IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 11 December 2023, accepted 25 December 2023, date of publication 29 December 2023,
date of current version 5 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3348270

APPLIED RESEARCH

Adaptive Information Fusion Using Evidence Theory and Uncertainty Quantification

FERNANDO ARÉVALO[1], (Senior Member, IEEE), M. P. CHRISTIAN ALISON, M. TAHASANUL IBRAHIM[2], (Member, IEEE), AND ANDREAS SCHWUNG[2]

[1]Faculty of Electrical Engineering and Information Technology, Ruhr-Universität Bochum, 44801 Bochum, Germany
[2]Department of Automation Technology and Learning Systems, South Westphalia University of Applied Sciences, 59494 Soest, Germany

Corresponding author: Fernando Arévalo (Fernando.ArevaloNavas@ruhr-uni-bochum.de)

This work was supported by the Open Access Publication Fund of South Westphalia University of Applied Sciences.

ABSTRACT Fault detection systems support the operator, providing insight during the decision-making while having an (unknown) fault. Data-based models are a common option for a detection system. However, systems that rely purely on data-based models are normally trained with a specific set of data, which cannot necessarily prevent data drift. Thus, an anomaly or unknown condition detection mechanism is required to handle data with new fault cases. Besides, the model's capability to adapt to the unknown condition is equally important to anomaly detection—in other words, its capability to update itself automatically. Alternatively, expert-centered models are powered by the knowledge of operators, which provides the models with production context and expert domain knowledge. The challenge lies in combining both systems and which framework can be used to achieve this fusion. We propose a novel adaptive information fusion methodology to define fault detection systems using evidence theory and uncertainty quantification. The main contribution of this paper is providing a general framework for the fusion of n number of information sources using the evidence theory. The fusion provides a more robust prediction and an associated uncertainty that can be used to assess the prediction likeliness. Moreover, we provide a methodology for the information fusion of two primary sources: an ensemble classifier based on machine data and an expert-centered model. We demonstrate the information fusion approach using data from an industrial setup, which rounds up the application part of this research. Furthermore, we address the problem of data drift by proposing a methodology to update the data-based models using an evidence theory approach. We validate the approach using the Benchmark Tennessee Eastman while doing an ablation study of the model update parameters.

INDEX TERMS Data drift, ensemble classification, knowledge model, model update, information fusion, Dempster-Shafer evidence theory, fault detection system, anomaly detection.

I. INTRODUCTION

Fault detection systems accompany the operators during the machinery operation by early identification of (unknown) faults. Fault detection systems are a key component of decision assistance systems because they provide insights into the machine's condition [1], [2], [3]. Decision assistance systems use the results of fault detection systems in order to provide recommendations to handle faults or improve the machine's performance. Due to their high performance, data-based models are a popular choice when selecting a detection system with reported applications in medicine [4], industry [1], [3], road infrastructure [5], and agriculture [2]. Usually, the data-based models are trained using a specific dataset presenting good results. However, not all data-based models can handle new upcoming faults in the data. Hence, an anomaly detection system must have a mechanism to recognize an upcoming anomaly and the capability to learn upcoming data that differs from the original training data. Equally important is the system's capability to adapt or retrain the data-based models automatically. The retraining or automatic update of the models must consider a minimum size of training data that assures that the models capture the essential patterns to be learned.

The associate editor coordinating the review of this manuscript and approving it for publication was Yifan Zhou.

TABLE 1. List of symbols, abbreviations, and acronyms.

Symbol	Description
DSET	Dempster-Shafer evidence theory
DSRC	Dempster-Shafer rule of combination
YRC	Yager rule of combination
EC	Ensemble classifier
ET	Evidence theory
UQ	Uncertainty quantification
FDS	Fault detection system
ECET	Ensemble classification using evidence theory
KLAFATE	Knowledge transfer framework using evidence theory
INFUSION	Adaptive information fusion using ET and UQ
m	Mass function
U	Uncertainty
w	confidence weight
p	prediction
D^{Tr}	Training data
D^{Va}	Validation data
D^{Te}	Testing data
k	sensitivity to zero factor
F_D	Fusion using DSET rule of combination
F_Y	Fusion using Yager rule of combination
Ws	Window size
Th	Threshold size
Pt	Detection patience

Systems composed by the combination or fusion of several individual models often present better results and robustness than individual models (e.g., bagging and boosting). Though data-based models attain high performance, alternatively, expert-centered knowledge-based models provide versatile features, which are production context and expert domain knowledge. The challenge here lies in how to combine a data-based model and a knowledge-based model. Thus, a common framework is required to perform a fusion of both systems. Such a framework must provide not only a way to combine the models' outputs but to quantify the uncertainty. The uncertainty provides information regarding how reliable the combined system output is.

We propose a novel adaptive information fusion methodology for fault detection systems using evidence theory and uncertainty quantification. The novelty of this paper is that it presents a common framework that allows the fusion of several information sources on the decision level using evidence theory. Besides, we quantify the uncertainty of the system output to provide a better assessment of system output reliability. An essential contribution of this paper is the ability of the data-based model to handle unknown fault cases in the data, which allows the model to update the models automatically.

The individual contributions of this paper are:

- A methodology for the automatic model update of ECs, while feeding up data with unknown fault cases. The methodology includes an uncertainty monitoring strategy that improves the anomaly detection of the EC, stores the data of the unknown condition, and retrains the pool of classifiers of the EC. We present the parameters of the automatic update module: threshold size, window size, and detection patience. The automatic update methodology is rounded up with experiments using the benchmark dataset Tennessee Eastman. The EC is tested using different fault class scenarios, in which we test the impact of a window during anomaly detection. Moreover, we present a detailed analysis of the automatic update parameters regarding retrained EC performance.
- A general framework to combine n number of information sources on the decision level to generate a robust system prediction. The framework uses the Dempster-Shafer evidence theory. Besides, the framework quantifies the uncertainty of the prediction, which can be used to assess the reliability of the system prediction.
- A methodology to combine a multiclass EC with an expert-centered knowledge-based model, in which we apply the general framework of the information fusion. The system architecture shows the components of each model, namely, the inference model and model update module. The application of the information fusion system is tested with the data of an industrial setup using a small-scaled bulk good system. The performance of the individual models (EC and knowledge-based) is compared with the combined system.

This paper is structured as follows: Section II presents a literature survey on the main topics of this paper. The theoretical background is described in section III. Our proposed approach is detailed in section IV. Section IV-C and section IV-D present the methodology for information fusion and model update, respectively. Section V portrays a use case for retraining the EC using the benchmark Tennessee Eastman. Whereas section VI presents a use case for information fusion using the data of a bulk good system laboratory plant. Finally, section VII summarizes the conclusions and future work.

II. RELATED WORK

This section reviews the literature on information fusion, updates of data-based models, and fault detection systems and decision assistance systems.

A. FAULT DETECTION AND DECISION ASSISTANCE SYSTEMS

Assistance systems provide valuable information for the users in the shopfloor. Assistance systems are divided into cognitive (e.g., providing information to the operator) and physical assistance systems (e.g., supporting the operators through exoskeletons) [6]. Moreover, cognitive assistance systems can be classified as decision (e.g., providing information that eases the decision-making process) and perceptual assistance systems (e.g., providing information for specific tasks) [6]. Research contributions using cognitive assistance systems are reported in manual assembly of products [7], [8] and maintenance of wind turbines [6]. This paper focuses on decision assistance systems, which can range from recommendation systems [9], [10], interactive systems [11],

and digital assistance systems [8]. Architectures of decision assistance systems commonly contemplate the components: data collection, a fault detection system, a knowledge base, and an (interactive) user interface [12]. In a previous work [13], we address these components within a knowledge transfer framework for a decision assistance system. The knowledge base plays a crucial role in decision assistance systems because it provides the information that supports the user when a (faulty) condition is active [12]. There are different ways to build a knowledge base, namely using ontologies [12], [14], [15], knowledge graphs [11], [16], failure mode and effects analysis (FMEA) and engineering knowledge [13], [17], and knowledge-based frameworks for multi-modal and multi-structured data [18], [19], [20], [21]. Notable examples of knowledge-based frameworks can be found in industrial applications for (prescriptive) maintenance [22], [23], machine condition assessment [13], providing work instructions [24], life cycle engineering [18], product lifecycle management [19], [25], and cyber-physical production systems [26], [27]. The fault detection system (FDS) is vital to identify the current state of the machinery or process. The fault detection system is usually powered either by a data-based model [13] or a knowledge-centered model [12]. Industrial applications of FDS are reported in the detection of safety equipment [1], fault diagnosis in machines [3], [28], and prescriptive maintenance [23]. Current literature contributions propose decision assistance system architectures that address most of the components and even present methodologies to combine several (data-based) FDS at the decision level [29], [30]. However, no contribution addresses the fusion of n number of FDS within a general framework regardless of the model type. Moreover, there are approaches to quantify the uncertainty of data-based models [28], [31] and knowledge-based models [13], [32], respectively. However, no contribution addresses the uncertainty quantification within a general framework regardless of the model type. This research differentiates from the state-of-the-art, in which we propose a general fusion framework to combine n number of fault detection systems at the decision level using evidence theory. Moreover, we provide a methodology to quantify the uncertainty of the individual fault detection systems, as well as the uncertainty of the resulting system. For this purpose, we present an improved architecture of the decision assistance system presented in [13]. We provide a detailed description of the architecture regarding components and their relationships, focusing on the fusion of fault detection systems and the role of uncertainty.

B. INFORMATION FUSION

Information fusion is a popular approach to combining several sources of information because the combined system often yields better performance and robustness. Information fusion on the decision level is a common practice using data-based models (e.g., supervised classifiers in the case of bagging) [33]. The use of information fusion and data-based models is reported in [29] and [30], in which evidence theory combines models at the decision level. Information fusion using evidence theory provides an additional feature: the uncertainty quantification [13]. The uncertainty serves to assess the output reliability of the combined system [34]. Alternatively, knowledge-based models are expert-centered approaches containing valuable expert domain and environment context [35]. In the case of knowledge-based approaches, information fusion has been applied in the combination of expert knowledge at the decision level [36], [37], [38], and quantifying the uncertainty of expert knowledge [13], [32]. Though combining the strength of data-based and knowledge-based models might be considered a logical step to follow, finding a common framework to perform the fusion is challenging. Besides, knowledge-based models often have a low number of input features in comparison with data-based models. The last aspect requires special attention while performing an inference of the primary systems before performing an information fusion. Current research methodologies cover the information fusion of data-based models [29], [30]. However, existing literature does not report the fusion of data-based and knowledge-based models, though the heterogeneity of the sources could improve the overall result. We propose a methodology for the information fusion of a data-based model with an expert-centered model, in which we use the Dempster-Shafer evidence theory as a general framework for the fusion. Besides, we test the feasibility of the methodology using data from an industrial setup.

C. UPDATE OF DATA-BASED MODELS

The ability of data-based models to handle data with unknown fault cases has grown interest in the research community [39], [40]. A primary step is identifying the unknown fault case or anomaly from the upcoming data. There are different approaches reported in the literature to detect anomalies, which propose the use of evidence theory [28] and unsupervised learning [41], [42]. After identifying the anomaly from the data, the next step is updating the model. In this sense, some methodologies are focus on concept drift detection [43], [44], incremental learning [45], [46], emerging classes or labels [47], [48] [49], and incremental class [48]. Thus, detecting an anomaly is followed by an update or retraining of the data-based model. However, there are challenges associated with the retraining or updating of models: the size of the training data sufficient to capture the essence of the upcoming fault. An essential factor to consider is the performance evaluation of the retrained models. A careful study of the parameters is required because only some upcoming faults might be handled with the same set of retraining parameters. Existing literature addresses the anomaly detection [28], [41], [42], and even the identification of emerging classes (or unknown conditions) [47], [48], [49]. However, the model update using uncertainty remains unexplored. To this end, we propose a

methodology for updating data-based models using DSET, in which we monitor the uncertainty of the fusion to trigger a model update. We focus on the model update of data-based models, specifically for ensemble classification using evidence theory. Besides, we perform an ablation study of the retraining parameters while showing their impact on the model performance. We demonstrate the robustness of the model update using the benchmark Tennessee Eastman.

III. THEORETICAL BACKGROUND

This section presents the basic theory for performing information fusion and the transformation of model predictions using an evidential treatment. The equations of this section are applied during the development of the sections IV-C and IV-D.

A. EVIDENCE THEORY

Dempster-Shafer [50] defined a frame of discernment $\Theta = \{A, B\}$ for the focal elements A and B. The power set 2^{Θ} is defined by $2^{\Theta} = \{\phi, \{A\}, \{B\}, \Theta\}\}$. The definition of a basic probability assignment (BPA) is given by: m: $2^{\Theta} \rightarrow [0, 1]$, in which the BPA must comply with $m(\phi) = 0$, and $\sum_{A\subseteq\Theta} m(A) = 1$. The last equation represents the sum of BPAs. The focal elements of Θ are mutually exclusive: $A \cap B = \phi$.

The *Dempster-Shafer rule of combination* (DSRC) defines how to perform the fusion of two mass functions (e.g., sources of information) using the equation:

$$\begin{aligned} m_{DS}(A) &= (m_1 \oplus m_2)(A) \\ &= \frac{\sum_{B\cap C=A\neq\phi} m_1(B)m_2(C)}{1 - \sum_{B\cap C=\phi} m_1(B)m_2(C)} \end{aligned} \tag{1}$$

where $m_{DS}(A)$ is the fusion of the mass functions m_1 and m_2. The conflicting evidence b_k is defined by:

$$b_k = \sum_{B\cap C=\phi} m_1(B)m_2(C) \tag{2}$$

It is important to remark that, while using DSRC, the conflicting evidence is distributed by each focal element.

Yager [51] defined an alternative rule of combination, which in contrast to DSRC, assigns the conflicting evidence to the focal element Θ. The *Yager rule of combination* (YRC) is defined by the equation:

$$m_Y(A) = \sum_{B\cap C\neq\phi} m_1(B)m_2(C) \tag{3}$$

where $m_Y(A)$ is the fusion of the mass functions $m_1(B)$ and $m_1(C)$. The focal element Θ of the mass function $m_Y(A)$ is defined by: $m_Y(\theta) = q(\theta) + q(\phi)$, where $q(\phi)$ represents the conflicting evidence. Likewise DSRC, the conflicting evidence $q(\phi)$ is represented by:

$$q(\phi) = \sum_{B\bigcap C=\phi} m_1(B)m_2(C) \tag{4}$$

In the case of multiple fusion operations, the mass functions are combined using the following equation:

$$m(A) = \Big(\big(m_1 \oplus m_2\big) \ldots \oplus m_N\Big)(A) \tag{5}$$

where $m(A)$ is the fusion of the n mass functions, and $N \in \mathbb{N}$.

B. EVIDENTIAL TREATMENT OF MODEL PREDICTIONS

We consider models with a common frame of discernment $\Theta = \{L_1, L_2, \ldots, L_N\}$, where N represents the number of labels or classes,, and $N \in \mathbb{N}$. The power set is represented by $2^{\Theta} = \{\phi, \{L_1\}, \{L_2\}, \{L_1, L_2\}$. The last term represents the overall uncertainty U. Each model (e.g., classifier or a rule-based system) provides a prediction in the form of a unique label $p = L_1$ or as an array, $p = [L_1, L_2, \ldots, L_n]$. In section III-A, the sum of BPAs is defined as $\sum_{A\subseteq\Theta} m(A) = 1$. In [13], we proposed a strategy to transform a prediction into a mass function. This operation plays an essential role in the fusion of different information sources. We presented a sum of BPA that considers the weights of each focal element w_m, and the quantification of the overall uncertainty U: $S_{wbpa} = \sum_{j=1}^{N} m_j \cdot w_{m_j} + U = 1$, where $n \in \mathbb{N}$ and w_m is the weight of the evidence m. The following conditions must be fulfilled: $\forall m_j.\ \ m_j > 0$ and $w_{m_j} \rightarrow [0, 1]$. The overall uncertainty is defined as $U = 1 - \sum_{j=1}^{N} m_j \cdot w_{m_j}$, in which a high value of U represents a high uncertainty on the body of evidence (e.g., lack of evidence). We consider that the focal elements are mutually exclusive, which means that only one label is active at the time, which transforms S_{wbpa} into $S_{wbpa} = m_{R_j} \cdot w_{m_{R_j}} + U = 1$. However, we adapted the *sensitivity to zero* approach of Cheng et al. [52], using the equation [36]: $k = 1 - 10^{-F}$, where $k \in \mathbb{R}$, $F \in \mathbb{N}$, and $F \gg 1$. Thus, we transform S_{wbpa} into:

$$S_{awbpa} = \sum_{j=1}^{N} m'_{p_j} \cdot w_{m_{p_j}} + U = 1 \tag{6}$$

where m'_{p_j} represents the j_{th} focal element, and is defined using:

$$m'_{p_j} = \begin{cases} k & \text{if } p_j = True \\ \dfrac{1-k}{N-1} & \text{otherwise} \end{cases} \tag{7}$$

where k is the approximation factor, N is the number of focal elements of Θ, and $N \in \mathbb{N}$. The active prediction p can be transformed into a mass function m using: $\mathbf{m} = \mathbf{m'_p}.\mathbf{w_p}$. The mass function can be represented as a row vector using the following equation:

$$m = [m'_{p_1} \cdot w_{p_1} \quad \ldots \quad m'_{p_N} \cdot w_{p_N} \quad U] \tag{8}$$

and the uncertainty U is defined as:

$$U = 1 - \sum_{j=1}^{N} m'_{p_j} \cdot w_{m_{p_j}} \quad U] \tag{9}$$

IV. INFUSION: ADAPTIVE INFORMATION FUSION USING EVIDENCE THEORY AND UNCERTAINTY QUANTIFICATION

This research proposes an adaptive INformation FUsion approach using evidence theory and uncertainty quantification (INFUSION). This section covers the topics: theoretical background, prediction systems, information fusion, model update of the prediction system, and the decision assistance system.

