Contrastive Ensemble Active Learning for Data Quality Improvements for Resilient Manufacturing AI Model Prediction

Xuancheng Jin[1], Yingyan Zeng[2], Hui Liu[1], Ran Jin[1]
[1]*Grado Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, USA*
[2]*Department of Mechanical and Materials Engineering, University of Cincinnati, Cincinnati, USA*
{xuanchengj, huiliu, jran5}@vt.edu, zengyy@ucmail.uc.edu

***Abstract*—In Manufacturing Industrial Internet, Deep Neural Networks (DNNs) are widely used to improve production efficiency and quality via computation services. The resilience of DNNs in inference tasks is easily affected by data quality, computation pipeline, and cyber-physical layer of manufacturing. To create a proactive-adaptive resilient Artificial Intelligence (AI) models, it is important to predict the AI model performance based on online streaming data, such as runtime metrics of computation tasks. The class imbalance (i.e., a small percentage of AI performance degradation under hazardous conditions) and time-varying distribution are commonly encountered, which makes the AI model performance prediction difficult. We propose a contrastive ensemble active learning (ConEAL) framework that couples a contrastive representation learner with a contextual-bandit ensemble active-learning policy proposed in our earlier work. The representation learning module standardizes predictors and creates stochastic views for each sample to learn more stable representation space. By employing a staged loss function to train the encoder, the representation space adapts to the data stream as new classes emerge. Based on the projected representation, ConEAL ensembles different sampling agentsand adaptively weights them with a composite reward in order to acquire informative samples to reduce the annotation efforts. Validation is performed based on an AI resilience experiment in the Fog-Cloud Manufacturing Industrial Internet testbed, where ConEAL achieves better prediction performance of abnormal AI model performance than benchmark methods while requiring fewer labeled samples.**

***Index Terms*—AI resilience, anomaly detection, contrastive learning, ensemble active learning, Manufacturing Industrial Internet**

I. Introduction

In Manufacturing Industrial Internet (MII), Deep Neural Networks (DNNs) are increasingly deployed to improve production efficiency and quality through computation services. These services include quality inspection, predictive maintenance, and process optimization, all of which depend on low-latency inference and continuous sensing [1]. However, the resilience of DNNs in such environments is easily affected by hazardous from multiple layers: noisy and incomplete input data, instability and singularity in computation pipelines, and failures in the cyber–physical infrastructure [2]. When resilience of AI models is compromised, manufacturing performance may degrade, leading to incorrect decisions and system-level risks.

Resilience in industrial AI systems should not be limited to detecting anomalies after they occur. Instead, proactive–adaptive resilience is required: the capability to predict the performance of AI models in inference tasks (e.g., the deployment stage of AI models after training) in advance and adapt accordingly. Such prediction based on online streaming data (e.g., runtime computation metrics) enables timely intervention and adjustment to sustain reliable operation. Without this forward-looking capacity, conventional AI systems remain reactive, providing only delayed warnings that may come too late to prevent costly disruptions in manufacturing. Therefore, building proactive–adaptive resilience, i.e., the ability to predict potential AI performance degradation and adapt to changing conditions, is critical for resilient manufacturing AI systems.

As shown in Fig. 1, manufacturing computation services are deployed in a fog–cloud architecture, where AI models are executed on distributed fog nodes coordinated by the cloud. The AI pipelines process raw data streams and return inference results that are used for decision-making in real time. Meanwhile, the runtime metrics which record the operation status of each computation node such as CPU and memory utilization, download and upload bandwidths etc., [3] are collected to monitor the resilience of the AI system. However, the collected runtime metrics are inevitably affected by multiple hazards. These hazards contain data-level issues such as noisy data, shifting distribution, imbalanced class distribution, which directly reduce the quality of the features provided to the AI pipeline. In addition, there also exist system-level failures at the fog or communication layer. These potential failures hinder the reliable execution of inference tasks. Together, they create noisy, imbalanced, and time-varying data streams that cause the AI models to degrade into low-performance states. Moreover, because anomaly detection modules cannot proactively adapt to these conditions, the system often fails to provide early warnings, highlighting the need for resilience-oriented learning frameworks. Contrastive Learning (CL) [4] offers a promising direction, as it can build robust representations from

The authors acknowledge the funding support from National Science Foundation, CMMI-2331985.

