of this study is to assess the impact of key elements on model performance by removing specific mechanisms and analyzing the resulting behavior. To ensure fairness we excluded any usage of evaluation data, including any data containing fault modes, from the design and tuning process. Some metrics, such as missed detection, can only be meaningfully compared when other metrics, like false alarms, are held constant. However, achieving such consistency requires a complex and multidimensional thresholding and evaluation process that would inherently involve evaluation data. To prevent this and remain true to real-world diagnostic design principles of this study, we focused on designing models using only the nominal training data. Each ablation experiment was performed independently, with the modified models trained and evaluated under identical conditions. Two components were considered here:

- *Out of distribution*: Epistemic uncertainty, which captures model uncertainty due to limited data, is estimated using an ensemble of PNNs and used as a measure of OOD. To assess its significance, we followed the original CBD logic using only alarms.
- *Adaptive thresholding*: J is modeled through the predicted variance in each PNN. To examine its impact, we replaced the adaptive thresholding with a fixed threshold selected to yield a 1% false alarm rate on the training set.

As a point of comparison with prior work in the literature [6], we replaced the OOD indicator with a SVM anomaly detector.

- *SVM-based anomaly detection (comparative study)*: The One-Class SVM model was tuned to achieve a 1% anomaly rate on the model input of the training set, corresponding to the anomaly level captured by the epistemic uncertainty measure.

Table I presents the results of the ablation study. Removing both components influences the most critical metrics, the false alarm rate (S_{FA}) and the probability of false alarm (p_{FA}). Eventhough, this comes at the cost of an increased missed detection (S_{MD} and p_{MD}), the aggregated detection error (p_{D}) has its lowest value in the presence of both components in decision logic. The same pattern is observed for removal of each individual component accross all case studies.

The comparison study reveals that the performance of ensemble probabilistic regression models and SVM-based anomaly detection can vary across different application domains. The ensemble method outperforms in the engine datasets, while in the two-tank and aftertreatment system datasets, the SVM demonstrates better performance. A consideration when comparing ensemble models with alternative

TABLE I

ABLATION STUDY RESULTS ACROSS DIFFERENT SCENARIOS. A ✓ INDICATES THE PRESENCE OF THE COMPONENT, WHILE AN ✗ INDICATES ITS REMOVAL.

System	Scenario	OOD	Adaptive J	$S_{\mathbf{FA}}(\%)$	$S_{\mathbf{MD}}(\%)$	$p_{\mathbf{FA}}(\%)$	$p_{\mathbf{MD}}(\%)$	$p_{\mathbf{D}}(\%)$
	T1	✓	✓	**2.224**	43.828	**1.206**	16.753	**1.132**
	T2	✓	✗	2.790	43.245	4.221	15.718	1.365
Two-tank	T3	✗	✓	9.112	23.047	1.206	15.075	3.498
	T4	✗	✗	14.103	**21.138**	4.321	**14.241**	4.230
	T5	SVM	✓	**1.837**	45.022	1.206	17.708	**0.938**
	AT1	✓	✓	**3.950**	72.870	**3.652**	50.886	**2.895**
	AT2	✓	✗	8.681	66.083	11.789	41.899	5.051
Aftertreatment	AT3	✗	✓	4.657	64.002	3.791	47.890	3.273
	AT4	✗	✗	13.773	**51.139**	15.025	**36.538**	6.231
	AT5	SVM	✓	**3.178**	66.162	3.791	49.667	**2.505**
	E1	✓	✓	**0.407**	67.348	**0.501**	38.358	**0.656**
	E2	✓	✗	2.144	65.380	2.258	37.041	1.590
Engine	E3	✗	✓	0.731	57.972	0.509	35.885	0.803
	E4	✗	✗	4.788	**54.298**	2.531	**34.632**	2.925
	E5	SVM	✓	0.725	64.285	0.509	41.701	0.950

anomaly detection methods, such as SVMs, lies in how these models handle deviations in the input space. One-Class SVM enforces boundaries around the training distribution, flagging any deviation as an anomaly. In contrast, ensemble method estimates uncertainty based only on learned patterns. If a feature was not influential during training, deviations in that feature will not necessarily trigger high uncertainty, even if the feature shift indicates an anomaly. This can lead to blind spots where anomalies go undetected. SVM, on the other hand, explicitly models the entire feature distribution and detects anomalies regardless of whether a feature was important during training. However, a notable limitation of SVM-based approaches is their susceptibility to the curse of dimensionality. As the input dimensionality increases, the data becomes sparser, distance metrics become less informative, and decision boundaries become less reliable, which can degrade performance in high-dimensional feature spaces [42].