As a first insight into this theme, we present a general system overview as seen in Fig. 1.

The general system is conformed by n systems used as information sources. The motivation behind this is the creation of a more robust system. The general system overview is composed of the blocks:

- The batch data is the numerical representation of the physical behavior of a machine. The data is split in three categories: training data D^{Tr}, validation data D^{Va}, and testing data D^{Te}. The data is used during the training and inference processes of the models.
- The modules form the decision assessment system:
  - n Systems, in which each system has a model and a model update module. For instance, model 1 has two outputs: the model prediction $\hat{y}_{Sys_1}$ and its associated uncertainty U_{Sys_1}.
  - The fusion module, which combines the predictions $\hat{y}_{Sys_1}..\hat{y}_{Sys_n}$ of the information sources *model 1..model n* into the ensemble prediction $\hat{y}^E_{Sys}$.
  - The model update module is triggered either by each system uncertainty (e.g., U_{Sys_1}) or by the ensemble uncertainty U^E_{Sys}.
  - The assessment module matches each ensemble prediction with its corresponding assessment.
  - The knowledge base has the assessment for each ensemble prediction.
- The assessment is presented to the user (operator) through a user interface.

A primary motivation of this paper is the integration of data-based and knowledge-based models because the combined outcome profits from the strengths of both models. Therefore, the n systems of Fig. 1 are transformed into two major systems: an ensemble classifier (EC) that groups different data-based models and a knowledge-based model. Section IV-A details both systems.

A. PREDICTION SYSTEMS

As presented in Fig. 1, a (prediction) system is conformed by an inference model and an update module. The trained model represents the physical system and is used to predict the system's answer while feeding data to it. The inference model can be data-based (e.g., a supervised classifier), an ensemble classifier (EC) formed by several models, a model built on equations representing the physical system, an ontology, or a knowledge-based model. The model update module adapts the system when the initial conditions have changed (or unknown events occur). The update is performed automatically or manually, depending on the module strategy.

A model M_i is trained using a training dataset D^{Tr} (in the case of data-based models), or is modeled using the relationships between the process variables and thresholds (in the case of a knowledge-based model). A training dataset D^{Tr} contains $N_{o_{Tr}}$ number of observations, $N_{f_{Tr}}$ number of features, and $N_{c_{Tr}}$ number of classes. A frame of discernment Θ is formed by all the labels (or classes) that the model can predict: $\Theta = \{C_1, \ldots, C_N\}$, where $N \in \mathbb{N}$.

Thus, a model M_i outputs the prediction $\hat{y}_i$ while feeding the testing data D^{Te}:

$$\hat{y}_i = M_i(D^{Te}) \tag{10}$$

where $\hat{y}_i \in \Theta$. The prediction $\hat{y}_i$ is transformed into the mass function m_i using equations (6)-(9):

$$m_i = f_m(\hat{y}_i, w_{M_i}) \tag{11}$$

where w_{M_i} represents the (confidence) weights for each class predicted by the model M_i.

We focus this research on a prediction system using EC and rule-based knowledge models. Previous research deepened in these two topics separately [13], [28]. Fig. 2 details the INFUSION system, where the prediction systems are adjusted to a data-based and knowledge-based model. Thus, the data-based model is represented by the EC using the ensemble classification and evidence theory (ECET) approach [28], and the knowledge-based model is built using the knowledge transfer framework and evidence theory (KLAFATE) methodology [13]. It is important to remark that each system has an inference model and a model update module. It is important to note that ECET is an EC formed by n systems, specifically the n supervised classifiers. ECET presents a similar structure from Fig. 1 for the system's prediction, except for the model update module.

The model update module of KLAFATE is manual because it relies on the expertise of the team expert. The methodology is explained in detail in [13]. The automatic model update module of ECET is introduced in this research and is explored in detail in section IV-D. The main blocks of this module are:

- The pool of classifiers and the list of hyperparameters reported in [28].
- The (re)-training pool of classifiers module, which is formed by the blocks:
  - train model using either the prior training data D^{Tr}, or using the re-training data $D^{Tr'}$.
  - model validation either the prior validation data D^{Va}, or the new validation data $D^{Va'}$.
  - uncertainty quantification
- The anomaly detection module which monitors the ensemble uncertainty U_E and the anomaly prediction $\hat{y}_{AN}$ of ECET, and the system uncertainty U_{Sys} and the system prediction $\hat{y}_{Sys}$.

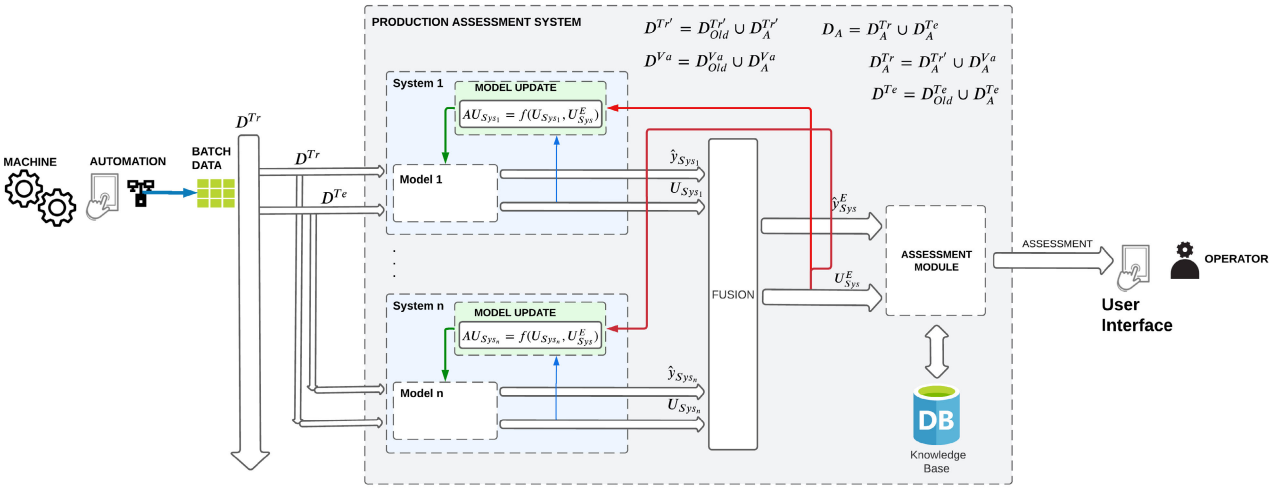


FIGURE 1. General system overview.

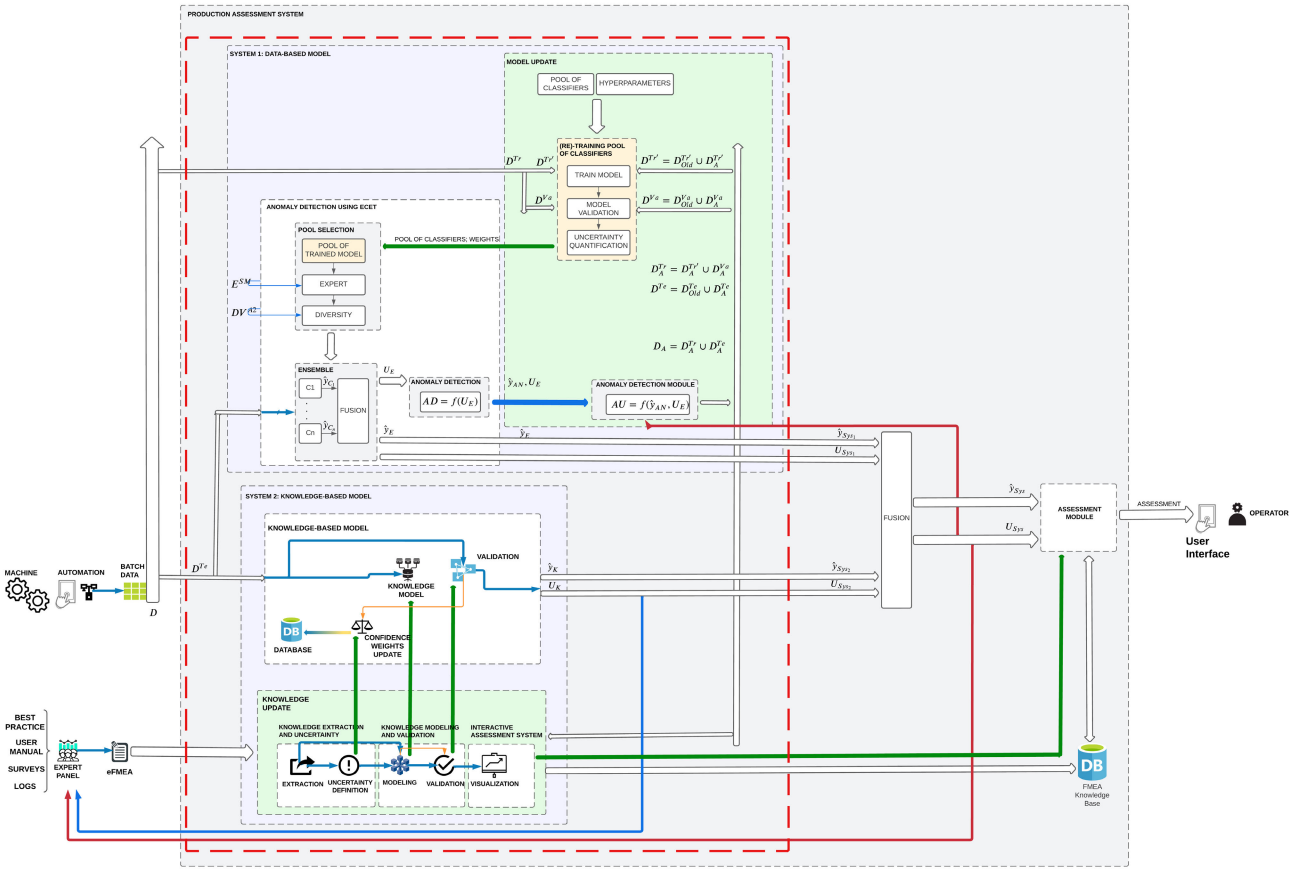


FIGURE 2. INFUSION overview.

1) ECET PREDICTION SYSTEM

In [28], we presented an approach of *ensemble classification using evidence theory* (ECET), in which we propose the use of information fusion to combine the predictions of N number of classifiers. In this paper, we extend the contribution of [28] by formalizing the approach theoretically. This theoretical formalization plays a crucial role in section IV-C and section IV-D, which correspond to the methodologies of information fusion and model update, respectively. Thus, given a n number of classifiers, each classifier produces an output $\hat{y}_i$ using equation (10), where $\hat{y}_i \in \Theta$. The output is subsequently transformed into a mass function m_i using equations (6)-(8). The ensemble classifier (EC) is obtained by combining all the classifiers, specifically using the DSRC on

the mass function of each classifier prediction. As described in equation (5), the DSRC can be used for multiple fusion operations. However, the fusion is performed in pairs. For instance, in the case of three classifiers, the fusion of m_1 (corresponding to the output $\hat{y}_1$ of model C_1) and m_2 is performed first, the result of this fusion $m_1 \oplus m_2$ is then combined with m_3. The fusion of the pair of mass functions m_i and $m_{D_{i-1}}$ is represented using:

$$F_{D_i} = \begin{cases} m_i \oplus m_{D_{i-1}} & \text{if } i > 1 \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where $i \in \mathbb{N}$, m_i is the mass function of the current classifier, and $m_{D_{i-1}}$ is the fusion of the previous mass functions. After obtaining the resulting fusion F_{D_i}, the previous mass function $m_{D_{i-1}}$ is updated:

$$m_{D_{i-1}} = \begin{cases} F_{D_i} & \text{if } i > 1 \\ m_i & \text{otherwise} \end{cases} \tag{13}$$

where $i \in \mathbb{N}$. The last element of the fusion F_{D_i}, which is a row vector, corresponds to the uncertainty U_{D_i}:

$$U_{D_i} = \begin{cases} F_{D_i}[N] & \text{if } i > 1 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

, where N is the cardinality of the frame of discernment Θ, and $N \in \mathbb{N}$. After performing the last fusion, the system prediction $\hat{y}_{EN}$ is calculated using:

$$\hat{y}_{EC} = \arg\max_{\Theta} F_{D_i} \tag{15}$$

where $\hat{y}_{EC} \in \Theta$. The system uncertainty is calculated using: $U_D = U_{D_i}$. A similar procedure is performed when using the YRC to calculate the fusion F_{Y_i}, the previous mass function $m_{D_{i-1}}$, and the uncertainty U_{Y_i}. It is important to remark that the current mass function m_i is used for DSRC and YRC.

2) KLAFATE PREDICTION SYSTEM

In [13] we presented a knowledge-based model using the *knowledge transfer framework using evidence theory* (KLAFATE) [13]. The knowledge was extracted from a failure mode and effects analysis (FMEA) and modeled in rules. Thus, a knowledge rule R_i is defined as the function: $R_i = f(V_1, \ldots, V_{N_V}, T_1, \ldots, T_{N_T})$, where V_1 represents a process variable, T_1 is a threshold or limit value of the process value, N_V is the number of process variables, N_T is the number of thresholds, N_V and $N_T \in \mathbb{N}$. The knowledge rules are mutually exclusive: $R_i \cap R_{i+1} = \phi$. The knowledge model is represented as a set of rules [13]:

$$L_{T_R} = \begin{cases} L_{T_{R_1}} & \text{if} \quad R_1 \\ \ldots & \\ L_{T_{R_m}} & \text{if} \quad R_m \\ L_{T_{R_{m+1}}} & \text{otherwise} \end{cases} \tag{16}$$

where $L_{T_{R_i}}$ represents the approximated rule R_i, m is the number of knowledge rules, $m \in \mathbb{N}$, and L_{T_R}, $R_i \in \Theta$. The active rule is obtained using equations (6)-(9):

$$L_{T_{R_i}} = \begin{cases} k & \text{if } R_i = \text{True} \\ \dfrac{1-k}{N-1} & \text{otherwise} \end{cases}$$

where k is the approximation factor, N is the cardinality of Θ, $k \in \mathbb{R}$, and $N \in \mathbb{N}$. Thus, the mass function is defined using equation (8):

$$m = [L_{T_{R_1}} \cdot w_{R_1} \quad \ldots \quad L_{T_{R_N}} \cdot w_{R_N} \quad U] \tag{17}$$

where w_{R_1} is the (confidence) weight of the rule R_1, and U is the overall uncertainty. The uncertainty U is calculated using the equation (9):

$$U = 1 - \sum_{j=1}^{N} L_{T_{R_j}} \cdot w_{R_j} \tag{18}$$

The (confidence) weight w_{R_j} is defined using the equation [13]:

$$w_{R_j} = \frac{1}{N_R} \sum_{i=1}^{N_R} w_{R_{C_i}}(V, T)$$

The mass function m_{R_i} is transformed into the prediction $\hat{y}_{KE}$ using:

$$\hat{y}_{KE} = \arg\max_{\Theta} m_{R_j} \tag{19}$$

where $\hat{y}_{KE} \in \Theta$.