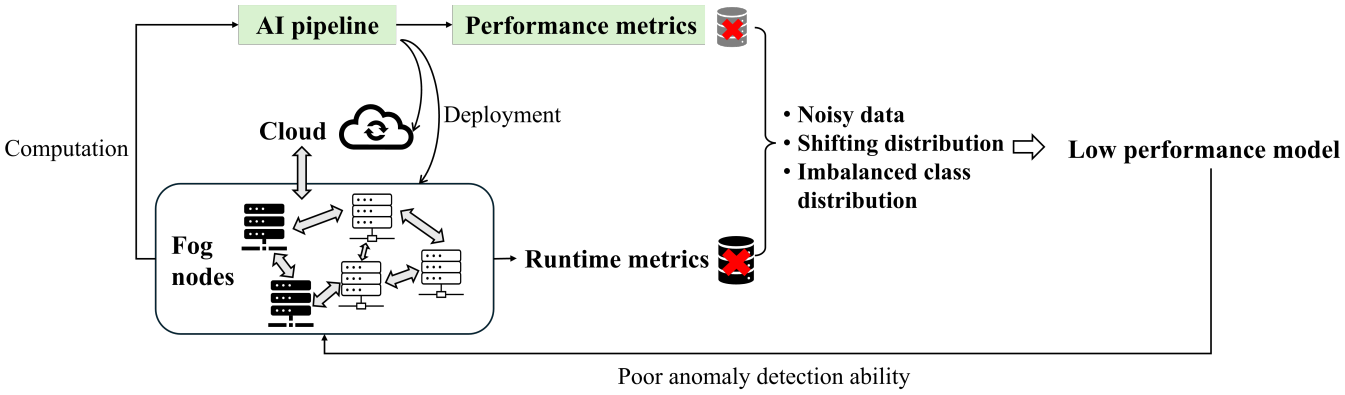


Fig. 1. Challenges of AI resilience in a Manufacturing Industrial Internet. AI pipelines are deployed on distributed fog nodes, where the performance metrics and runtime metrics are generated and collected during the inference tasks. However, these metrics are often affected by unexpected hazards such as noisy data, shifting distribution, and imbalanced class distribution, which can lead to degraded AI models with poor anomaly detection ability. Red crosses on the dataset icons indicate the impact of these unexpected hazards on both performance metrics and runtime metrics.

normal-only data and adaptively refine them once anomalies appear, thereby supporting more reliable prediction of AI performance under such streaming hazards.

To overcome the challenges of data imbalance, time-varying distribution, and one-class initialization, we propose a contrastive-learning–enhanced ensemble active-learning (ConEAL) framework. Firstly, we build a contrastive representation learner. The encoder standardizes predictors, creates two stochastic views per instance to introduce perturbations that encourage invariance, so that a robust representation space can be learned with missing data. Then, it is trained with a staged loss function to progressively adapt the representation space as new classes emerge. During the one-class initialization phase (i.e., the first stage), unsupervised contrastive learning is employed to extract invariant and robust feature representations without requiring abnormal samples. When abnormal classes emerge (i.e., the second stage), contrastive signals are incorporated to transform the representation space, thereby making anomalies more distinguishable and enhancing downstream classification performance. Furthermore, based on the proposed Ensemble Active Learning by Contextual Bandits (CBEAL) [5], we design a novel composite reward function that adaptively tunes the weights of integrated active learning agents, This enables the framework to acquire streaming data more efficiently and improve the overall predictive resilience.

We validated the performance of the proposed ConEAL on a dataset collected from an AI system in MII testbed [3]. ConEAL achieves higher Macro-F1 scores with fewer labeled samples, directly enhancing resilience in streaming MII settings, rather than relying on exhaustive annotations.

The remainder of this paper is organized as follows. Section II summarizes the related work. Section III introduces the proposed ConEAL method. The methodology is further validated by a real MII case study in Section IV. Finally, the conclusions and future work are discussed in Section V.