V. DISCUSSION

Several key assumptions underpin the logic of ensemble probabilistic neural networks in this work. One of the main statistical assumptions is that the prediction distributions follow a normal Gaussian distribution. Another crucial assumption is that the individual model predictions are independent, however, this assumption is not strictly valid, as all models in the ensemble are trained on the same dataset. Despite this limitation, ensemble method remains effective by utilizing the randomness introduced through different weight initialization and training dynamics.

The performance of the adaptive thresholding mechanism within the proposed framework is highly dependent on the availability of correlation to aleatoric uncertainty source as a feature in the set of available model inputs. If the necessary excitation required for identifying variance in predictions is absent from the input features, the model will struggle to learn meaningful variance estimates.

While the use of an OOD indicator helps to reduce false alarms by filtering anomalies arising from operational shifts, it can also lead to an increase in missed detections. This trade-off is due to the presence of two distinct types of anomalies, those caused by operating point changes and those stemming from actual faults. Rejecting anomalies related to operational shifts is beneficial for reducing false alarm rates. However, if the OOD mechanism also suppresses fault-induced anomalies, it may impair the diagnostic model's ability to detect real faults (p_{D}). In the case studies presented here, the probability of detection and hence the overall diagnostic performance improved across all applications, as the number of benign anomalies outweighed the missed fault-related anomalies. Nevertheless, this balance is data-dependent, and in principle, the use of OOD indicators could degrade detection performance if critical fault signatures are mistakenly treated as outliers. Note that structural analysis can be helpful in this context. While the MSO sets are derived from the structural analysis of the model, the selection of causality used in residual design remains a choice, as it influences the valid input range associated with different causalities. This is an important aspect that can be used to improve the diagnostic quality derived from data-driven residuals, as discussed in [43].

A neural network-based residual, trained on fault-free data and aligned with structural analysis causality, offers a powerful tool for modeling complex system behavior and strong candidate for residual generation of consistency based diagnosis. However, these networks do not necessarily conform to the sensitivity structure defined in the decision matrix. Neural networks primarily learn patterns based on statistical correlations rather than explicitly capturing the true causal relationships between signals. When strong correlations exist between specific inputs and the target signal, the network may prioritize these over the underlying causal dynamics, potentially leading to missed detections due to the loss of critical connections necessary for identifying faults. Moreover, fault isolation in structural models typically relies on the assumption that the residual model accurately reflects the true dynamics between signals. However, neural networks, which often focus on correlations rather than causation, may violate this assumption, resulting in false alarms as normal variations are erroneously classified as faults. One way to address this issue, as explored by [6], is through the use of augmented

training data tailored to each residual generator. One benefit of employing structural analysis is the ability to divide a system into submodels. This allows for the individual definition of each submodel's nominal performance. Consequently, data collected from a specific fault scenario can serve as the nominal performance for one submodel while representing a fault for another. By enhancing the training set with data from a broader range of nominal operating conditions, including data from decoupled faults that are non-sensitive to a specific residual, the model can learn a more accurate representation of valid behaviors.

VI. CONCLUSION

In this paper, we have presented an uncertainty aware data-driven diagnostic framework for consistency based diagnosis, using ensemble probabilistic neural networks based on physical causal relationships. The uncertainty characterization is integrated into the diagnostic decision-making process, providing an OOD rejection mechanism that issues warnings, and an adaptive thresholding strategy that adjusts the diagnostic scrutiny according to the stochastic nature of the data. The framework was evaluated using consistency-based performance metrics, and experimental results on both simulated and real-world datasets show the capabilities of the proposed architecture and training pipeline.

APPENDIX A
DERIVATION OF ENSEMBLE PREDICTIVE MEAN AND VARIANCE

We consider an ensemble of M independent probabilistic models, each producing a Gaussian predictive distribution $\mathcal{N}(\hat{\mu}_i, \hat{\sigma}_i^2)$ for a given input. Treating this ensemble as a uniformly weighted mixture of Gaussians, we approximate the aggregate predictive distribution by matching its first and second moments. The predictive mean of the mixture is obtained by the linearity of expectation:

$$\hat{\mu}_* = \mathbb{E}[Y] = \frac{1}{M}\sum_{i=1}^{M}\mathbb{E}[Y \mid i] = \frac{1}{M}\sum_{i=1}^{M}\hat{\mu}_i.$$

To compute the predictive variance, we apply the law of total variance:

$$\hat{\sigma}_*^2 = \mathrm{Var}(Y) = \mathbb{E}[\mathrm{Var}(Y \mid i)] + \mathrm{Var}(\mathbb{E}[Y \mid i]).$$