B. DECISION ASSISTANCE SYSTEM

The decision assistance system provides an interactive source of assessment for the user while receiving the process data. It provides the current status of the system (e.g., system prediction and uncertainty), the assessment (e.g., troubleshooting through the FMEA knowledge base) in the case of a fault case, and notifies in case of an unknown condition for the consequent model update.

The knowledge of the FMEA is stored as a knowledge tuple TU_i [13]:

$$TU_i = (P, SP, FM, C, E, \mathbf{RE}, R, w_R) \tag{20}$$

where FM represents a failure mode, P is a process, SP a subprocess, $\mathbf{C}$ a set of causes, $\mathbf{E}$ a set of effects, $\mathbf{RE}$ a set of recommendations, and $i \in \mathbb{N}$. A set of recommendation is also represented as: $\mathbf{RE} = [RE_1, \ldots, RE_{N_{RE}}]$, where $N_{RE} \in \mathbb{N}$. The latest representation applies to the sets of effects and causes.

In the assessment context, the rule R corresponds to the system prediction $\hat{y}_{Sys}$, and the confidence weight w_R to the system weight $w_{\hat{y}_{Sys}}$, where $R, \hat{y}_{Sys} \in \Theta_{Sys}$, and $w_{Sys} = 1$. It is important to remark, that each system prediction $\hat{y}_{Sys}$ is linked to a knowledge tuple TUi, a failure mode FM, and to a weight w_{Sys}: $\hat{y}_{Sys} \iff TU_i$, $\hat{y}_{Sys} \iff FM$, and $\hat{y}_{Sys} \iff w_{\hat{y}_{Sys}}$. In contrast, a system prediction $\hat{y}_{Sys}$ can be associated to a

set of causes **C**, effects **E**, and recommendations **RE**. The assessment module is modeled through a matching function that associates a system prediction $\hat{y}_{Sys}$ to the rest of the knowledge of the tuple TU_i:

$$P, SP, FM, C, E, \mathbf{RE} = f_{Ma}(\hat{y}_{Sys}, TU) \tag{21}$$

where $i \in \mathbb{N}$. The matching function f_{Ma} provides the assessment while feeding the system prediction $\hat{y}_{Sys}$, specifically returning the troubleshooting information associated with the failure mode: the process P, the subprocess SP, the set of causes **C**, the set of effects **E**, and the set of recommendations **RE**. The decision assistance system was described in detail in a previous work [13].

C. INFORMATION FUSION

Information fusion has a growing research interest because it improves robustness while combining different models. To this end, we propose a novel framework for combining n number of models using DSET. Moreover, this framework is used for the fusion of a data-based model and a knowledge-based model.

Thus, as presented in Fig. 1, the system is formed by n number of subsystems. The system mass function m_{Sys} is obtained after applying the information fusion to all subsystems:

$$m_{Sys}(A) = \left(\left(m_{Sys_1} \oplus m_{Sys_2}\right) \ldots \oplus m_{Sys_n}\right)(A) \tag{22}$$

where $n \in \mathbb{N}$, and $m_{Sys}(A) \in \Theta_{Sys}$. The system mass function m_{Sys} is also referred as F_{Sys}. It is important to remark that all the systems share the same frame of discernment: $\Theta_{KE} = \Theta_{EC} = \Theta_{Sys}$, and

$$\Theta_{Sys} = \{C_1, \ldots, C_{N_{Sys}}\} \tag{23}$$

where C_1 represents the first class (or fault case), N_{Sys} is the number of classes (or fault cases), and $N_{Sys} \in \mathbb{N}$.

The equation (22) can also be represented as:

$$m_{Sys}(A) = \begin{cases} (\bigoplus_{i}^{N_{Sys}} m_{Sys_i})(A) & \text{if } i > 1 \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

where $i, N_{Sys} \in \mathbb{N}$.

This paper adapts the system to two main subsystems: a data-based model M_{EC} and a knowledge-based model M_{KE}.

As a first step we obtain the outputs $y_{\hat{EC}}$ and $y_{\hat{EC}}$ by feeding data to the models M_{KE} and M_{EC}:

$$y_{\hat{EC}} = M_{EC}(D^{Te}) \tag{25}$$

and

$$y_{\hat{KE}} = M_{KE}(D^{Te}) \tag{26}$$

where D^{Te} is the testing data.

The predictions $y_{\hat{EC}}$ and $y_{\hat{KE}}$ are transformed into the mass functions m_{EC} and m_{KE} respectively, using equations (6)-(9):

$$m_{EC} = f_m(y_{\hat{EC}}, w_{M_i}) \tag{27}$$

and

$$m_{KE} = f_m(y_{\hat{KE}}, w_{M_i}) \tag{28}$$

where $w_{M_i} = 1 \; \forall i$, and $i \in \mathbb{N}$.

The next step is to obtain the system fusion F_{Sys} by applying either DSRC or YRC.

Thus, the system fusion $F_{D_{Sys}}$ is calculated using DSRC and applying the equations (1), (2), (22), (24):

$$\begin{aligned} F_{D_{Sys}}(A) &= (\bigoplus_{i}^{N_{Sys}} m_{Sys_i})(A) \\ &= (m_{Sys_1} \oplus m_{Sys_2})(A) \\ &= (m_{EC} \oplus m_{KE})(A) \end{aligned} \tag{29}$$

Likewise, the system fusion $F_{Y_{Sys}}$ is calculated using YRC and applying the equations (3), (4), (22), (24):

$$F_{Y_{Sys}} = (m_{EC} \oplus m_{KE})(A) \tag{30}$$

. The system uncertainty U_D is calculated using the last DSRC fusion F_{D_i}: $\hat{y}_{Sys}$ using:

$$U_{D_i} = F_{D_i}[|\Theta_{Sys}|] \tag{31}$$

where $F_{D_i}[|\Theta_{Sys}|]$ corresponds to the overall uncertainty of the system fusion F_{D_i}. Likewise, the system uncertainty U_Y is calculated using the last YRC fusion F_{Y_i}:

$$U_{Y_i} = F_{Y_i}[|\Theta_{Sys}|] \tag{32}$$

where $F_{Y_i}[|\Theta_{Sys}|]$ corresponds to the overall uncertainty of the system fusion F_{Y_i}.

The last step is the calculation of the system mass function m_{Sys} and the system uncertainties using DSRC U_D and YRC U_Y. The system mass function m_{Sys} is obtained from the last DSRC system fusion $F_{D_{Sys}}$: $m_{Sys} = F_{D_i}$. The mass function m_{Sys}, then, is transformed into the prediction $\hat{y}_{Sys}$ using:

$$\hat{y}_{Sys} = \arg\max_{\Theta} m_{Sys} \tag{33}$$

where $\hat{y}_{Sys} \in \Theta_{Sys}$. Algorithm 1 describes the steps for the information fusion of N_{Sys} number of subsystems while feeding the testing data D^{Te}, where $N_{Sys} \in \mathbb{N}$. Algorithm 1 is an updated version of the algorithm presented in [28].

D. MODEL UPDATE

The anomaly detection functionality is crucial in the model update because it identifies when an unknown condition is present. We present an (automatic) model update for ECET based on uncertainty monitoring. The (manual) model update of KLAFATE was proposed in [13]. The model update is a sequence of five steps: anomaly detection, collection of unknown data, data isolation using a window, retraining, and inference.

Algorithm 1 Information Fusion of N_{Sys} Systems [28]

procedure Information Fusion
  for $j = 1$ to N_{Sys} **do** ▷ N_{Sys} Subsystems
    for $i = 1$ to $N_{D^{Te}}$ **do** ▷ $N_{D^{Te}}$ Samples
      $\hat{y}_i \leftarrow M_j(S_i)$ ▷ by Eq. (25)
      $m_i \leftarrow f_m(\hat{y}_i, w_i^{M_j})$ ▷ by Eq.(6)-(9), (27)
      if i = 1 **then**
        $F_{D_{i-1}} = F_{Y_{i-1}} = 0$
        $m_{D_{i-1}} = m_{Y_{i-1}} = m_i$
        $U_{D_{i-1}} = U_{Y_{i-1}} = 0$
      else
        $F_{D_i} = m_i \oplus m_{D_{i-1}}$ ▷ by Eq. (29)
        $F_{Y_i} = m_i \oplus m_{Y_{i-1}}$ ▷ by Eq.(30)
        $m_{D_{i-1}} = F_{D_i}$
        $m_{Y_{i-1}} = F_{Y_i}$
        $U_{D_i} = F_{D_i}[|\Theta_{Sys}|]$ ▷ by Eq.(31)
        $U_{Y_i} = F_{Y_i}[|\Theta_{Sys}|]$ ▷ by Eq.(32)
  $m_{Sys} = F_{D_i}$
  $\hat{y}_{Sys} = \arg\max_\Theta m_{Sys}$ ▷ by Eq.(33)
  $U_D \leftarrow U_{D_i}$
  $U_Y \leftarrow U_{Y_i}$
  return $\hat{y_{Sys}}$, U_D, U_Y

Algorithm 2 Model Update of ECET

procedure Model Update
  $\hat{y_{EC}} \leftarrow M_{EC}(S_j)$
  $m_{EC} \leftarrow f_m(\hat{y_{EC}}, w^{EC})$ ▷ by Eq.(6)-(9)
  if $C_A = True$ **then** ▷ by Eq. (36)
    $\hat{y}_A = A_K$
    $D_{Temp_j} \leftarrow collect\_data(X_A, \hat{y}_A)$
    $i_A \leftarrow i_A + 1$
    if $C_S = True$ **then** ▷ by Eq. (39)
      $D_A \leftarrow D_{Temp}$
      $D_A^{'Tr}, D_A^{Va}, D_A^{Te} \leftarrow split\_data(D_A)$
      $D^{'Tr} \leftarrow D_{Old}^{'Tr} \cup D_A^{'Tr}$ ▷ by Eq. (43)
      $D^{Va} \leftarrow D_{Old}^{'Tr} \cup D_A^{Va}$ ▷ by Eq. (44)
      $\hat{M}_{Tr} \leftarrow retrain(M, D^{'Tr})$
      $M_{Tr} \leftarrow \hat{M}_{Tr}$ ▷ Replace old models
  else
    $\hat{y}_A = \arg\max_\Theta m_{EC}$ ▷ by Eq.(33)
    $i_A \leftarrow 0$
  return M_{Tr}

1) MODEL UPDATE FOR ECET

Performing ECs are usually the result of a suitable dataset that fits the patterns of the existing data. However, the occurrence of new unknown fault cases might undermine the performance of the ECs, leading to a retraining procedure of the models. To this end, our methodology provides the theoretical basis for updating the data-based models using DSET, in which we monitor the uncertainty of the fusion to trigger a model update. The *model update of ECET* is performed automatically using an anomaly detection strategy, in which the uncertainty is monitored. However, The model update can be set as semi-automatic (e.g., the user receives a notification from executing the model update module) in case the unknown condition needs to be analyzed in detail first. Algorithm 2 describes the sequence of the model update.

We proposed an *anomaly detection* strategy using ECET in [28], in which an unknown condition A_K was detected:

$$\hat{y}_A = \begin{cases} A_K & \text{if } C_A = True \\ \hat{y}_{EC} & \text{otherwise} \end{cases} \tag{34}$$

where $\hat{y}_A$ is a parallel prediction to the EC prediction $\hat{y}_{EC}$, $A_K \in \mathbb{Z}$, and $K \in \mathbb{N}$. The condition for anomalies C_A is defined as:

$$C_A = (U_D > Tr_{D_{Mx}}) \text{ and } (U_Y > Tr_{Y_{Mx}}) \tag{35}$$

where $U_D = b_k$, $U_Y = q(\phi)$, $Tr_{D_{Mx}}$ represents the maximum threshold for U_D, $Tr_{Y_{Mx}}$ is the maximum threshold for U_Y. The terms b_k and $q(\phi)$ are calculated using the equations (1)-(2), and (3)-(4), respectively.

In this paper, we propose the monitoring of the EC uncertainties $U_{D_{EC}}$ and $Y_{D_{EC}}$, as well as the system uncertainties $U_{D_{Sys}}$ and $Y_{D_{Sys}}$. The condition for anomalies from equation (35) is transformed into:

$$C_A = C_{A_{EC}} \; or \; C_{A_{Sys}} \tag{36}$$

where $C_{A_{EC}}$ and $C_{A_{Sys}}$ represent the condition for anomalies of EC and system, respectively. Thus, the anomaly detection of the system is defined as:

$$\hat{y}_{A_{Sys}} = \begin{cases} A_K & \text{if } C_A = True \\ \hat{y}_{Sys} & \text{otherwise} \end{cases} \tag{37}$$

The *data collection of (unknown) conditions* needs to satisfy the condition C_D:

$$C_D = C_A \; and \; C_S \tag{38}$$

where C_S is the condition that satisfies a minimum number of consecutive data samples. The condition C_S is defined as:

$$C_S = i_A > S_{Mn} \tag{39}$$

where i_A is the number of consecutive data samples, S_{Mn} is the minimum number of consecutive data samples, and $i_A, S_{Mn} \in \mathbb{N}$.

The collected data of the unknown condition D_A has the same features f_{Tr} of the (old) original data D, such as $f_A = f_{Tr}$. In contrast, the number of observations o_A might differ from that of the original data o_{Tr}. Thus, the data D_A is represented by a number of observations N_{o_A}, in which each observation is composed by the features $X_A = f_A$ and the associated label (or class) $\hat{y}_A$.

The data D_A is represented as:

$$X_{A_{S_{Mn} \times N_{f_A}}} \times Y_{A_{S_{Mn} \times 1}} \tag{40}$$

where S_{Mn} is the minimum number of consecutive samples of the unknown condition, N_{f_A} is the number of features, $S_{Mn}, N_{f_A} \in \mathbb{N}$, $X_A \in \mathbb{R}$, and $Y_A \in \mathbb{Z}$.

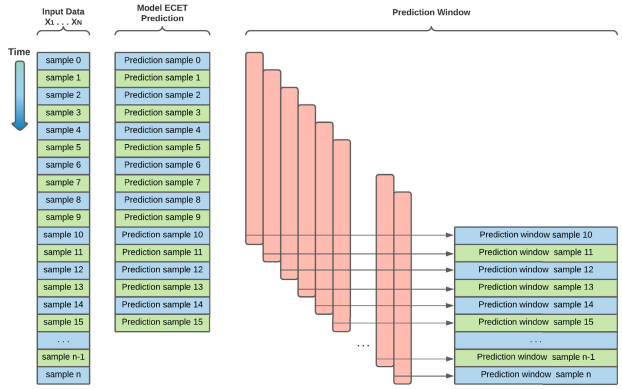


FIGURE 3. EC using a window.

The data D_A is split into training D_A^{Tr} and testing data D_A^{Te}:

$$D_A = \begin{cases} D_A^{Tr} \cup D_A^{Te} & \text{if } C_D = True \\ 0 & \text{otherwise} \end{cases} \tag{41}$$

The training data is split D_A^{Tr} into training data $D_A'^{Tr}$ and validation data D_A^{Va}:

$$D_A^{Tr} = D_A'^{Tr} \cup D_A^{Va} \tag{42}$$

The next step is to integrate the existing data D with the collected data D_A using the following equations:

$$D'^{Tr} = D_{Old}'^{Tr} \cup D_A'^{Tr} \tag{43}$$

$$D^{Va} = D_{Old}^{Va} \cup D_A^{Va} \tag{44}$$

$$D_A^{Te} = D_{Old}^{Te} \cup D_A^{Te} \tag{45}$$

The EC prediction $y_{\hat{EC}}$ usually has not a constant steady value because of the diversity of the classifier's predictions. For this reason, we propose a *window* on the EC prediction $y_{\hat{EC}}$ that can ease the data isolation of the unknown condition. The window smoothes the EC output because it considers a window of N_w number of the last samples for the calculation of the windowed EC output $y_{\hat{EC}}$:

$$\hat{y}_{EC_i}^{w} = \frac{1}{N_w + 1} \sum_{k=i-N_w}^{i} \hat{y}_{EC_k} \tag{46}$$

where $y_{\hat{EC}_i}{}^{w} \in \Theta_{Sys}$.

A graphical representation of the window procedure is exemplified in Fig. 3.

Having the data and the frame of discernment updated, we can proceed with the *retraining of the pool of classifiers*. The retraining is performed using the training methodology presented in [28].

The last step is *to test the EC* using the testing data D^{Te}. For this purpose, we first update the frame of discernment Θ_{Sys}:

$$\Theta_{Sys} = \Theta_{Sys_{Old}} \cup A_K \tag{47}$$

where $\Theta_{Sys_{Old}}$ is the old frame of discernment, A_K is the new focal element, and $N, K \in \mathbb{N}$.