II. Related Work

A. AI Computation System

In the MII, AI methods, particularly DNNs, are widely adopted for tasks such as quality modeling and virtual inspection [6]. These models learn informative representations from complex industrial data, thereby enhancing product quality, improving production efficiency, and lowering operational costs [7]. As a result, AI computation systems have become essential to manufacturing processes and decision-making chains [8].

The AI computation systems operate in highly dynamic and uncertain environments, where data may contain noise, missing values, and distribution shifts [3]. In addition, AI pipelines are vulnerable to singularity, while computation and communication infrastructures can experience node crashes, latency, or instability [2]. These challenges make it critical to ensure that AI computation systems remain uninterrupted under such complex conditions.

B. Anomaly Prediction and AI Resilience

To address the reliability challenges of AI computation systems in complex environments, the concept of AI resilience has become essential to ensuring stable operation in the MII. The prediction of abnormal AI model performance serves as a key mechanism to enhance AI resilience, as it enables the AI system to detect early signs of abnormal behavior and initiate timely interventions before severe performance degradation occurs [9]–[11].

Existing methods for anomaly prediction include supervised classification models and deep learning techniques. For example, supervised models such as Random Forest classifiers have been successfully applied to network anomaly prediction, achieving a great performance improvement in real-world traffic datasets [12]. For deep learning methods, autoencoder-based approaches are widely used in industrial monitoring study employs an autoencoder to model normal multimodal sensor behavior and uses reconstruction error thresholds to

detect anomalies in cooling systems [13]. However, these approaches face several limitations. Most of approaches rely heavily on large amounts of labeled anomaly data, which is often scarce in real-world settings. They also suffer from data imbalance, making it difficult for them to generalize well in real-world scenarios.

C. Contrastive Learning

In recent years, contrastive learning has been applied to address class imbalance in various modeling tasks by improving the representation of minority classes [14]–[16]. For example, in medical image segmentation, contrastive learning has been used to minimize intra-class distance and maximize inter-class separation, enabling clearer clustering of minority samples in the feature space [17]. The BaCon method targets imbalanced semi-supervised learning by designing contrastive objectives at the feature level and incorporating class centers with temperature scaling to mitigate pseudo-label bias [18]. ResCom introduces a class-balanced contrastive loss with a Siamese queue and hard pair mining strategy to improve discrimination under long-tailed distributions [19].

Given its ability to learn robust representations under class imbalance, contrastive learning holds strong potential for AI system anomaly prediction. In real-world manufacturing environments, abnormal system states are often rare and underrepresented. By leveraging contrastive objectives to model the differences between normal and abnormal states, we can enhance the AI system's ability to recognize subtle deviations before explicit failures occur. Unlike traditional approaches that rely on extensive labeled anomaly data or fixed thresholds, contrastive learning provides a more flexible and data-efficient solution. This makes contrastive learning a promising solution for early and accurate anomaly prediction in AI systems, especially when labeled anomaly data is extremely scarce.

III. Proposed Methodology

We propose a ConEAL framework that learns a representative space with discriminative and structure-aware capabilities via contrastive learning, on which an ensemble active learning model is developed to efficiently and effectively select informative samples from the full dataset to improve modeling performance.

The overview of the proposed method is shown in Fig. 2. Consider the online data annotation scenario with a raw data $\mathcal{X}_t$ collected by sensors or computation logs at time t, $t = 1, 2, \ldots, T$. In $\mathcal{X}_t$, there are multivariate time series data including runtime metrics and computation records of varying length, together with tabular outputs describing the learning performance of deployed AI pipelines. The feature space combines static process variables with window-level summaries of time series measurements: raw streams are segmented into windows and converted into statistical descriptors (e.g., mean, standard deviation, skewness, kurtosis). After preprocessing and alignment, features are retained as tabular inputs $\mathbf{x}_t$ for the base learner f_t. Here we define the inputs $\mathbf{x}_t$ as predictors. The streaming data are first encoded by the proposed contrastive learning module, where the encoder f_t extracts latent representations $\mathbf{h}_t$ and the projection head g_ϕ maps them into normalized embeddings $\mathbf{z}_t$ for contrastive optimization. Once pretrained, the base learner outputs class probability vector $\mathbf{P}^f(\hat{y}_t \mid D_t)$ serve as the representation for downstream prediction.