The first term, $\mathbb{E}[\mathrm{Var}(Y \mid i)]$, corresponds to the average of the individual predictive variances. Using the definition of variance $(\mathrm{Var}(X) = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right]$, and substituting $X = \hat{\mu}_i$) the second term becomes:

$$\mathrm{Var}(\mathbb{E}[Y \mid i]) = \mathbb{E}\left[(\hat{\mu}_i - \hat{\mu}_*)^2\right] = \frac{1}{M}\sum_{i=1}^{M}(\hat{\mu}_i - \hat{\mu}_*)^2.$$

Putting both terms together, the ensemble predictive variance becomes:

$$\hat{\sigma}_*^2 = \frac{1}{M}\sum_{i=1}^{M}\hat{\sigma}_i^2 + \frac{1}{M}\sum_{i=1}^{M}(\hat{\mu}_i - \hat{\mu}_*)^2.$$

Fig. 9. Schematic view of two coupled water tanks.

TABLE II
POSSIBLE FAULTS IN THE TWO-TANK SYSTEM

Fault ID	Description
Fa	Actuator fault in the pump
Fh1	Fault in water level sensor y_1 (Tank 1)
Fh2	Fault in water level sensor y_2 (Tank 2)
Ff1	Fault in flow sensor y_3 (between tanks)
Ff2	Fault in flow sensor y_4 (outflow)
Fl1	Leakage between Tank 1 and sensor 3
Fl2	Leakage between sensor 3 and Tank 2
Fl3	Leakage between Tank 2 and sensor 4
Fc1	Partial obstruction between Tank 1 and Tank 2
Fc2	Partial obstruction after Tank 2

APPENDIX B
TWO TANK

The water tank system used in this study is illustrated in Figure 9. The system consists of two vertically aligned tanks, with a pump delivering water into the upper tank (Tank 1). The water then flows to the lower tank (Tank 2), and exits from there. The control input u regulates the pump based on the level in Tank 1. The system includes four sensor measurements: y_1 measures the water level in Tank 1, y_2 measures the water level in Tank 2, y_3 captures the flow between Tank 1 and Tank 2, and y_4 monitors the outflow from Tank 2. Table II summarizes the types of faults introduced in the simulation. These cover sensor faults, actuator faults, leakages, and flow obstructions. The governing equations used to create the structural model for this case study can be found in [39]. The selection of residuals is based on the maximum isolability and detectability of faults.

APPENDIX C
AFTERTREATMENT SYSTEM

The aftertreatment system case study focuses on the Selective Catalytic Reduction (SCR) subsystem in a diesel engine, which reduces nitrogen oxide (NOx) emissions to comply with regulatory standards. A reducing agent, typically urea, is injected into the exhaust gas stream where it reacts with NOx to produce nitrogen and water. Figure 10 illustrates the dosing subsystem responsible for controlling the urea injection. Urea is pumped from a storage tank through a series of hoses and filters to a dosing chamber, where it is injected into the exhaust via a nozzle. The governing equations used to create the structural model for this case study can be found in [6].

The system features several key signals: three pressure sensors positioned before the pump filter ($y_{p,tp}$), after the

TABLE III
SUMMARY OF FAULT SCENARIOS AND COMPONENT FAILURES IN AFTERTREATMENT SYSTEM.

Label	Symbol	Intensity	Description	Component
PAS	f_{PAS}	Mild	Clogging after pressure sensor	Heating circuit
PBS	f_{PBS}	Mild	Clogging before pressure sensor	Pump main filter
RBS1/2/3	f_{RBS}	Mild/Mod/Severe	Return pipe clogging	Orifice
SAS1/2/3	f_{SAS}	Mild/Mod/Severe	Suction pipe clogging after sensor	Pump input filter
SBS1/2/3	f_{SBS}	Mild/Mod/Severe	Suction pipe clogging before sensor	Urea tank filter

pump ($y_{p,ap}$), and inside the dosing unit ($y_{p,du}$), as well as two signals from the ECU, the pump speed (n_p) and the dosing control signal (u_{DC}). The control signal is pulse-width modulated, and u_{DC} refers to its duty cycle.
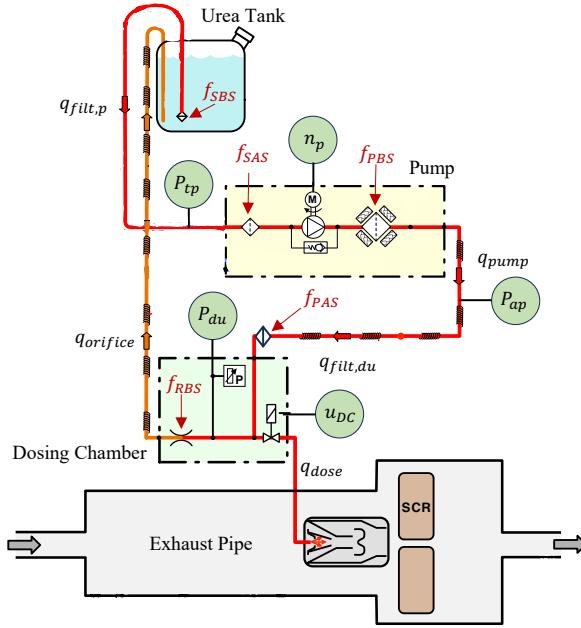
Fig. 10. Schematic view of components in the aftertreatment system model. The red line represents the supply pipe, and the orange line represents the return pipe. Available signals are shown in green, and fault locations are indicated in red.