Thus, the updated Θ_{Sys} is transformed into:

$$\Theta_{Sys} = \{F_1, \ldots, F_N, A_K\} \tag{48}$$

2) MODEL UPDATE FOR KLAFATE

Though knowledge-based models contain valuable expert-domain knowledge, the modeling process is time-consuming and requires frequent updates to avoid knowledge obsoleteness. To this end, our methodology provides the theoretical framework for uncertainty monitoring using DSET, which can be used to trigger the update of the knowledge model by the team of experts. The *model update of KLAFATE* is triggered by an uncertainty rise, either on the system or the knowledge model. Thus, the expert team is gathered to analyze the possibility of an unknown condition. Consequently, the expert team recommends adding information sources by including signals, process variables, or hardware to capture new physical signals. The latest purpose is to ease the identification of unknown conditions to create new knowledge rules in the FMEA. Once the expert team analyzes the acquired knowledge, the knowledge rules are validated using key performance indicators (KPI) in the short and long term. The process to create a rule-based system is described in [13].

V. USE CASE: MODEL UPDATE FOR ENSEMBLE CLASSIFICATION USING TENNESSEE EASTMAN DATASET

As described in section IV-D, the approach's novelty is a methodology for updating data-based models while injecting unknown fault cases in the data. The methodology uses primarily an uncertainty monitoring approach based on DSET. This section presents the results of the improved anomaly detection approach and the model update methodology. The robustness of the approaches is tested using the benchmark Tennessee Eastman. We present a description of the dataset. We describe the experiment design explaining the defined scenarios and the performance metrics. The subsection results provide the performance of the experiments. A discussion subsection closes this section by presenting the findings and limitations of the approach. The model update for the data-based model (ECET) and knowledge-based model (KLAFATE) are green highlighted in Fig. 2.

A. DESCRIPTION OF THE TENNESSEE EASTMAN DATASET

The benchmark Tennesse Eastman (TE) was created by Down and Vogel with the motivation to provide an industrial-like dataset based on the Tennesse Eastman chemical plant [53]. The TE chemical plant have five principal process components: condenser, reactor, compressor, separator, and stripper. The dataset is amply used in literature to compare the performance of data-based models. The dataset models a chemical process considering 21 fault cases and a normal operation case. The dataset is divided into training sets and testing sets. The training set consists of 480 rows of data containing 52 features for each fault. In contrast, the training set of the normal condition contains 500 rows of data. The testing set consists of 960 rows of data, in which the first 160 rows belong to the normal condition and the rest 800 rows

belong to the fault case. Given the prediction difficulty, the fault cases are usually grouped into three categories: easy cases (1, 2, 4, 5, 6, 7, 12, 14, 18), medium cases (8, 10, 11, 13, 16, 17, 19, 20) and hard cases (3, 9, 15 and 21) [54]. A detailed dataset description can be found in [53] and [28].

B. EXPERIMENT DESIGN

We followed the procedure proposed in [28], in which we used the benchmark TE to test the performance of the proposed approaches. Besides, we considered a pool of ten classifiers (e.g., five NN-based models and five non-NN-based models) as the basis of the ECs. We considered only experiments using ML-based ECs, and Hybrid ECs (a combination of non-NN-based classifiers and NN-based classifiers). The procedure is documented in detail in [28]. We trained the classifiers of the ECs using the fault cases (0,1,2,6,12) as the basis of the experiments. We defined two experiment scenarios: data isolation using a window and an update of ECs. We develop the approach using the IDE Anaconda and the libraries Scikit-learn and PyTorch [55], [56], [57]. We perform the experiments on a Ubuntu 20.04.3 LTS environment using a CPU i7-7700 @3.60GHz x 8, 32GB RAM, and a GPU NVIDIA GeForce GTX 1660 SUPER.

1) DATA ISOLATION USING A WINDOW

We selected the MC ECs M3 and H5-2 from the previous work [28] with the best performance criteria. The EC M3 consists of non-NN classifiers, whereas the EC H5-2 is hybrid. We compared the results obtained by performing a variation on the window size. The base classifiers' and ECs' hyperparameters are detailed in [28].

2) UPDATE OF ECS

We selected the ML-based ECs M3, M4, and M5 to perform the experiments and comparisons. Given the constraint of limited retraining data, we discard NN-based and Hybrid ECs. The procedure consists of two data batches for each experiment. The first batch contains the known fault cases (0,1,2,6,12) and one anomaly case (e.g., fault case 7). The EC identifies the anomaly through uncertainty monitoring, collects the anomalous data, and retrains the EC if the data is sufficient. We assign the anomaly data with the arbitrary label 30. The second batch contains testing data of the fault cases (0,1,2,6,12) and the anomaly (e.g., fault case 7). For comparison purposes, the original label 7 is changed by the new label 30. We defined three main experiments, namely, the retraining of the ECs using all the fault cases (1,...,21), the study of the retraining parameters (e.g., threshold size, window size, and detection patience) using the fault cases (7,8,15), and the fine-tuned retrained ECs using all the faults (1,...,21). We selected the fault cases (7,8,15) as anomalies to have a case for each primary data group (easy, medium, and hard).

3) PERFORMANCE METRICS

We use the performance metrics F1-score (F1) and fault detection rate (FDR, also known as recall). F1 and FDR are detailed in [58].

C. RESULTS

This subsection presents the experiment results of the model update approach. For this purpose, the experiments are divided into two parts: data isolation using a window and a model update of EC.

1) DATA ISOLATION USING A WINDOW

We perform experiments using different window sizes to study their impact on the EC performance. We compare the effects of using no-window ($w = 0$) and a window ($w = 20$, $w = 50$).

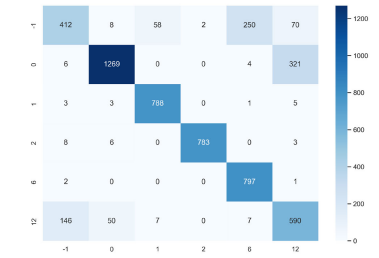
Table 2 presents the F1-scores of the BIN EC M5 and MC EC H5-2. The hyperparameters of the base classifiers and ECs were reported in detail in [28]. The BIN EC M5 presents comparable results while varying the window size with average F1-scores of 0.6%, 0.64%, and 0.65% for the window sizes (0, 20, 50), respectively. In contrast, the MC EC H5-2 presented higher results using a window (20,50) compared to no-window $w = 0$. The MC EC H5-2 presented average F1-scores of 0.63%, 0.81%, and 0.88% for the window sizes (0,20,50), respectively.

Fig. 4 presents the plots of the MC EC H5-2 trained with fault cases (0,1,2,6,12) and using the anomaly fault case (7) while doing a variation on the window size (0, 20, 50). Figures 4a, 4b and 4c show the confusion matrices for the window sizes $w = 0$, $w = 20$, and $w = 50$, respectively. The confusion matrices for the window sizes $w = 20$ and $w = 50$ present better results than the confusion matrix with window size $w = 0$. The predictions plots of figures 4e, and 4f confirm the results of the confusion matrices, in which the predictions (blue) are closer to the ground truth (red) for EC using the window sizes $w = 20$ and $w = 50$. The anomaly case (7) is represented as the label (-1) in the predictions plot. It is important to remark that the approach using a window smooths the EC predictions. Figures 4g, 4h and 4i show the precision-recall curves for the MC EC H5-2 for the window sizes $w = 0$, $w = 20$, and $w = 50$, respectively. The MC EC H5-2 using a window size $w = 50$ presents the highest performance for the known fault cases and the unknown condition.

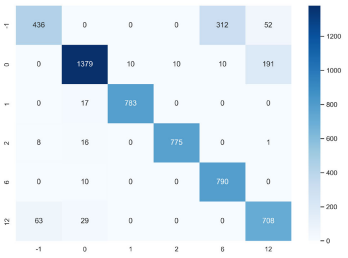
2) MODEL UPDATE OF EC

We perform three different experiments in this subsubsection: the model update of the EC (retraining), the study of the variation of the retraining parameters, and finally, selecting a fine-tuned retrained EC.

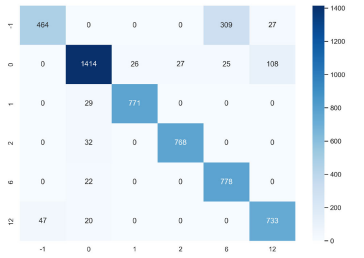
We test the *model update of the EC* using all the fault cases of the TE dataset. For this purpose, we selected the MC ECs M3, M4, and M5. The hyperparameters of the base classifiers and ECs were reported in detail in [28]. Table 3



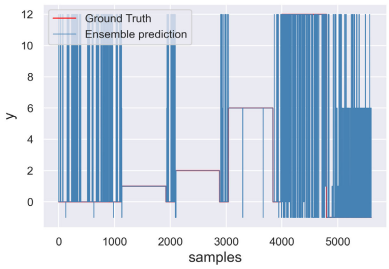
(a) Confusion matrix of EC H5-2 w=0



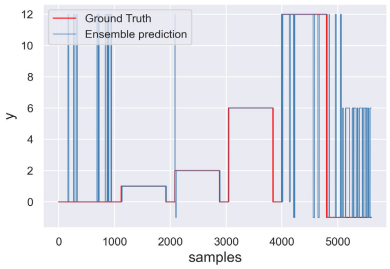
(b) Confusion matrix of EC H5-2 w=20



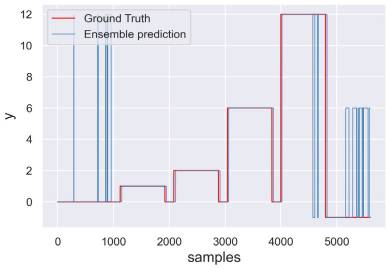
(c) Confusion matrix of EC H5-2 w=50



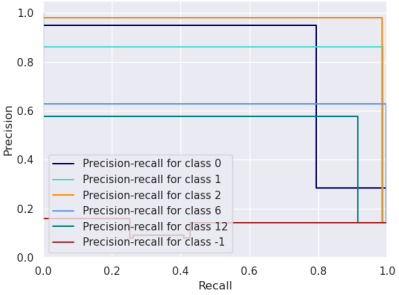
(d) Predictions of EC H5-2 w=0



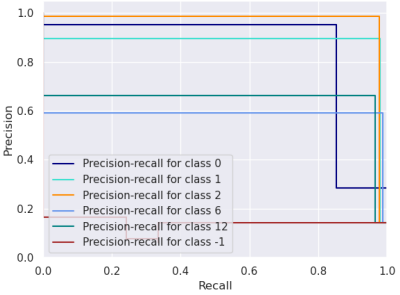
(e) Predictions of EC H5-2 w=20



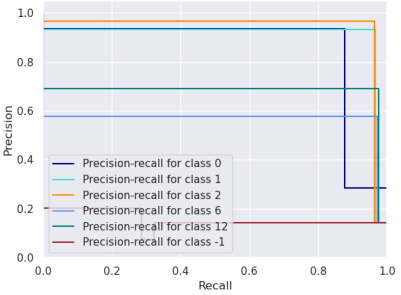
(f) Predictions of EC H5-2 w=50



(g) PR-RE curve of EC H5-2 w=0



(h) PR-RE curve of EC H5-2 w=20



(i) PR-RE curve of EC H5-2 w=50

FIGURE 4. Anomaly detection using different window sizes for the MC EC H5-2 trained with the known cases 0,1,2,6,12, and using the fault case (7) as an anomaly. The confusion matrices of H5-2 are displayed in (a)-(c), the predictions in (d)-(f), and the precision-recall curves in (g)-(i).

presents the F1-scores of the MC ECs M3, M4, and M5 trained with the fault cases (0,1,2,6,12). The MC ECs M3, M4, and M5 present comparable results with an average F1-score of 0.39, 0.36, and 0.37, respectively. The EC MC M3 detected the anomalies (7,17) with F1-scores higher or equal to 0.43 and the anomalies (13,14) with F1-scores higher or equal to 0.33 and less than 0.43. The EC MC M4 detected the anomalies (8,14,17) with F1-scores higher or equal to 0.67 and the anomalies (7,10,11,15) with F1-scores higher or equal to 0.38 and less than 0.54. Alternatively, the EC M5 detected the anomalies (14,18,20) with F1-scores higher or equal to 0.54 and the anomalies (8,17) with F1-scores higher or equal to 0.43 and less than 0.54.

Fig. 5 presents the plots of the MC ECs M3, M4, and M5 trained with fault cases (0,1,2,6,12) and using the anomaly fault 7. Figures 5a, 5b and 5c show the confusion matrices for the ECs M3, M4, and M5, respectively. The confusion matrix of the MC EC M3 presents better results than the confusion matrices of the other ECs. Alternatively, the prediction plots of figures 5d, 5e and 5f present mixed results, in which M3 identifies the anomaly better, but the case (12) is confused with the anomaly. In addition, M5 presents a better prediction of the known fault cases but has a lower anomaly detection. The uncertainty quantification (UQ) using DSET is presented in figures 5g, 5h and 5i for the MC ECs M3, M4, and M5, respectively. The MC EC M5 presents steadier values than the MC ECs M3 and M4, which confirms the prediction pattern. The latest can be enunciated as the lower the uncertainty, the better the classification performance (likeliness). Figures 5j, 5k and 5l show the precision-recall curves for the ECs M3, M4, and M5, respectively. The MC EC M3 presents the highest anomaly detection and generally the highest results for the known cases. In contrast, the MC ECs M4 and M5 present mixed results with lower performance in comparison to the EC MC M3.

The next step is the *study of the retraining parameters*. For this purpose, we test the effects of the threshold size, window size, and detection patience. We chose the MC EC

TABLE 2. Anomaly detection results of selected ensemble multiclass classifiers using all the fault cases, and F1-score.

Fault	BIN EC M5			MC EC H5-2		
	w=0	w=20	w=50	w=0	w=20	w=50
1	0.61	0.70	0.74	0.61	0.79	0.88
2	0.48	0.53	0.62	0.55	0.72	0.72
3	0.50	0.41	0.36	0.63	0.88	0.93
4	0.50	0.44	0.35	0.60	0.87	0.95
5	0.65	0.77	0.80	0.60	0.81	0.89
6	0.91	0.94	0.92	0.42	0.35	0.32
7	0.67	0.74	0.74	0.59	0.67	0.75
8	0.44	0.38	0.25	0.72	0.82	0.93
9	0.58	0.57	0.56	0.64	0.88	0.95
10	0.55	0.61	0.58	0.64	0.87	0.91
11	0.65	0.70	0.76	0.61	0.84	0.94
12	0.57	0.58	0.65	0.62	0.77	0.86
13	0.52	0.55	0.61	0.70	0.81	0.88
14	0.63	0.74	0.76	0.62	0.87	0.94
15	0.59	0.66	0.71	0.64	0.89	0.94
16	0.57	0.62	0.60	0.63	0.83	0.93
17	0.61	0.69	0.69	0.63	0.87	0.95
18	0.80	0.80	0.78	0.85	0.93	0.96
19	0.62	0.68	0.68	0.62	0.88	0.93
20	0.61	0.64	0.61	0.60	0.84	0.89
21	0.63	0.72	0.81	0.62	0.88	0.92
Avg F1-score	0.60	0.64	0.65	0.63	0.81	0.88

TABLE 3. Classification results of the ECs after retraining using all the fault cases, and F1-score. The retraining parameters are threshold size $th = 100$, window size $ws = 20$, and detection patience $pt = 15$.

Fault	RT MC EC (0,1,2,6,12)		
	M3	M4	M5
1	0.98	0.99	0.99
2	0.99	0.99	0.98
3	0.03	0.01	0.00
4	0.30	0.15	0.26
5	0.22	0.07	0.15
6	1.00	1.00	1.00
7	0.71	0.41	0.28
8	0.23	0.46	0.43
9	0.07	0.03	0.10
10	0.25	0.13	0.12
11	0.29	0.15	0.09
12	0.95	0.95	0.95
13	0.36	0.21	0.08
14	0.33	0.66	0.60
15	0.22	0.07	0.00
16	0.03	0.27	0.02
17	0.43	0.41	0.47
18	0.06	0.02	0.78
19	0.30	0.08	0.00
20	0.28	0.55	0.54
21	0.05	0.05	0.00
Avg F1-score	**0.39**	**0.36**	**0.37**

M3 to perform the experiments and selected the threshold sizes (150,250,350) and anomalies (7,8,15).

a: EFFECTS OF THE THRESHOLD SIZE

Table 4 presents the F1-scores of the MC ECs M3, M4, and M5 trained with the fault cases (0,1,2,6,12). The retraining parameters window size and detection patience are fixed with values of $ws = 20$ and $pt = 15$, respectively. The MC EC M3 presented higher results using a threshold size $th = 150$ with an average F1-score of 0.81 for the anomaly (7), compared with the values of 0.57 and 0.50, corresponding to the threshold sizes (250, 250). The MC EC M3 presents comparable results for the anomaly (8) with average F1-scores of 0.81, 0.82, and 0.82 for the threshold sizes (150, 250, 350), respectively. In contrast, the MC EC M3 presented higher results using a threshold size $th = 350$ with an average F1-score of 0.74 for the anomaly 15, in comparison with the values of 0.54 and 0.55, which correspond to the threshold sizes (150, 250), respectively.