These predictions are then passed to a set of heterogeneous active learning agents $AG_1, AG_2, \ldots, AG_Q$. Each agent produces an acquisition decision based on its own sampling criterion. Their outputs are aggregated into a weighted ensemble decision vector, where the agent weights $\alpha_{1,t}, \alpha_{2,t}, \ldots, \alpha_{Q,t}$ are adaptively updated using the Exp4.P-EWMA strategy according to the reward r_t received from annotation feedback [5]. If the ensemble decision accepts the sample, the true label y_t is queried from the annotator; otherwise the sample is discarded without annotation. However, if the labeled pool has already reached the budget B, additional samples are rejected even if they were selected by the ensemble. This ensures that annotation costs remain bounded while focusing effort on the most informative instances.

We assume that the classification problem has c classes, and $y_t \in C = \{0, 1\}$ is the label of the sample $\mathbf{x}_t$. Denote the labeled data pool at time t as $D_t = (\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_{m_t}, y_{m_t})$ with $|D_t| = m_t$. The base learner f_0 is first trained by initial dataset D_0. To develop the active learning agents for ConEAL framework, we make some assumptions: (i) The sample size of the initial training set D_0 is not large enough to guarantee satisfactory modeling performance and the samples in D_0 are all from normal class. (ii) The streaming data have highly imbalanced class distribution. (iii) There is an unlabeled sample $\mathbf{x}_t$ in the stream data at each time stamp without a cost. (iv) Human annotators can provide the error-free, accurate anotations of a sample $\mathbf{x}_t$ if the model decides to acquire it. (v) The base learner f_t will be online updated (i.e., online machine learning) whenever acquiring certain number (po) of samples.

A. Contrastive Learning Representation Framework

A contrastive learning framework is tailored to stream multivariate data for binary detection. The previous assumption is that we have 2 classes. The core idea of contrastive learning is to learn a representation space by enforcing samples from the same instance (i.e., positive pairs) to be mapped closer together, while pushing the representations of different instances (i.e., negative pairs) apart [4]. This property makes contrastive learning especially useful in scenarios with limited labels or missing classes, since the model can first learn a general representation space from unlabeled data and later adapt to specific classes when labels become available.

Specifically, we firstly standardize inputs and generate two stochastic views per instance via elementwise masking and small Gaussian noise. Then, we pretrain each view with a lightweight multilayer perceptron (MLP) [20] encoder and a non-linear projection head with ℓ_2-normalized outputs. During the pretraining stage, both the encoder and a projection head are jointly optimized with contrastive objectives and we adopt