Operational data was collected at a sampling rate of 10 Hz under different drive cycles simulating both nominal and faulty conditions. Due to hardware and bandwidth constraints, signals were initially recorded at varying frequencies, which were later synchronized via interpolation. Nominal operation includes a baseline drive cycle followed by a pressure-altering cycle designed to excite dynamic system responses. Clogging faults were identified by the industrial partner as the most critical for system maintenance. Consequently, data from a range of clogging scenarios was collected by physically replacing standard components with partially obstructed ones. These faults were categorized by location and severity: suction-side clogging (SAS, SBS), return-side clogging (RBS), and pressure-side clogging (PAS, PBS). Some scenarios include multiple severity levels (mild, moderate, and severe), as summarized in Table III.

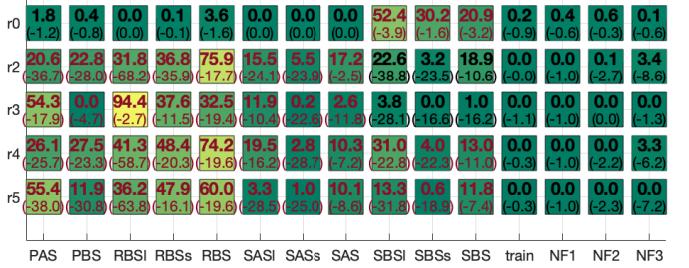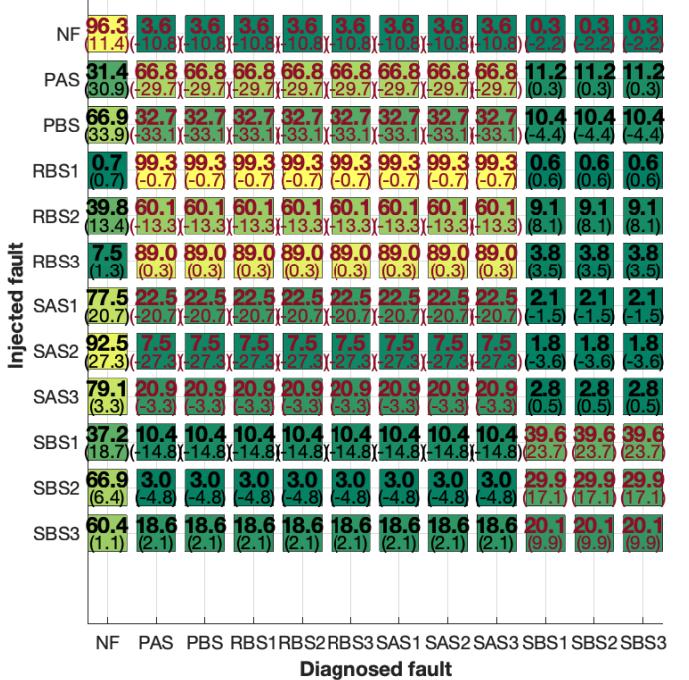The selection of residuals is based on the maximum isolability and detectability of faults, following the approach inspired by [6]. The illustration of sensitivity and diagnosis results for the aftertreatment system is shown in Figure 11 and Figure 12, respectively.

Fig. 11. Aftertreatment systems sensitivity matrix.