Fig. 6 displays the EC M3 performance for each class while effectuating variations on the threshold size (150,250,350) for the anomalies (7,8,15). The best performance corresponds to the anomaly (8), in which the EC M3 detects the fault cases (0,1,2,6,12) often correctly, and it has limited anomaly detection. In contrast, the EC M3 presents a lower performance while applying the anomalies (7,15).

b: EFFECTS OF THE WINDOW SIZE

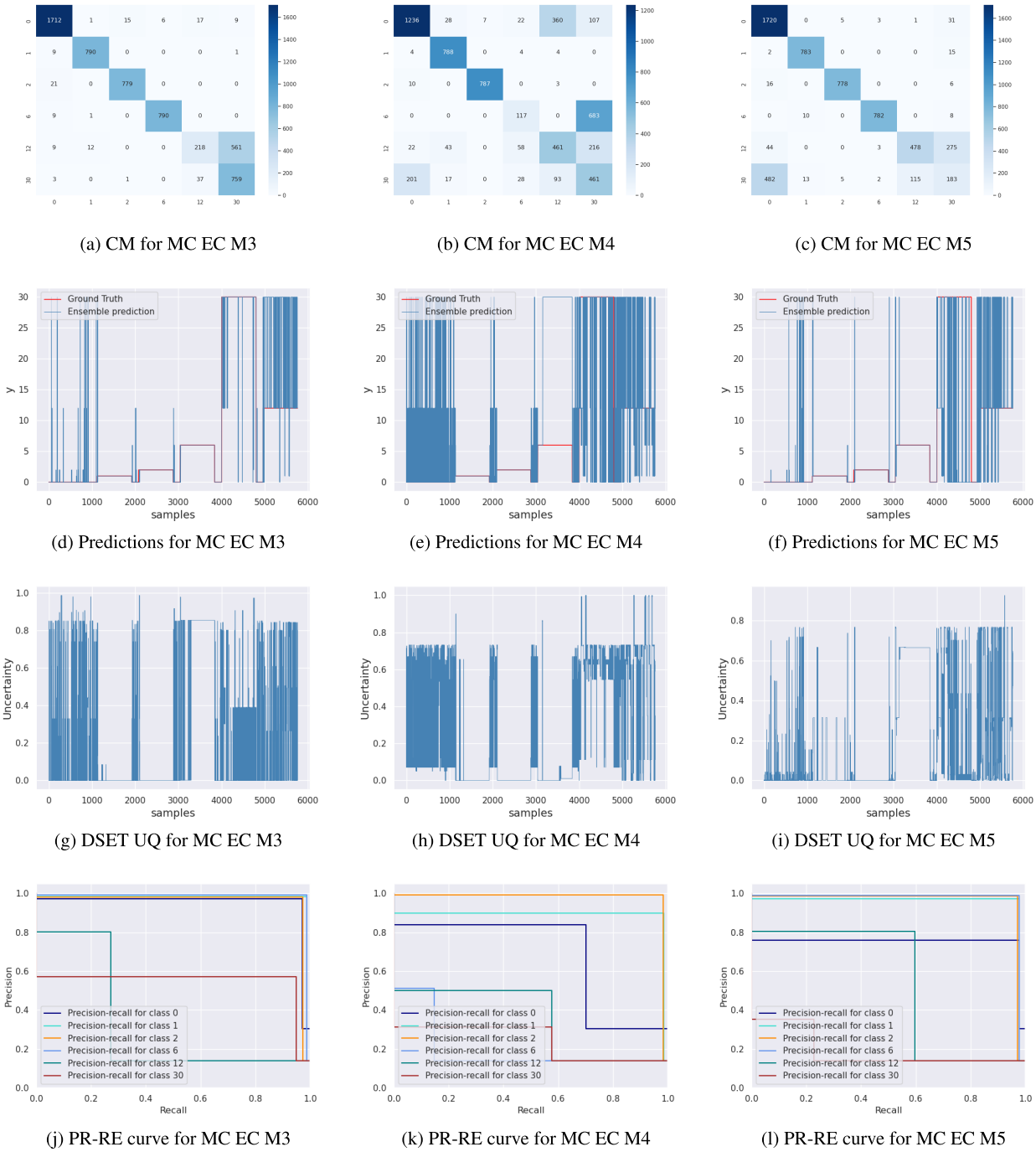
Table 5 presents the F1-scores of the MC ECs M3, M4, and M5 trained with the fault cases (0,1,2,6,12). The retraining parameters threshold size and detection patience are fixed, with values of $th = 250$ and $pt = 15$, respectively. The MC EC M3 presented average F1-scores higher than 0.84 using window size (10,50) for the anomaly (7). Alternatively, the MC EC M3 presented average F1-scores higher than 0.72 for the anomaly (8) using the window size (20,50). In contrast, the MC EC M3 presented higher results using a window size $ws = 50$ with an average F1-score of 0.74 for the anomaly (15), in comparison with the values of 0.50 and 0.55, which correspond to the window sizes (150, 250), respectively.

Fig. 7 displays the EC M3 performance for each class while effectuating variations on the memory size (10,20,50) for the anomalies (7,8,15). The best performance corresponds to the anomaly (8) using a window size $ws = 20$, in which the EC M3 detects the fault cases (0,1,2,6,12) mostly correct, and it has a limited anomaly detection. In contrast, the EC M3 presents a lower performance while applying the anomalies (7,15).

c: EFFECTS OF THE DETECTION PATIENCE

Table 6 presents the F1-scores of the MC ECs M3, M4, and M5 trained with the fault cases (0,1,2,6,12). The retraining parameters threshold size and window size are fixed with values of $th = 250$ and $ws = 20$, respectively. The MC EC M3 presented an average F1-scores of 0.84 using detection patience of $pt = 5$ and $pt = 30$, respectively, compared to the average F1-score of 0.57 for $pt = 15$. In the case of anomaly (8), the MC EC M3 presented higher results using detection patience $pt = 15$ with an average F1-score of 0.82, in comparison with the values of 0.78 and 0.58, which correspond to the detection patience (5,30), respectively. The MC EC M3 presented average F1-scores higher than 0.73 for the detection patience (5,30), while the average F1-score of 0.55 is obtained with the detection patience $pt = 15$.

Fig. 8 displays the EC M3 performance for each class while effectuating variations on the detection patience (5,15,30) for the anomalies (7,8,15). The best performance corresponds to



(a) CM for MC EC M3 (b) CM for MC EC M4 (c) CM for MC EC M5

(d) Predictions for MC EC M3 (e) Predictions for MC EC M4 (f) Predictions for MC EC M5

(g) DSET UQ for MC EC M3 (h) DSET UQ for MC EC M4 (i) DSET UQ for MC EC M5

(j) PR-RE curve for MC EC M3 (k) PR-RE curve for MC EC M4 (l) PR-RE curve for MC EC M5

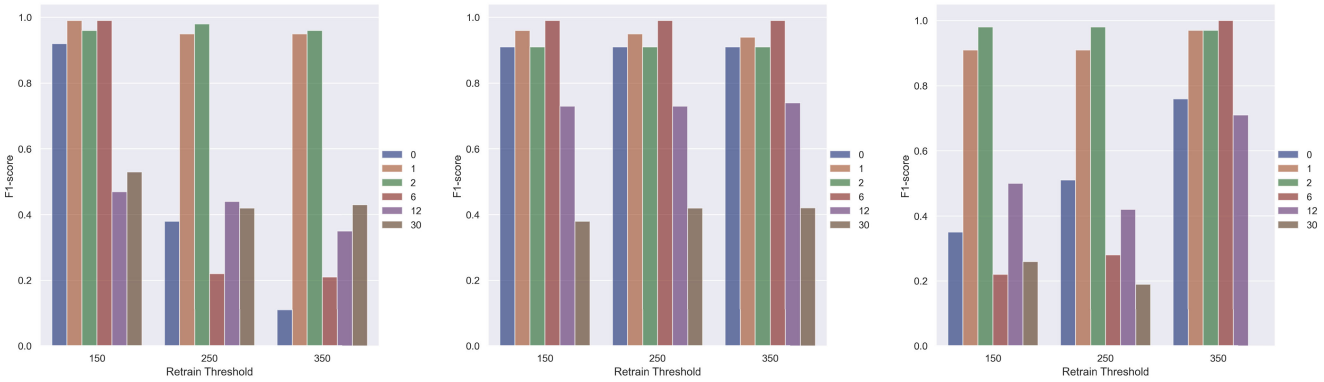
FIGURE 5. Anomaly Detection and UQ results for MC ECs M3, M4 and M5 trained with the fault cases (0,1,2,6,12): Confusion matrices (a)-(c), classification results (d)-(f), DSET UQ (g)-(i), and precision-recall curves (j)-(l) while injecting anomaly 7.

the anomaly (8) using detection patience $pt = 15$, in which the EC M3 detects the fault cases (0,1,2,6,12) mostly correct and has a limited anomaly detection. In contrast, the EC M3 presents a lower performance while applying the anomalies (7,15).

Finally, we present the performance of the ECs with the *tuned retraining parameters*. Table 7 presents the F1-scores of the MC ECs M3, M4, and M5 retrained with the fault cases (0,1,2,6,12) and the respective anomaly. In this case, the anomalies cases are all fault cases except for the original training cases. The retraining dataset contains the original fault cases and the detected data from the anomaly (unknown fault case from the data). The retraining parameters are threshold size $th = 250$, window size $ws = 20$, and detection patience $pt = 15$. The MC ECs M3, M4 and M5 present comparable results with an average F1-score of 0.39, 0.42, and 0.42, respectively. The MC EC M3 detected the anomalies (7,11) with F1-scores higher or equal

TABLE 4. Anomaly detection results of MC EC M3 using the fault cases (0,1,2,6,12), the anomalies (7,8,15), thresholds variations (150,250,350), window size (20), patience (15), and F1-score.

Fault	MC EC M3								
	A7			A8			A15		
	Th=150	Th=250	Th=350	Th=150	Th=250	Th=350	Th=150	Th=250	Th=350
0	0.92	0.38	0.11	0.9	0.91	0.91	0.35	0.51	0.76
1	0.99	0.95	0.95	1	0.95	0.94	0.91	0.91	0.97
2	0.96	0.98	0.96	0.9	0.91	0.91	0.98	0.98	0.97
6	0.99	0.22	0.21	1	0.99	0.99	0.22	0.28	1.00
12	0.47	0.44	0.35	0.7	0.73	0.74	0.5	0.42	0.71
30	0.53	0.42	0.43	0.4	0.42	0.42	0.26	0.19	0.00
Avg F1-score	**0.81**	**0.57**	**0.50**	**0.81**	**0.82**	**0.82**	**0.54**	**0.55**	**0.74**

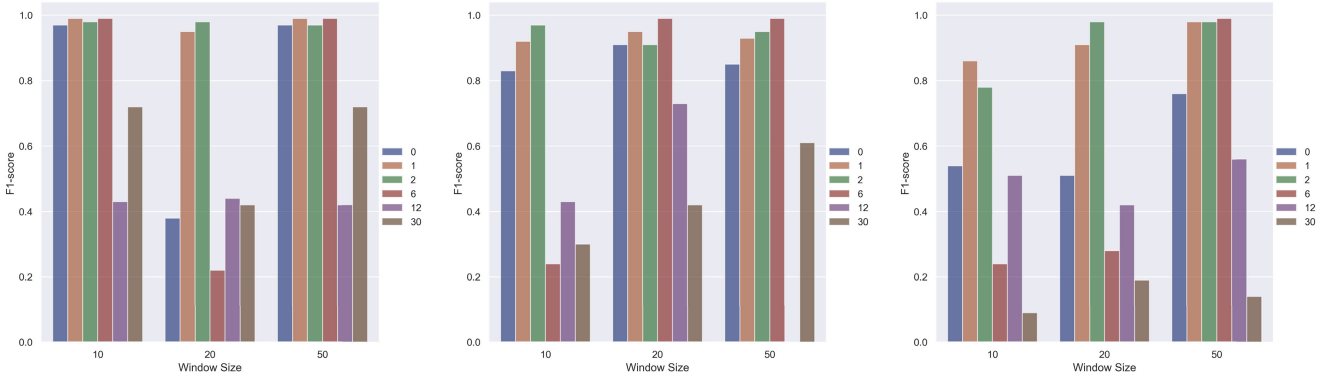


(a) Bar chart for M3 using A7 (b) Bar chart for M3 using A8 (c) Bar chart for M3 using A15

FIGURE 6. F1-score results after retraining for the ECs BIN M4, MC M3, and MC M5: (a)-(c) Bar plots for the known cases (0,1,2,6,12) and the new case (30, corresponding to the injected anomaly 7). The plots represent the ECs results using memory size 20 and patience 15, while varying the threshold (150,250,350).

TABLE 5. Anomaly detection results of MC EC M3 using the fault cases (0,1,2,6,12), the anomalies (7,8,15), window size variations (10,20,50), threshold (250), patience (15), and F1-score.

Fault	MC EC M3								
	A7			A8			A15		
	me=10	me=20	me=50	me=10	me=20	me=50	me=10	me=20	me=50
0	0.97	0.38	0.97	0.8	0.91	0.85	0.54	0.51	0.76
1	0.99	0.95	0.99	0.9	0.95	0.93	0.86	0.91	0.98
2	0.98	0.98	0.97	1	0.91	0.95	0.78	0.98	0.98
6	0.99	0.22	0.99	0.2	0.99	0.99	0.24	0.28	0.99
12	0.43	0.44	0.42	0.4	0.73	0	0.51	0.42	0.56
30	0.72	0.42	0.72	0.3	0.42	0.61	0.09	0.19	0.14
Avg F1-score	**0.85**	**0.57**	**0.84**	**0.62**	**0.82**	**0.72**	**0.50**	**0.55**	**0.74**

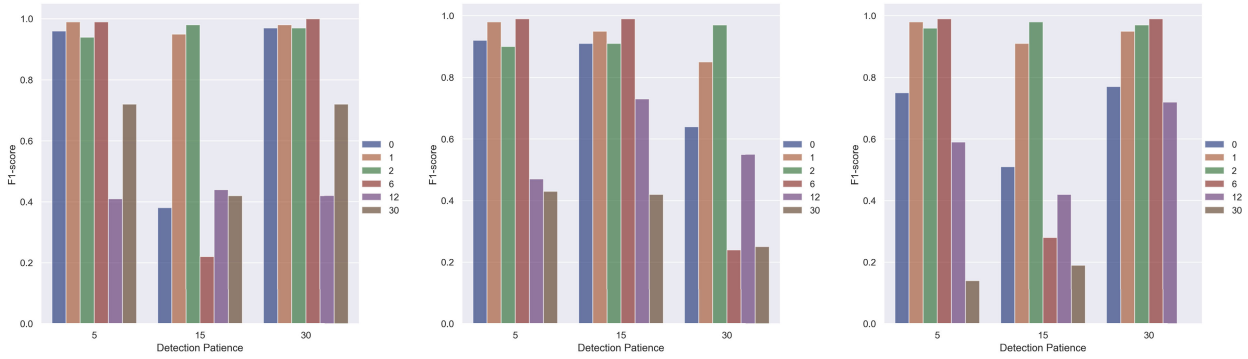


(a) Bar chart for M3 using A7 (b) Bar chart for M3 using A8 (c) Bar chart for M3 using A15

FIGURE 7. F1-score results after retraining for the ECs BIN M4, MC M3, and MC M5: (a)-(c) Bar plots for the known cases (0,1,2,6,12) and the new case (30, corresponding to the injected anomaly 7). The plots represent the ECs results using threshold 250 and patience 15, while varying the memory size (10,20,50).

TABLE 6. Anomaly detection results of MC EC M3 using the fault cases (0,1,2,6,12), the anomalies (7,8,15), patience variations (5,15,30), threshold (250), memory size (20), and F1-score.