	PAS	PBS	RBSl	RBSs	RBS	SASl	SASs	SAS	SBSl	SBSs	SBS	train	NF1	NF2	NF3
r0	1.8 (-1.2)	0.4 (-0.8)	0.0 (0.0)	0.1 (-0.1)	3.6 (-1.6)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	52.4 (-3.9)	30.2 (-1.6)	20.9 (-3.2)	0.2 (-0.9)	0.4 (-0.6)	0.6 (-0.3)	0.1 (-0.6)
r2	20.6 (-36.7)	22.8 (-28.0)	31.8 (-68.2)	36.8 (-35.9)	75.9 (-17.7)	15.5 (24.1)	5.5 (23.9)	17.2 (-2.5)	22.6 (-38.8)	3.2 (23.5)	18.9 (-10.6)	0.0 (-0.0)	0.0 (-1.0)	0.1 (-2.7)	3.4 (-8.6)
r3	54.3 (-17.9)	0.0 (-4.7)	94.4 (-2.7)	37.6 (-11.5)	32.5 (-19.4)	11.9 (10.4)	0.2 (22.6)	2.6 (11.8)	3.8 (-28.1)	0.0 (-16.6)	1.0 (-16.2)	0.0 (-1.1)	0.0 (-1.0)	0.0 (0.0)	0.0 (-1.3)
r4	26.1 (-25.7)	27.5 (-23.3)	41.3 (-58.7)	48.4 (-20.3)	74.2 (-19.6)	19.5 (16.2)	2.8 (28.7)	10.3 (-7.2)	31.0 (-22.8)	4.0 (-22.3)	13.0 (-11.0)	0.0 (-0.3)	0.0 (-1.0)	0.0 (-2.2)	3.3 (-6.2)
r5	55.4 (-38.0)	11.9 (-30.9)	36.2 (-63.8)	47.9 (-16.1)	60.0 (-19.6)	3.3 (28.5)	1.0 (25.0)	10.1 (-8.6)	13.3 (-31.8)	0.6 (-18.5)	11.8 (-7.4)	0.0 (-0.3)	0.0 (-1.0)	0.0 (-2.3)	0.0 (-7.2)

Fig. 12. Aftertreatment systems fault isolation performance matrix.

Injected fault (rows) × Diagnosed fault (columns):

	NF	PAS	PBS	RBS1	RBS2	RBS3	SAS1	SAS2	SAS3	SBS1	SBS2	SBS3
NF	96.3 (11.4)	3.6 (-10.8)	3.6 (-10.8)	3.6 (-10.8)	3.6 (-10.8)	3.6 (-10.8)	3.6 (-10.8)	3.6 (-10.8)	3.6 (-10.8)	0.3 (-2.2)	0.3 (-2.2)	0.3 (-2.2)
PAS	31.4 (30.9)	66.8 (-29.7)	66.8 (-29.7)	66.8 (-29.7)	66.8 (-29.7)	66.8 (-29.7)	66.8 (-29.7)	66.8 (-29.7)	66.8 (-29.7)	11.2 (0.3)	11.2 (0.3)	11.2 (0.3)
PBS	66.9 (33.9)	32.7 (-33.1)	32.7 (-33.1)	32.7 (-33.1)	32.7 (-33.1)	32.7 (-33.1)	32.7 (-33.1)	32.7 (-33.1)	32.7 (-33.1)	10.4 (-4.4)	10.4 (-4.4)	10.4 (-4.4)
RBS1	0.7 (0.7)	99.3 (-0.7)	99.3 (-0.7)	99.3 (-0.7)	99.3 (-0.7)	99.3 (-0.7)	99.3 (-0.7)	99.3 (-0.7)	99.3 (-0.7)	0.6 (0.6)	0.6 (0.6)	0.6 (0.6)
RBS2	39.8 (13.4)	60.1 (-13.3)	60.1 (-13.3)	60.1 (-13.3)	60.1 (-13.3)	60.1 (-13.3)	60.1 (-13.3)	60.1 (-13.3)	60.1 (-13.3)	9.1 (8.1)	9.1 (8.1)	9.1 (8.1)
RBS3	7.5 (1.3)	89.0 (0.3)	89.0 (0.3)	89.0 (0.3)	89.0 (0.3)	89.0 (0.3)	89.0 (0.3)	89.0 (0.3)	89.0 (0.3)	3.8 (3.5)	3.8 (3.5)	3.8 (3.5)
SAS1	77.5 (20.7)	22.5 (-20.7)	22.5 (-20.7)	22.5 (-20.7)	22.5 (-20.7)	22.5 (-20.7)	22.5 (-20.7)	22.5 (-20.7)	22.5 (-20.7)	2.1 (-1.5)	2.1 (-1.5)	2.1 (-1.5)
SAS2	92.5 (27.3)	7.5 (-27.3)	7.5 (-27.3)	7.5 (-27.3)	7.5 (-27.3)	7.5 (-27.3)	7.5 (-27.3)	7.5 (-27.3)	7.5 (-27.3)	1.8 (-3.6)	1.8 (-3.6)	1.8 (-3.6)
SAS3	79.1 (3.3)	20.9 (-3.3)	20.9 (-3.3)	20.9 (-3.3)	20.9 (-3.3)	20.9 (-3.3)	20.9 (-3.3)	20.9 (-3.3)	20.9 (-3.3)	2.8 (0.5)	2.8 (0.5)	2.8 (0.5)
SBS1	37.2 (18.7)	10.4 (-14.8)	10.4 (-14.8)	10.4 (-14.8)	10.4 (-14.8)	10.4 (-14.8)	10.4 (-14.8)	10.4 (-14.8)	10.4 (-14.8)	39.6 (23.7)	39.6 (23.7)	39.6 (23.7)
SBS2	66.9 (6.4)	3.0 (-4.8)	3.0 (-4.8)	3.0 (-4.8)	3.0 (-4.8)	3.0 (-4.8)	3.0 (-4.8)	3.0 (-4.8)	3.0 (-4.8)	29.9 (17.1)	29.9 (17.1)	29.9 (17.1)
SBS3	60.4 (1.1)	18.6 (2.1)	18.6 (2.1)	18.6 (2.1)	18.6 (2.1)	18.6 (2.1)	18.6 (2.1)	18.6 (2.1)	18.6 (2.1)	20.1 (9.9)	20.1 (9.9)	20.1 (9.9)