Fault	MC EC M3								
	A7			A8			A15		
	pt=5	pt=15	pt=30	pt=5	pt=15	pt=30	pt=5	pt=15	pt=30
0	0.96	0.38	0.97	0.9	0.91	0.64	0.75	0.51	0.77
1	0.99	0.95	0.98	1	0.95	0.85	0.98	0.91	0.95
2	0.94	0.98	0.97	0.9	0.91	0.97	0.96	0.98	0.97
6	0.99	0.22	1	1	0.99	0.24	0.99	0.28	0.99
12	0.41	0.44	0.42	0.5	0.73	0.55	0.59	0.42	0.72
30	0.72	0.42	0.72	0.4	0.42	0.25	0.14	0.19	0
Avg F1-score	**0.84**	**0.57**	**0.84**	**0.78**	**0.82**	**0.58**	**0.74**	**0.55**	**0.73**



(a) Bar chart for M3 using A7 (b) Bar chart for M3 using A8 (c) Bar chart for M3 using A15

FIGURE 8. F1-score results after retraining for the ECs BIN M4, MC M3, and MC M5: (a)-(c) Bar plots for the known cases (0,1,2,6,12) and the new case (30, corresponding to the injected anomaly 7). The plots represent the ECs results using threshold $th = 250$ and window size $me = 20$, while varying the patience (5,15,30).

to 0.55 and the anomalies (9,13,17) with F1-scores higher or equal to 0.34 and less than 0.42. The MC EC M4 detected the anomalies (8,14,17) with F1-scores higher or equal to 0.67 and the anomalies (7,10,11,15) with F1-scores higher or equal to 0.38 and less than 0.54. Alternatively, the EC M5 detected the anomalies (14,18) with F1-scores higher or equal to 0.68 and the anomalies (7,11,15,17,20) with F1-scores higher or equal to 0.31 and less than 0.54.

D. COMPARISON WITH LITERATURE

Though the current approach can automatically update the models while detecting unknown fault cases from the data, the stored data to retrain the models might be insufficient for some fault cases. Thus, the stored data for some fault cases might not capture the essential patterns to identify the condition. In contrast, the contributions of literature presented in the comparison consider all the extent of the testing data.

Table 8 compares the anomaly detection results between the proposed approach and the literature. The multiclass ECs M3, M4, and M5 are originally trained using the fault cases (0,1,2,6,12). The testing data consists of the fault cases (3,9,15,21), which represent unknown conditions to the ECs. For this purpose, each EC is retrained with one fault case at a time. We use the F1-score as a performance metric to compare the proposed approach with other literature contributions. It is essential to mention that the MC EC H5-2 from a previous work [28] uses the full extent of testing data, as well in the case of Top-K DCCA [41]. The results of the ECs M3, M4, and M5 present lower results with average F1-scores of 20.36%, 3.50%, and 2.59%, respectively. The results of H5-2 and Top-K DCCA present general scores of 63.69% and 50.04%, respectively. Only M3 presents a score of 31.07% for the fault case 21, which still lies under the better performance results of H5-2 and Top-K DCCA with scores of 63.1% and 50.05%, respectively.

Table 9 compares the anomaly detection results between our approach and the literature. We use the FDR to compare our results with the literature results. The retrained MC ECs M3, M4, and M5 present lower results with average FDR scores of 53.02%, 41.68%, and 35.04%, respectively. The MC ECs M3 and H3-4 present FDR scores of 87.97% and 73.76%, respectively. The approaches DPCA-DR, AAE, and MOD-PLS have FDR scores of 83.51%, 78.55%, and 83.83%, respectively.

E. DISCUSSION

The ECs improved the anomaly detection capability after implementing the *window size*. In the case of the MC EC M5, the general F1-score improved from 0.6 to 0.65 using a window of $w = 50$ for the latest score. In the case of H5-2, the results are remarkable, in which the general F1-score score improved from 0.63 to 0.88 using a window of $w = 50$ for the latest score. However, a side effect of the window is a delay effect on the ensemble prediction, which is reflected while comparing Fig. 4d and Fig. 4f.

TABLE 7. Classification results of the RT ECs after retraining using all the fault cases, and F1-score. The retraining parameters are threshold size *th* = 250, window size *ws* = 20, and detection patience *pt* = 15.

Fault	RT MC EC (0,1,2,6,12)		
	M3	M4	M5
1	0.98	0.99	0.99
2	0.99	0.99	0.98
3	0.00	0.02	0.01
4	0.00	0.03	0.01
5	0.00	0.18	0.13
6	1.00	1.00	1.00
7	0.72	0.54	0.31
8	0.29	0.71	0.44
9	0.34	0.00	0.00
10	0.27	0.38	0.22
11	0.55	0.51	0.54
12	0.95	0.95	0.95
13	0.35	0.20	0.15
14	0.26	0.77	0.76
15	0.13	0.39	0.31
16	0.26	0.09	0.20
17	0.42	0.67	0.51
18	0.06	0.02	0.68
19	0.20	0.07	0.05
20	0.28	0.23	0.52
21	0.07	0.13	0.01
Avg F1-score	**0.39**	**0.42**	**0.42**

There are remarkable effects on the EC M3 performance while doing variations on the retraining parameters, namely, threshold size, window size, and detection patience. The results are mixed, and the average performance depends on the studied anomaly. However, from the results, it is possible to identify that a *threshold* of $Th = 150$ presented the best average results for anomaly 7. In contrast, a threshold of $Th = 350$ presented the best results for anomaly 8. Alternatively, the plots of Fig. 6 visualize the performance of each class while doing variations on the threshold. The MC EC M3 presents an overall good performance while applying anomaly 8, in which the EC classifies the known cases mostly correctly and has a limited detection of the anomaly. In contrast, the anomaly detection feature decreases the performance of the known fault cases for some fault cases, which is visually represented in Fig 6a while applying anomaly 7. *Variation of the window size* reported favorable average performance results for a window of $me = 50$ while considering all the anomalies (7,8,15). In contrast, the plots of Fig. 7 show that the best results correspond to the window size $me = 20$ while applying anomaly 8, in which the EC classifies known cases properly, and it has a limited detection of the anomaly. Likewise the threshold experiments, a similar effect of decreasing classification performance of the known cases is detected. Generally, a *patience* of $pt = 5$ presented the best average results for all the anomalies (7,8,15). In contrast, the plots of Fig. 8b show that the best results correspond to the patience $pt = 15$ while applying anomaly 8, in which likewise the window size experiment, the EC classifies the known cases mostly correctly, and it has a limited detection of the anomaly. Likewise the threshold and window size experiments, the performance of the EC is affected by some faults while using the anomaly detection approach.

The retrained MC ECs M3, M4, and M5 presented mixed results using the same retraining parameters: threshold size $th = 250$, window size $me = 20$, and patience $pt = 15$. The average F1-score of M3, M4, and M5 presented values of 0.67, 0.44, and 0.42, respectively. For this configuration, M3 presented the best results, however, it is important to remark that the anomalies (14)-(19) are not detected. In contrast, M4 and M5 detected the faults (14,17,18), though the average scores are lower than M3 scores.

The performance of the retrained MC ECs presented mixed results. For instance, the EC M3 detected the anomaly cases (4,5,7,11,13) with FDR scores higher than 77% and the anomalies (10,20,21) with FDR scores higher than 53%. However, the results of the retrained ECs presented a lower performance than other literature contributions. The average FDR scores of M3, M4, and M5 are 50.18%, 43.60%, and 51.44%. It is important to remark that the retrained models only use 250 samples as training data (only 52% of the available data), in which other fault cases might be included as a side effect of the parameter patience.

The proposed approach presented a novel methodology for anomaly detection and automatic update of ECs. The ablation study provided a systematic procedure to test the parameters that influence anomaly detection and, subsequently, the automatic update capability. The current approach differentiates from the state-of-the-art, in which uncertainty quantification and information are utilized as the core modules for anomaly detection and automatic update methodologies.

VI. USE CASE: INFORMATION FUSION OF ENSEMBLE CLASSIFICATION MODEL AND KNOWLEDGE-BASED MODEL USING INFUSION ON A BULK GOOD SYSTEM

As described in section IV-C, the approach's novelty is a methodology for the information fusion of data-based and knowledge-based models. The methodology primarily uses a novel framework for combining n number of models using DSET.

This section presents the results of the information fusion approach and an ablation study considering the different system configurations. The system configurations consist of the detection system using: the data-based model, the knowledge model, or a hybrid model (data-based model together with a knowledge model) using information fusion. We test the approach using a dataset of an industrial setup, namely, a bulk good system laboratory plant. We describe the testbed and the dataset. We present the results and a discussion of the findings. Fig. 2 displays the main blocks of this section: the data-based model (ECET), the knowledge-based model (KLAFATE), and the outer module for the information fusion of both models.

A. DESCRIPTION OF THE BULK GOOD SYSTEM LABORATORY PLANT AND DATASET

The bulk good system (BGS) laboratory plant is an industrial setup used for testing production and fault detection experiments. The BGS consists of four stations that represent

TABLE 8. Classification results of the ECs after retraining using all the fault cases, and F1-score. The retraining parameters are threshold size $th = 250$, window size $ws = 20$, and detection patience $pt = 15$.

Fault	RT MC EC (0,1,2,6,12)			MC EC (0,1,2,6,12) [28]	Top-K DCCA [41]
	M3	M4	M5	H5-2	
3	0.00	7.43	0.00	64.3	53.82
9	28.87	6.21	5.81	63.01	52.31
15	21.49	0.00	0.45	64.35	43.98
21	31.07	0.35	4.09	63.1	50.05
Avg F1-score	**20.36**	**3.50**	**2.59**	**63.69**	**50.04**

TABLE 9. Classification results of the ECs after retraining using all the fault cases, and FDR. The retraining parameters are threshold size $th = 250$, window size $ws = 20$, and detection patience $pt = 15$.

Fault	RT MC EC (0,1,2,6,12)			MC EC (0,1,2,6,12)		DPCA-DR	AAE	MOD-PLS
	M3	M4	M5	M3	H3-4			
1	98.50	98.50	98.50	93.5	82.13	99.6	100	99.88
2	97.75	97.63	97.75	94.75	83.63	98.5	99	98.75
3	0.00	8.50	0.00	91.88	75	2.1	34.88	18.73
4	0.00	3.38	0.00	89.75	76.5	99.8	98.62	99.88
5	0.00	12.75	11.00	90.63	75.25	99.9	55	99.88
6	99.88	99.88	99.88	97.5	64.75	99.9	100	99.88
7	96.25	82.75	23.13	88.25	77.5	99.9	100	99.88
8	47.25	43.63	0.00	87	76.38	98.5	97.88	98.5
9	41.75	4.75	4.25	90.75	74.88	2	33.62	12.11
10	47.00	24.00	31.25	88.13	75.13	95.6	74	91.01
11	45.88	52.38	0.00	92.5	73.63	96.5	82	83.15
12	92.50	91.75	92.00	81.5	80.5	99.8	99.75	99.75
13	35.88	35.50	0.00	84.38	71.38	95.8	96.25	95.38
14	77.63	86.88	100.00	91	71.75	99.8	100	99.88
15	36.88	0.00	0.50	91.25	66.25	38.5	31.25	23.22
16	83.50	0.00	7.25	90.25	76.25	97.6	64.75	94.26
17	0.00	69.00	98.75	91.25	75.75	97.6	96	97
18	62.63	12.63	5.75	37.63	56	90.5	95	91.14
19	20.13	2.88	5.00	92	74.5	97.1	53.87	94.13
20	51.75	48.25	57.63	89.38	72.63	90.8	78.62	91.26
21	78.38	0.25	3.13	94.13	69.25	53.9	59	72.66
Avg F1-score	**53.02**	**41.68**	**35.04**	**87.97**	**73.76**	**83.51**	**78.55**	**83.83**

standard modules of a bulk good handling system on a small scale: loading, storing, filling, and weighing stations. A detailed description of the BGS and applications can be found in [13] and [36]. The stations are built using state-of-the-art hardware regarding industrial controllers, communication protocols, sensors, and actors. The BGS dataset contains 14055 rows of data, each containing 133 features and three classes. The features represent information about sensors, actors, and controllers. The classes represent the different machine conditions, namely, low quality (LQ), low production (LP), and normal production (NP or the normal condition). Each class is associated with a failure mode (fm), which in this case, translates into LQ (fm1), LP (fm2), and NP (fm3). In the case of the class NP, it does not represent a failure mode but is represented in the same framework for consistency purposes of the knowledge model.

B. EXPERIMENT DESIGN

This subsection presents the methodology followed for the ECET and INFUSION experiments using the BGS dataset. Besides, we describe the performance metric used to compare the experiments.

1) ECET USING THE BGS DATA

We followed the same methodology of [28] for the creation of MC ECs using the BGS data, which includes the pool of base classifiers, the grid of hyperparameters of each classifier, and the grid of hyperparameters for each EC. We used the data-based models: decision tree (DTR), K-nearest neighbors (KNN), AdaBoost (ADB), support vector machine (SVM), and naive Bayes (NBY). For this purpose, we first trained the pool of classifiers using only ML models, which implies the search for the proper hyperparameters for each model. The second step is creating the ECs, using the EC hyperparameters. The last step presents the inference results of the ECs while injecting the BGS data.

2) INFUSION USING THE BGS DATA

The knowledge-based model KLAFATE was presented in [13], in which we describe the knowledge rules. We only use the failure modes fm1, fm2, and fm3 for the INFUSION experiments. We present a comparison table using knowledge, data fusion, and knowledge and data fusion models. The KLAFATE model represents the knowledge model. The data fusion models are represented by the ECET ECs models and a fusion of two data-based models. Lastly, the knowledge and data fusion models are represented by the combination of the SVM-KNN-KLAFATE models and the INFUSION models composed of an MC EC and the KLAFATE model.

3) PERFORMANCE METRICS

We use the F1-score as the main performance metric to compare the different experiments. Panda et al. [58] present a detailed description of the F1-score calculation.

TABLE 10. Grid of hyperparameters for base classifiers using the *BGS* dataset and the cases (1,2,3).

Model	Hyperparameters	MC 1,2,3
ADB	learning_rate	0.01
	n_estimators	10
DTR	criterion	entropy
	max_depth	10
KNN	criterion	manhattan
	n_neighbors	7
	weights	distance
NBY	-	NP
SVM	C	1000
	gamma	0.01
	kernel	rbf

C. RESULTS

This subsection presents the results using the BGS data for the ECET and the INFUSION architectures. For this purpose, we present the F1-score results of the models or ECs. Besides, we display the confusion matrix, classification predictions, and uncertainty for the different architectures.

1) ECET USING THE BGS DATA

The first is to train the pool of base classifiers, which we performed using the module grid search of scikit-learn. Table 10 presents the hyperparameters of the base classifiers trained with the cases (1,2,3), which corresponds to the failure modes (fm1, fm2, fm3), respectively.

The next step is applying the ECET methodology to find the most performing MC ECs. We obtained the ML-based MC ECs, shown in Table 11. The hyperparameters expert (Exp), diversity (Div), version of diversity (Ver), and pre-cut (PC) are set to False.

Table 12 presents the F1-scores of the MC ECs M3, M4, and M5 and the base MC classifiers DTR, KNN, and ADB. The MC ECs M3, M4, and M5 present the same average F1-score of 1.00, whereas the base classifiers DTR, KNN, and ADB have values of 1.0, 1.0, and 0.96, respectively.

Fig. 9 presents the plots of MC ECs M3, M4, and M5 trained using the cases (1,2,3), which correspond to the failure modes (fm1, fm2, fm3), respectively. Fig. 9a, 9b, 9c show the confusion matrices for the MC ECs M3, M4, and M5, respectively. The confusion matrices present the same performance for the MC ECs M3, M4, and M5. Fig. 9d, Fig. 9e, Fig. 9f display the predictions in blue color compared with the ground truth in red color for the MC ECs M3, M4, and M5, respectively. Likewise, in the previous case, the prediction plots are identical for the MC ECs M3, M4, and M5. Fig. 9g, Fig. 9h, Fig. 9i present the DSET UQ for MC ECs M3, M4, and M5, respectively. In contrast to the previous plots, the uncertainty is reduced as the ensemble size increases. In the case of the MC EC M5, the model presents the clearest plot, except for the fm3, which has a noisy behavior. Figures 9j, 9k, 9l show the precision-recall curves for the MC ECs M3, M4, and M5, respectively. The precision-recall curves present the same performance for the MC ECs M3, M4, and M5. Similarly to the confusion matrices, the precision-recall curves reflect a high performance of the ECs.