APPENDIX D
AUTOMOTIVE GASOLINE ENGINE AIR-PATH

This case study is based on the LiU-ICE Industrial Fault Diagnosis Benchmark [40], focusing on the air path of a turbocharged gasoline engine. The air path is a complex subsystem involving components such as the air filter, throttle,

intercooler, intake manifold, and turbocharger. Figure 13 illustrates the schematic of the automotive gasoline engine air-path system. The governing equations used to create the structural model for this case study can be found in [40]. The set of sensors and actuators, summarized in Table IV.
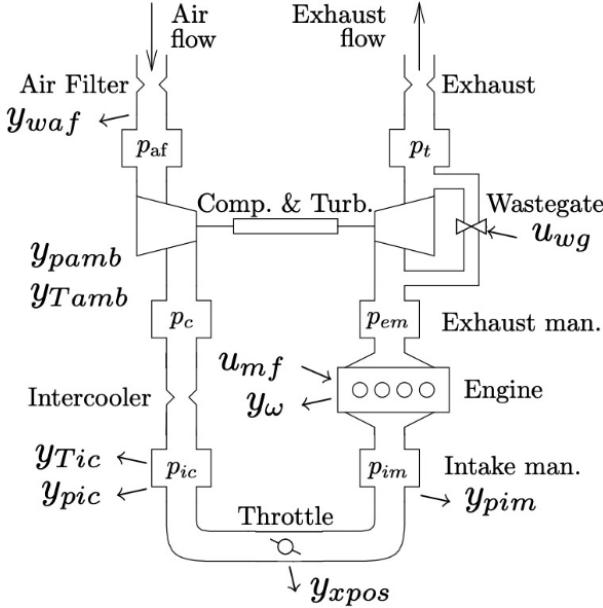

Fig. 13. Schematic of the automotive gasoline engine air-path system [40].

TABLE IV
SENSORS AND ACTUATORS IN THE GASOLINE ENGINE AIR-PATH SYSTEM.

Signal	Description
Sensors	
y_{pic}	Intercooler pressure
y_{Tic}	Intercooler temperature
y_{pim}	Intake manifold pressure
y_{waf}	Mass flow through the air filter
y_{xpos}	Throttle actuator position
y_{ω}	Engine speed
y_{pamb}	Ambient pressure
y_{Tamb}	Ambient temperature
Actuators	
u_{mf}	Requested injected fuel mass
u_{wg}	Requested wastegate actuator position

Data was collected from an engine test bench under various operating conditions, including both nominal and faulty scenarios. The datasets encompass different driving cycles, capturing a wide range of engine behaviors. Each dataset contains approximately 30 minutes of data sampled at 20 Hz. The benchmark includes several fault types, summarized in Table V. Sensor faults are introduced as multiplicative deviations, while the leakage fault is physically induced. Each fault scenario is introduced approximately 120 seconds into the dataset and persists until the end. The evaluation data includes one to two datasets for each fault type, with varying fault magnitudes. Due to the large number of residual candidates for this case, a minimal set of residuals that achieves the highest detection and isolation performance is selected. The
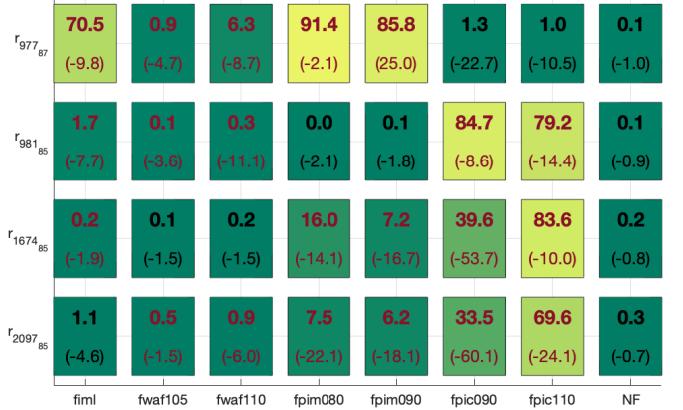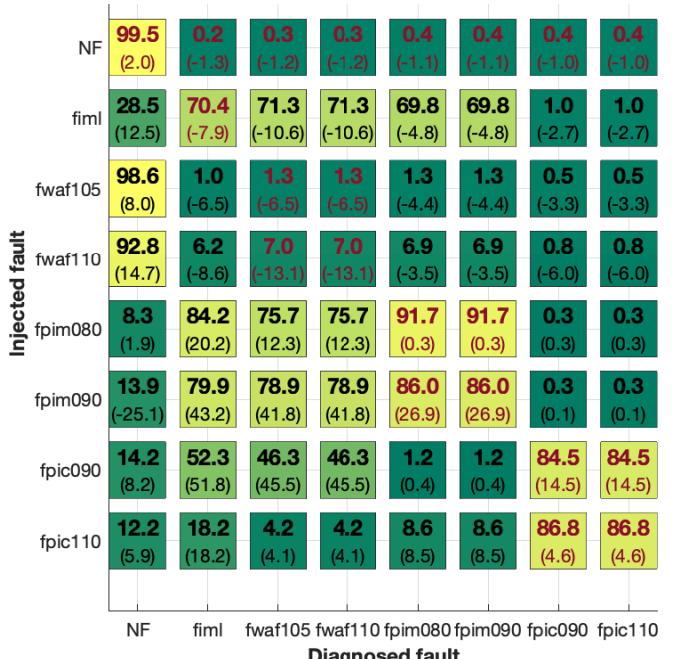

Fig. 14. Engine sensitvity matrix.

Fig. 15. Engine fault isolation performance matrix

illustration of sensitivity and diagnosis results for Engine is illustrated in Figure 14 and Figure 15 respectively.

TABLE V
FAULT SCENARIOS IN THE GASOLINE ENGINE AIR-PATH BENCHMARK.

Component	Fault location
$fpim$	Intake manifold pressure sensor
$fpic$	Intercooler pressure sensor
$fwaf$	Air mass flow sensor
$fiml$	Leakage in the intake manifold

REFERENCES

[1] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki, and J. Schröder, *Diagnosis and fault-tolerant control*. Springer, 2006, vol. 2.

[2] B. Pulido, J. M. Zamarreño, A. Merino, and A. Bregon, "State space neural networks and model-decomposition methods for fault diagnosis of complex industrial systems," *Engineering Applications of Artificial Intelligence*, vol. 79, pp. 67–86, 2019.

[3] S. J. Qin, "Survey on data-driven industrial process monitoring and diagnosis," *Annual reviews in control*, vol. 36, no. 2, pp. 220–234, 2012.

[4] I. Matetić, I. Štajduhar, I. Wolf, and S. Ljubic, "A review of data-driven approaches and techniques for fault detection and diagnosis in hvac systems," *Sensors*, vol. 23, no. 1, 2023.

[5] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," *Machine learning*, vol. 110, no. 3, pp. 457–506, 2021.

[6] A. Mohammadi, M. Krysander, and D. Jung, "Consistency-based diagnosis using data-driven residuals and limited training data," *Control Engineering Practice*, vol. 159, p. 106283, 2025.

[7] B. Lakshminarayanan, "Decision trees and forests: a probabilistic perspective," Ph.D. dissertation, UCL (University College London), 2016.

[8] J. M. Bernardo and A. F. Smith, *Bayesian Theory, volume 405*. John Wiley & Sons, 2009.

[9] D. J. C. Mackay, *Bayesian methods for adaptive models*. California Institute of Technology, 1992.

[10] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118.

[11] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *International conference on machine learning*. PMLR, 2015, pp. 1613–1622.

[12] A. Graves, "Practical variational inference for neural networks," *Advances in neural information processing systems*, vol. 24, 2011.

[13] C. Louizos and M. Welling, "Structured and efficient variational deep learning with matrix gaussian posteriors," in *International conference on machine learning*. PMLR, 2016, pp. 1708–1716.

[14] J. M. Hernández-Lobato and R. P. Adams, "Probabilistic backpropagation for scalable learning of bayesian neural networks," in *ICML*, 2015.

[15] L. Hasenclever, S. Webb, T. Lienart, S. Vollmer, B. Lakshminarayanan, C. Blundell, and Y. W. Teh, "Distributed bayesian learning with stochastic natural gradient expectation propagation and the posterior server," *Journal of Machine Learning Research*, vol. 18, no. 106, pp. 1–37, 2017.