2) INFUSION USING THE BGS DATA

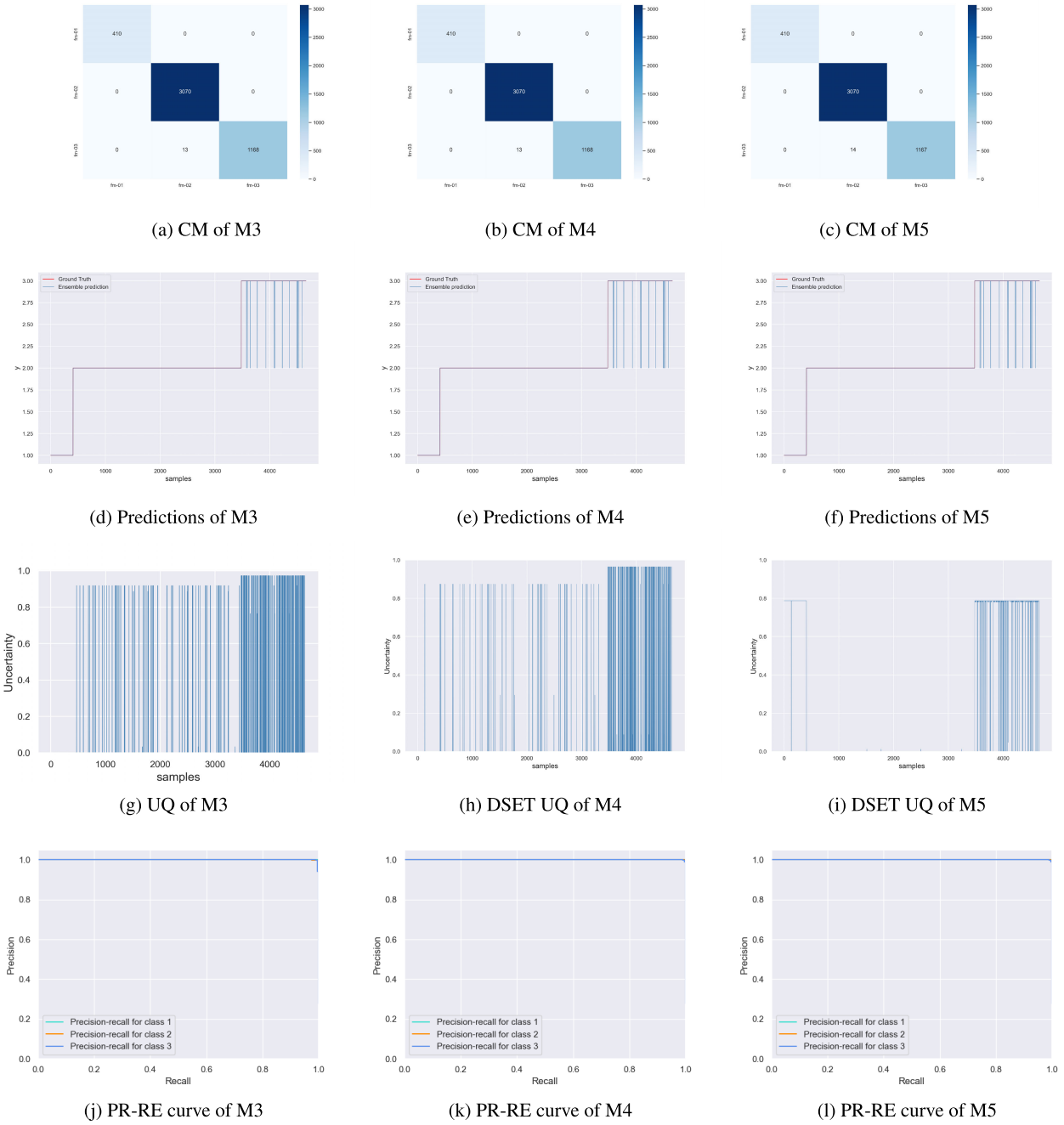
Table 13 presents the F1-scores of the knowledge-based model, the fusion of data-based models, and the fusion of data-based and knowledge-based models. The knowledge-based model is represented by the model using the KLAFATE methodology. The fusion of data-based models is represented by the models using the ECET methodology (M3, M4, M5) and an additional case performing a DSET fusion of the data-based models KNN and SVM (without the ECET methodology). The fusion of data-based models and the knowledge-based model is represented by the models using the INFUSION methodology (IFS3, IFS4, IFS5) and an additional case performing a fusion of the models KNN, SVM, and KLAFATE. The KLAFATE model presents an average F1-score of 0.75, whereas the individual cases (1,2,3) presented values of 0.95, 0.79, and 0.52, respectively. The ECET and INFUSION models (IFS3, IFS4, IFS5) present the best average F1-score with a value of 1.00. The fusion of SVM and KNN presents an average F1-score of 0.96, whereas the fusion of KLAFATE, SVM, and KNN presents an improved average F1-score with a value of 0.98.

Fig. 10 presents the plots of the main models: the KLAFATE knowledge-based model, ECET data-based model (M3), and the INFUSION model (fusion of KLAFATE and ECET). Fig. 10a, 10b, 10c show the confusion matrices for the models KLAFATE, ECET (M3), and INFUSION (IFS3), respectively. The confusion matrices with the best performance correspond to the models ECET and INFUSION. In contrast, KLAFATE presents a poor performance by detecting fm3. Fig. 10d, Fig. 10e, Fig. 10f display the predictions in blue color compared with the ground truth in red color for the models KLAFATE, ECET, and INFUSION, respectively. The clearest plots correspond to the ECET and INFUSION models, whereas the KLAFATE model presents a noisy plot. Fig. 10g, Fig. 10h, Fig. 10i present the DSET UQ for the models KLAFATE, ECET, and INFUSION, respectively. In the case of KLAFATE, the plot presents a continuous line since the expert team can only change the uncertainty's value. In contrast, ECET presents an extremely noisy plot for the fm3. In the case of INFUSION, the plot presents a steadier uncertainty. Figures 10j, 10k, 10l show the precision-recall curves for the models KLAFATE, ECET, and INFUSION, respectively. The precision-recall curves present the same high performance for the models ECET and SYS, whereas the model KLAFATE presents a lower performance (e.g., especially for fault cases 2 and 3).

It is important to remark on the INFUSION robustness, in which we perform the fusion of a high-performing ECET with a low-performing KLAFATE. The low performance of KLAFATE for some fault cases did not affect INFUSION's performance. INFUSION performance presents a steady high performance while examining table 13 and the confusion matrix from Fig. 10c. Alternatively, a detailed examination

TABLE 11. EC hyperparameters using the *BGS* dataset and the cases (1,2,3).

Type	Dataset	EC	Exp	Div	Ver	PC	Pool of base classifiers
MC	1,2,3	M3	False	False	False	False	ADB-DTR-KNN
		M4	False	False	False	False	ADB-DTR-KNN-NBY
		M5	False	False	False	False	ADB-DTR-KNN-NBY-SVM



(a) CM of M3 (b) CM of M4 (c) CM of M5

(d) Predictions of M3 (e) Predictions of M4 (f) Predictions of M5

(g) UQ of M3 (h) DSET UQ of M4 (i) DSET UQ of M5

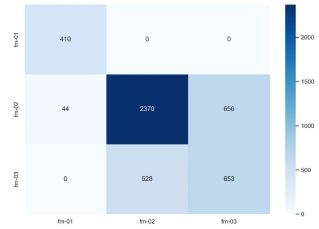
(j) PR-RE curve of M3 (k) PR-RE curve of M4 (l) PR-RE curve of M5

FIGURE 9. Results using different models KLAFATE, ECET (MC EC M3), and SYS using cases (1,2,3): Confusion matrices (a)-(c), classification results (d)-(f), DSET UQ (g)-(i), and precision-recall curves (j)-(l).

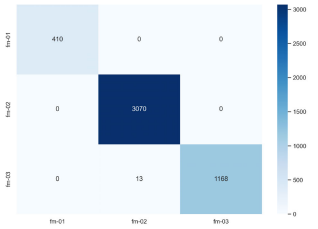
of the uncertainty provides an additional perspective on INFUSION's performance, in which the uncertainty presents areas with high values. Thus, uncertainty monitoring can be used to evaluate ECET and KLAFATE to determine the causes of low performance.

D. DISCUSSION

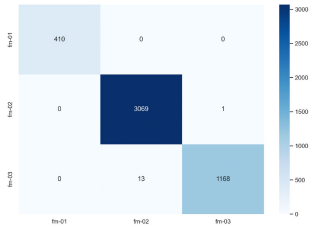
The knowledge-based model KLAFATE presented mixed results, in which some faults are well identified or predicted. However, the strength of this approach relies on how well the rule represents a machine condition. Representing knowledge



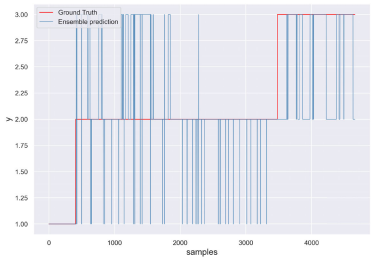
(a) CM of KLAFATE



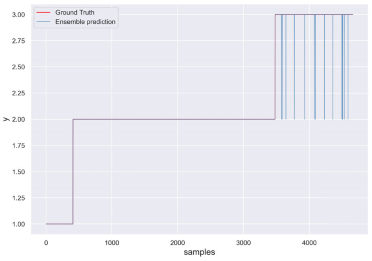
(b) CM of ECET



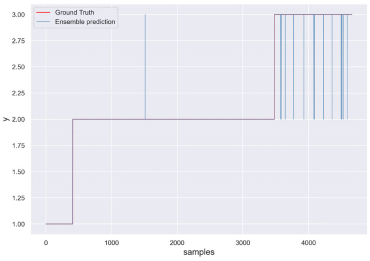
(c) CM of INFUSION



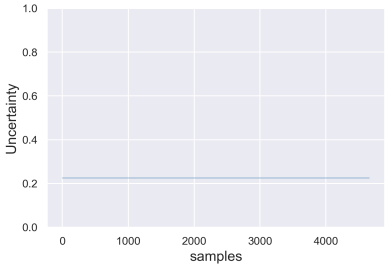
(d) Predictions of KLAFATE



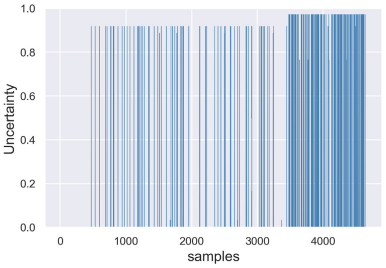
(e) Predictions of ECET



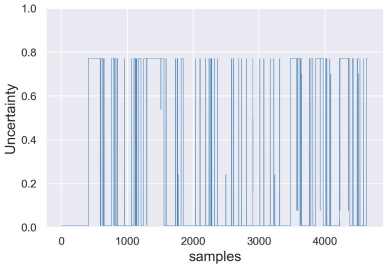
(f) Predictions of INFUSION



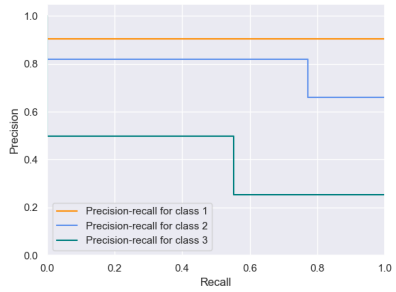
(g) UQ of KLAFATE



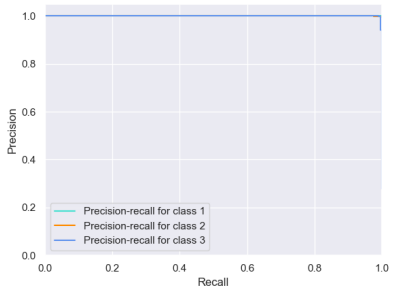
(h) DSET UQ of ECET



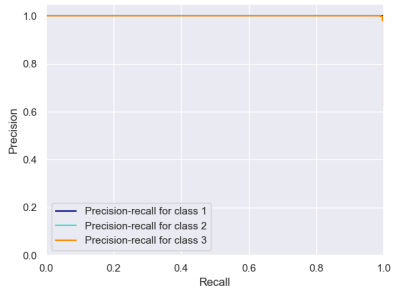
(i) DSET UQ of INFUSION



(j) PR-RE curve of KLAFATE



(k) PR-RE curve of ECET



(l) PR-RE curve of INFUSION

FIGURE 10. Results using different models KLAFATE, ECET (MC EC M3), and INFUSION (MC EC M3 and KLAFATE) using cases (1,2,3): Confusion matrices (a)-(c), classification results (d)-(f), DSET UQ (g)-(i), and precision-recall curves (j)-(l).

TABLE 12. Inference results of selected MC ECs using the cases (1,2,3) of the *BGS* dataset, and F1-score.

Fault	MC EC			INDIV		
	M3	M4	M5	DTR	KNN	ADB
1	1	1	1	1	1	1
2	1	1	1	1	1	0.97
3	0.99	0.99	0.99	1	0.99	0.91
Avg F1-score	1	1	1	1	1	0.96

rules is a challenging task and often time demanding. An additional positive characteristic of the knowledge-based model relies on its explainability: an expert user can directly observe the logic and transform the rules.

Alternatively, the data-based models using ECET outperformed the knowledge-based model, which is clearly reflected in the F1-scores of Table 13. However, the relationships between the features and outputs are often hidden (except for data-based models such as DTR, where the rules can be observed). It is important to remark on the number of features the models use, in which the knowledge-based models are built using less than ten

TABLE 13. Inference results of knowledge-based model KLAFATE, data fusion models (ECET and SVM-KNN), and knowledge and data fusion (INFUSION and SVM-KNN-KLAFATE) using the cases (1, 2, 3) of the *BGS* dataset, and F1-score. The ECET models are the MC ECs M3, M4 and M5. The INFUSION models are IFS3, IFS4, and IFS5.

Fault	Knowledge	Data Fusion				Knowledge and Data Fusion			
	KLAFATE	SVM-KNN	ECET			SVM-KNN-KLAFATE	INFUSION		
			M3	M4	M5		IFS3	IFS4	IFS5
1	0.95	1	1	1	1	1	1	1	1
2	0.79	0.97	1	1	1	0.98	1	1	1
3	0.52	0.92	0.99	0.99	0.99	0.95	0.99	0.99	0.99
Avg F1-score	0.75	0.96	1.00	1.00	1.00	0.98	1.00	1.00	1.00

features. In contrast, the ECET models are built using 133 features.

The fusion of data-based and knowledge-based models slightly improved the overall system's performance. The fusion model SVM-KNN-KLAFATE presented an improvement of fault 3 to the fusion model SVM-KNN, with scores of 0.95 and 0.92, respectively. In the case of INFUSION, the ECET results were already outstanding, resulting in a predominant effect on the fusion. The poor performance of some fault cases of KLAFATE did not affect the system performance.

The INFUSION methodology performed a fusion of the KLAFATE knowledge-based model and the ECET data-based models. No performance changes were reported since the ECET data-based models (M3, M4, and M5) presented already outstanding performance, and the INFUSION models (IFS3, IFS4, and IFS5) presented the same performance.

The INFUSION methodology presented a novel general framework that combines predictions of models of different domains. This general framework differentiates from the state-of-the-art, which presents a novel fault detection system that can combine n number of information sources (regardless of the model's information source) to achieve more robust predictions, as it was showcased in the results.

VII. CONCLUSION

We presented a novel adaptive information fusion methodology for fault detection systems using evidence theory and uncertainty quantification. We focused on two main topics of the fault detection system: improving anomaly detection and information fusion. The anomaly detection system was improved by adding the capability of automatic retraining of the models while feeding unknown fault cases into the data.

For this purpose, we presented an EC retraining strategy based on uncertainty monitoring of the EC predictions. The retraining results of the use case validated the approach, in which the benchmark TE dataset was used to test different anomalies. Different experiments were performed to analyze the impact of the main parameters of the retraining approach, namely, threshold size, window size, and detection patience. Though the results were not entirely comparable with the literature, the approach's claim was validated, in which the EC updated the models while feeding unknown fault cases into the data. The results demonstrated the applicability of the proposed approach to multivariate (industrial) data in a supervised classification problem, which can be extrapolated to different (industrial) use cases following the methodology.

Furthermore, we proposed an information fusion approach to combine an EC and a knowledge-based model at the decision level. The approach was tested using the data of an industrial setup. We performed an ablation study to compare the performance of the systems, namely, EC, knowledge-based models, and the fusion of both systems. The system performance reported better results while using an information fusion of the EC and the knowledge-based model, confirming, thus, the approach's claim. The proposed information fusion approach addressed combining models of different domains using a general framework for decision-level fusion. This general framework can be applied to address the information fusion of other (industrial) domains following the procedure defined in this research.