[16] A. Korattikara Balan, V. Rathod, K. P. Murphy, and M. Welling, "Bayesian dark knowledge," *Advances in neural information processing systems*, vol. 28, 2015.

[17] J. T. Springenberg, A. Klein, S. Falkner, and F. Hutter, "Bayesian optimization with robust bayesian neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[18] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.

[19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[20] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[21] T. P. Minka, "Bayesian model averaging is not model combination," *Available electronically at http://www. stat. cmu. edu/minka/papers/bma. html*, pp. 1–2, 2000.

[22] B. Clarke, "Comparing bayes model averaging and stacking when model approximation error cannot be ignored," *Journal of Machine Learning Research*, vol. 4, no. Oct, pp. 683–712, 2003.

[23] J. Lindqvist, A. Olmin, F. Lindsten, and L. Svensson, "A general framework for ensemble distribution distillation," in *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2020, pp. 1–6.

[24] J. Gertler and D. Singer, "A new structural framework for parity equation-based failure detection and isolation," *Automatica*, vol. 26, no. 2, pp. 381–388, 1990.

[25] J. Gertler, *Fault detection and diagnosis in engineering systems*. CRC press, 2017.

[26] J. De Kleer and B. C. Williams, "Diagnosing multiple faults," *Artificial intelligence*, vol. 32, no. 1, pp. 97–130, 1987.

[27] J. De Kleer, A. K. Mackworth, and R. Reiter, "Characterizing diagnoses and systems," *Artificial intelligence*, vol. 56, no. 2-3, pp. 197–222, 1992.

[28] W. Hamscher, L. Console, and J. De Kleer, *Readings in model-based diagnosis*. Morgan Kaufmann Publishers Inc., 1992.

[29] L. Travé-Massuyès, "Bridging control and artificial intelligence theories for diagnosis: A survey," *Engineering Applications of Artificial Intelligence*, vol. 27, pp. 1–16, 2014.

[30] M. Krysander, "Design and analysis of diagnosis systems using structural methods," Ph.D. dissertation, Institutionen för systemteknik, Linköping University, Sweden, 2006.

[31] X. Pucel, W. Mayer, and M. Stumptner, "Diagnosability analysis without fault models," in *20th international workshop on principles of diagnosis (dx-09)*, 2009, pp. 67–74.

[32] E. Frisk, M. Krysander, and D. Jung, "A toolbox for analysis and design of model based diagnosis systems for large scale models," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 3287–3293, 2017.

[33] D. Jung, "Isolation and localization of unknown faults using neural network-based residuals," *arXiv preprint arXiv:1910.05626*, 2019.

[34] C. Sankavaram, A. Kodali, K. R. Pattipati, and S. Singh, "Incremental classifiers for data-driven fault diagnosis applied to automotive systems," *IEEE access*, vol. 3, pp. 407–419, 2015.

[35] E. Frisk, A. Bregon, J. Aslund, M. Krysander, B. Pulido, and G. Biswas, "Diagnosability analysis considering causal interpretations for differential constraints," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 42, no. 5, pp. 1216–1229, 2012.

[36] M. Krysander, J. Åslund, and M. Nyberg, "An efficient algorithm for finding minimal overconstrained subsystems for model-based diagnosis," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 38, no. 1, pp. 197–206, 2007.

[37] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *Advances in neural information processing systems*, vol. 30, 2017.

[38] A. Malinin, B. Mlodozeniec, and M. Gales, "Ensemble distribution distillation," 2019.

[39] A. Mohammadi, T. Westny, D. Jung, and M. Krysander, "Analysis of numerical integration in rnn-based residuals for fault diagnosis of dynamic systems," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 2909–2914, 2023, 22nd IFAC World Congress.

[40] D. Jung, E. Frisk, and M. Krysander, "The LiU-ICE benchmark - an industrial fault diagnosis case study," *arXiv preprint arXiv:2408.13269*, 2024.

[41] E. Frisk and M. Krysander, "Residual selection for consistency based diagnosis using machine learning models," *IFAC-PapersOnLine*, vol. 51, no. 24, pp. 139–146, 2018, 10th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2018.

[42] J. Zheng, J. Li, C. Liu, J. Wang, J. Li, and H. Liu, "Anomaly detection for high-dimensional space using deep hypersphere fused with probability approach," *Complex & Intelligent Systems*, vol. 8, no. 5, pp. 4205–4220, 2022.

[43] D. Jung, M. Krysander, and A. Mohammadi, "Fault diagnosis using data-driven residuals for anomaly classification with incomplete training data," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 2903–2908, 2023, 22nd IFAC World Congress.