Future research includes a semi-supervised approach in which the EC results are confronted with an unsupervised model. This approach aims to validate the data samples of the detected anomaly by examining the location of the samples in the input space. Prospective works include comparing our approach with one-class models (e.g., autoencoders and isolation forests) and integrating such models with the information fusion framework. Furthermore, we will test other combination rules to improve the anomaly detection results, thus increasing the size of anomalous data. In addition, we will extend our information fusion framework by integrating state-of-the-art knowledge-based models (e.g., ontology-based approaches and knowledge frameworks for multi-modal and multi-structured data). Finally, we want to enhance our approach using techniques of explainable artificial intelligence (XAI) on the base classifiers of the EC in order to gain more insights into the learning process.

REFERENCES

[1] G. Gallo, F. D. Rienzo, F. Garzelli, P. Ducange, and C. Vallati, "A smart system for personal protective equipment detection in industrial environments based on deep learning at the edge," *IEEE Access*, vol. 10, pp. 110862–110878, 2022.

[2] V. Mazzia, A. Khaliq, F. Salvetti, and M. Chiaberge, "Real-time apple detection system using embedded systems with hardware accelerators: An edge AI application," *IEEE Access*, vol. 8, pp. 9102–9114, 2020.

[3] J. Rahm, M. Graube, R. Müller, T. Klaeger, L. Schegner, A. Schult, R. Bonse, S. Carsch, L. Oehm, and L. Urbas, "KoMMDia: Dialogue-driven assistance system for fault diagnosis and correction in cyber-physical production systems," IEEE 23rd in *Proc. Int. Conf. Emerg. Technol. Factory Automat. (ETFA)*, vol. 1, Sep. 2018, pp. 999–1006.

[4] M. Z. M. Shamim, "Hardware deployable edge-AI solution for prescreening of oral tongue lesions using TinyML on embedded devices," *IEEE Embedded Syst. Lett.*, vol. 14, no. 4, pp. 183–186, Dec. 2022.

[5] A. Gagliardi, V. Staderini, and S. Saponara, "An embedded system for acoustic data processing and AI-based real-time classification for road surface analysis," *IEEE Access*, vol. 10, pp. 63073–63084, 2022.
[6] M. Quandt, H. Stern, W. Zeitler, and M. Freitag, "Human-centered design of cognitive assistance systems for industrial work," *Proc. CIRP*, vol. 107, pp. 233–238, Jan. 2022.
[7] R. Sochor, L. Kraus, L. Merkel, S. Braunreuther, and G. Reinhart, "Approach to increase worker acceptance of cognitive assistance systems in manual assembly," *Proc. CIRP*, vol. 81, pp. 926–931, Jan. 2019.
[8] B. Pokorni and C. Constantinescu, "Design and configuration of digital assistance systems in manual assembly of variant-rich products based on customer journey mapping," *Proc. CIRP*, vol. 104, pp. 1777–1782, Jan. 2021.
[9] X. Qin and X. J. Zhang, "An industrial dyeing recipe recommendation system for textile fabrics based on data-mining and modular architecture design," *IEEE Access*, vol. 9, pp. 136105–136110, 2021.
[10] J. Jin, H. Guo, J. Xu, X. Wang, and F.-Y. Wang, "An end-to-end recommendation system for urban traffic controls and management under a parallel learning framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1616–1626, Mar. 2021.
[11] L. Nagy, T. Ruppert, and J. Abonyi, "Human-centered knowledge graph-based design concept for collaborative manufacturing," in *Proc. IEEE 27th Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2022, pp. 1–8.
[12] A.-P. Lohfink, S. D. Duque Anton, H. Leitte, and C. Garth, "Knowledge rocks: Adding knowledge assistance to visualization systems," *IEEE Trans. Vis. Comput. Graphics*, vol. 28, no. 1, pp. 1117–1127, Jan. 2022.
[13] F. Arévalo, M. T. Ibrahim, and A. Schwung, "Production assessment using a knowledge transfer framework and evidence theory," *IEEE Access*, vol. 10, pp. 89134–89152, 2022.
[14] J. Bakakeu, M. Brossog, J. Zeitler, J. Franke, S. Tolksdorf, H. Klos, and J. Peschke, "Automated reasoning and knowledge inference on OPC UA information models," in *Proc. IEEE Int. Conf. Ind. Cyber Phys. Syst. (ICPS)*, May 2019, pp. 53–60.
[15] T. Bauernhansl, M. Weyrich, L. Zarco, T. Müller, P. Marks, T. Schlegel, and J. Siegert, "Semantic structuring of elements and capabilities in ultra-flexible factories," *Proc. CIRP*, vol. 93, pp. 335–340, Jan. 2020.
[16] N. Sahlab, S. Kamm, T. Müller, N. Jazdi, and M. Weyrich, "Knowledge graphs as enhancers of intelligent digital twins," in *Proc. 4th IEEE Int. Conf. Ind. Cyber-Phys. Syst. (ICPS)*, May 2021, pp. 19–24.
[17] S. Biffl, S. Kropatschek, E. Kiesling, K. Meixner, and A. Luder, "Risk-driven derivation of operation checklists from multi-disciplinary engineering knowledge," in *Proc. 20th Int. Conf. Ind. Inform. (INDIN)*, Jul. 2022, pp. 7–14.
[18] A.-S. Wilde, F. Wanielik, M. Rolinck, M. Mennenga, T. Abraham, F. Cerdas, and C. Herrmann, "Ontology-based approach to support life cycle engineering: Development of a data and knowledge structure," *Proc. CIRP*, vol. 105, pp. 398–403, Jan. 2022.
[19] S. Schiller, M. Landwehr, G. Vinogradov, I. Dimitriadis, H. Akyürek, J. Lipp, P. Ganser, and T. Bergs, "Towards ontology-based lifecycle management in blisk manufacturing," *Proc. CIRP*, vol. 112, pp. 280–285, Jan. 2022.
[20] R. Witt, A.-L. Knott, S. Cramer, and R. H. Schmitt, "Application of a product-centred process-independent meta-model for multi-stage production data to enable predictive quality for additive manufacturing," *Proc. CIRP*, vol. 118, pp. 799–804, Jan. 2023.
[21] J. Wessel, A. Turetskyy, O. Wojahn, T. Abraham, and C. Herrmann, "Ontology-based traceability system for interoperable data acquisition in battery cell manufacturing," *Proc. CIRP*, vol. 104, pp. 1215–1220, Jan. 2021.
[22] R. Glawar, F. Ansari, C. Kardos, K. Matyas, and W. Sihn, "Conceptual design of an integrated autonomous production control model in association with a prescriptive maintenance model (PriMa)," *Proc. CIRP*, vol. 80, pp. 482–487, Jan. 2019.
[23] A. Padovano, F. Longo, L. Nicoletti, L. Gazzaneo, A. Chiurco, and S. Talarico, "A prescriptive maintenance system for intelligent production planning and control in a smart cyber-physical production line," *Proc. CIRP*, vol. 104, pp. 1819–1824, Jan. 2021.
[24] D. Gors, M. Birem, R. D. Geest, C. Domken, V. Zogopoulos, S. Kauffmann, and M. Witters, "An adaptable framework to provide AR-based work instructions and assembly state tracking using an ISA-95 ontology," *Proc. CIRP*, vol. 104, pp. 714–719, Jan. 2021.
[25] G. Bruno, D. Antonelli, and A. Villa, "A reference ontology to support product lifecycle management," *Proc. CIRP*, vol. 33, pp. 41–46, Jan. 2015.
[26] G. Bitsch, P. Senjic, and J. Askin, "Integration of legacy systems to cyber-physical production systems using semantic adapters," *Proc. CIRP*, vol. 118, pp. 259–263, Jan. 2023.
[27] I. Gräßler, D. Wiechel, and D. Roesmann, "Integrating human factors in the model based development of cyber-physical production systems," *Proc. CIRP*, vol. 100, pp. 518–523, Jan. 2021.
[28] F. Arévalo, M. T. Ibrahim, M. P. C. Alison, and A. Schwung, "Anomaly detection using ensemble classification and evidence theory," *IEEE Access*, vol. 11, pp. 53545–53587, 2023.
[29] J. Zhao, Y. Shi, W. Liu, T. Zhou, Z. Li, and X. Li, "A hybrid method fusing frequency recognition with attention detection to enhance an asynchronous brain-computer interface," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 31, pp. 2391–2398, 2023.
[30] V.-E. Neagoe and G.-L. Ghenea, "An approach of Dempster–Shafer decision fusion to diagnose COVID-19 in chest X-ray imagery by using controlled asymmetric training of the two CNNs ensemble," in *Proc. 14th Int. Conf. Electron., Comput. Artif. Intell. (ECAI)*, Jun. 2022, pp. 1–6.
[31] Z. Huo, M. Martínez-García, Y. Zhang, and L. Shu, "A multisensor information fusion method for high-reliability fault diagnosis of rotating machinery," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022.
[32] A.-L. Jousselme, "Semantic criteria for the assessment of uncertainty handling fusion models," in *Proc. 19th Int. Conf. Inf. Fusion (FUSION)*, Jul. 2016, pp. 488–495.
[33] J. Useya and S. Chen, "Comparative performance evaluation of pixel-level and decision-level data fusion of Landsat 8 OLI, Landsat 7 ETM+ and Sentinel-2 MSI for crop ensemble classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 4441–4451, Nov. 2018.
[34] J. P. D. Villiers, K. Laskey, A. L. Jousselme, E. Blasch, A. D. Waal, G. Pavlin, and P. Costa, "Uncertainty representation, quantification and evaluation for data and information fusion," in *Proc. 18th Int. Conf. Inf. Fusion (Fusion)*, Washington, DC, USA, 2015, pp. 50–57.
[35] D. Girardi, J. Kueng, and A. Holzinger, "A domain-expert centered process model for knowledge discovery in medical research: Putting the expert-in-the-loop," in *Proc. Brain Inform. Health, 8th Int. Conf. (BIH)*, vol. 9250, Sep. 2015, pp. 389–398.
[36] F. Arévalo, T. Nguyen, and A. Schwung, "Assistance system for a bulk good system based on information fusion," in *Proc. 22nd IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2017, pp. 1–8.
[37] L. Jiao, T. Denoeux, and Q. Pan, "A hybrid belief rule-based classification system based on uncertain training data and expert knowledge," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 46, no. 12, pp. 1711–1723, Dec. 2016.
[38] Y.-x. Sun and L. Song, "An approach of talents evaluation based on multi-expert decision-making," in *Proc. 33rd Chin. Control Decis. Conf. (CCDC)*, May 2021, pp. 3868–3871.
[39] K. Mukai, S. Kumano, and T. Yamasaki, "Improving robustness to out-of-distribution data by frequency-based augmentation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2022, pp. 3116–3120.
[40] J. Gawlikowski, S. Saha, A. Kruspe, and X. X. Zhu, "Towards out-of-distribution detection for remote sensing," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2021, pp. 8676–8679.
[41] G. S. Chadha, I. Islam, A. Schwung, and S. X. Ding, "Deep convolutional clustering-based time series anomaly detection," *Sensors*, vol. 21, no. 16, p. 5488, Aug. 2021.
[42] T. J. Rato and M. S. Reis, "Fault detection in the Tennessee eastman benchmark process using dynamic principal components analysis based on decorrelated residuals (DPCA-DR)," *Chemometric Intell. Lab. Syst.*, vol. 125, pp. 101–108, Jun. 2013.
[43] Y. Okawa and K. Kobayashi, "Concept drift detection via boundary shrinking," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–8.
[44] L. Kidane, P. Townend, T. Metsch, and E. Elmroth, "When and how to retrain machine learning-based cloud management systems," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. Workshops (IPDPSW)*, May 2022, pp. 688–698.
[45] J. Zhang, T. Wang, W. W. Y. Ng, and W. Pedrycz, "KNNENS: A k-nearest neighbor ensemble-based method for incremental learning under data stream with emerging new classes," in *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 9520–9527, Nov. 2023, doi: 10.1109/TNNLS.2022.3149991.
[46] Y. Yang, Z.-Q. Sun, H. Zhu, Y. Fu, Y. Zhou, H. Xiong, and J. Yang, "Learning adaptive embedding considering incremental class," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 3, pp. 2736–2749, Mar. 2023.
[47] J. Li, R. Huang, G. He, S. Wang, G. Li, and W. Li, "A deep adversarial transfer learning network for machinery emerging fault detection," *IEEE Sensors J.*, vol. 20, no. 15, pp. 8413–8422, Aug. 2020.

[48] Y. Kongsorot, P. Horata, and P. Musikawan, "An incremental kernel extreme learning machine for multi-label learning with emerging new labels," *IEEE Access*, vol. 8, pp. 46055–46070, 2020.
[49] X.-S. Wei, H.-J. Ye, X. Mu, J. Wu, C. Shen, and Z.-H. Zhou, "Multi-instance learning with emerging novel class," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 5, pp. 2109–2120, May 2021.
[50] G. Shafer, *A Math. Theory Evidence*. Princeton, NJ, USA: Princeton Univ. Press, 1976.
[51] R. R. Yager, "On the Dempster–Shafer framework and new combination rules," *Inf. Sci.*, vol. 41, no. 2, pp. 93–137, Mar. 1987.
[52] Y. Cheng and R. L. Kashyap, *Comparison Bayesian Dempster's Rules Evidence Combination*. Amsterdam, The Netherlands: Springer, 1988, pp. 427–433.
[53] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Comput. Chem. Eng.*, vol. 17, no. 3, pp. 245–255, Mar. 1993.
[54] G. S. Chadha, A. Panambilly, A. Schwung, and S. X. Ding, "Bidirectional deep recurrent neural networks for process fault classification," *ISA Trans.*, vol. 106, pp. 330–342, Nov. 2020.
[55] F. Pedregosa, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12 no. 10, pp. 2825–2830, 2012.
[56] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. 33rd Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates Inc., 2019, pp. 8026–8037.
[57] *Anaconda Software Distribution. Computer Software. Vers. 2-2.4. 0*, Anaconda, Austin, TX, USA, 2016.
[58] M. Panda, A. A. A. Mousa, and A. E. Hassanien, "Developing an efficient feature engineering and machine learning model for detecting IoT-botnet cyber attacks," *IEEE Access*, vol. 9, pp. 91038–91052, 2021.

FERNANDO ARÉVALO (Senior Member, IEEE) received the Engineering degree in electrical engineering from Universidad de El Salvador, San Salvador, El Salvador, in 2005, and the M.Sc. degree in systems engineering and engineering management from the South Westphalia University of Applied Sciences, Soest, Germany, in 2012. He is currently pursuing the Ph.D. degree with Ruhr-Universität Bochum, Bochum, Germany. From 2008 to 2015, he was a Project Engineer with Kimberly Clark and MTU Friedrichshafen. From 2016 to 2021, he was a Research Assistant with the Automation Technology Department, South Westphalia University of Applied Sciences. He was a Senior Consultant of the topics of the IIoT, digitalization of the industry, and data sciences with Adesso SE Company. He is currently a Data Scientist on flood forecasting and early warning systems with Wupper Association (Wupperverband). His research interests include data-driven fault diagnosis, information fusion, uncertainty quantification, knowledge extraction, the IoT, and augmented reality with applications in the process industry and environment monitoring.

M. P. CHRISTIAN ALISON received the dual bachelor's degree in industrial engineering from Swiss German University, Indonesia, and the South Westphalia University of Applied Sciences, Soest, Germany, in 2020, and the M.Sc. degree in systems engineering and engineering management from the South Westphalia University of Applied Sciences, in 2022. Since 2022, he has been a XR Consultant with PACE Aerospace & IT Company. His research interests include the future applications of augmented reality and virtual reality for process improvement and knowledge internalization.

M. TAHASANUL IBRAHIM (Member, IEEE) received the bachelor's degree in electrical and electronics engineering from American International University, Bangladesh, in 2014, and the M.Sc. degree in systems engineering and engineering management from the South Westphalia University of Applied Sciences, Soest, Germany, in 2020. Since 2021, he has been a Research Assistant with the Automation Technology and Learning Systems Department, South Westphalia University of Applied Sciences. His research interests include information fusion based on uncertainty, rule-based systems, deep neural networks in computer vision, classified maneuver prediction and knowledge attenuation techniques, and back-end development in container systems.

ANDREAS SCHWUNG was a Research and Development Engineer with MAN Diesel & Turbo SE, Oberhausen, Germany, from 2011 to 2015. Since 2015, he has been a Professor in automation technology with the South Westphalia University of Applied Sciences, Soest, Germany. His research interests include model-based control, networked automation systems, and intelligent data analytics with application in manufacturing, process industry, and electromobility.