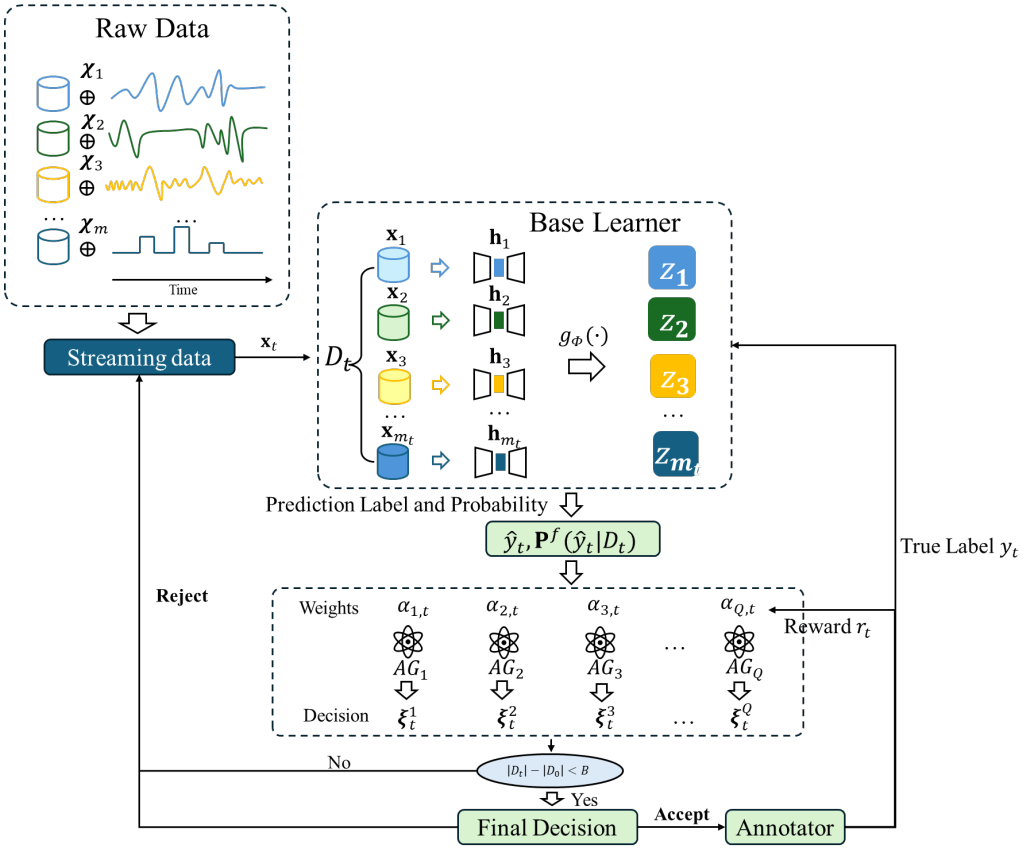


Fig. 2. Overview of the proposed ConEAL framework. During the one-class initialization phase, the streaming data are standardized and augmented with stochastic views by the encoders, and then projected to a representation space by contrastive learning as the pretraining process. In the online training phase, the streaming data are encoded and projected with encoders and projector heads to the informative representation space and then selectively annotated by Ensemble Active Learning by Contextual Bandits (CBEAL) [5] with a novel reward function adaptively tuning the sampling strategy to the data stream with imbalanced classs and shifting distribution.

a staged training strategy: (i) When only a single class has been observed, we pretrain with an unsupervised NT-Xent loss [21]. This contrastive loss constructs positive pairs from different stochastic views of the same instance and treats other instances as negatives, thereby encouraging the model to learn invariant and discriminative representations without requiring class labels. Such pretraining is particularly suitable for the normal data initialization phase in streaming settings, as it allows the model to exploit structural information from unlabeled data. (ii) Once more classes are available, we switch to a margin triplet loss constructed across classes [22], which explicitly enforces inter-class separability by pulling samples of the same class closer while pushing apart samples from different classes. This staged design ensures that the representation space is robustly created under limited supervision and then progressively refined for class-discriminative prediction as more abnormal samples emerge. After pretraining is completed , the projection head is discarded, and a class-balanced logistic classifier is trained on the frozen encoder representation.

Through enforcing invariance across augmented views, this procedure yields representations that are robust to noisy features, stabilizing downstream decision boundaries. Since staged contrastive objectives will progressively incorporate new abnormal classes, the learned latent space becomes more class-separable, with intra-class samples clustered tightly and inter-class samples pushed farther apart, so that a simple linear head suffices under limited labels.

1) Preliminaries and Notation: Let $\mathbf{x} \in \mathbb{R}^d$ denote the predictors of a training sample. We first standardize features

$$\tilde{\mathbf{x}} = \mathrm{diag}(\boldsymbol{\sigma})^{-1}(\mathbf{x} - \boldsymbol{\mu}), \tag{1}$$

where $\boldsymbol{\mu}$ is the feature-wise mean vector and $\boldsymbol{\sigma}$ is the feature-wise standard deviation matrix. Then, two stochastic views are generated by a light-weight augmentation for tabular data:

$$t(\tilde{\mathbf{x}}) = \mathbf{m} \odot \tilde{\mathbf{x}} + \boldsymbol{\epsilon}, \quad \mathbf{m} \sim \mathrm{Bernoulli}(p)^d, \ \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_n^2 \mathbf{I}), \tag{2}$$

where $\odot$ denotes elementwise product. Eq. (2) implements random feature dropout and small Gaussian noise. p is the keep probability that controls the masking strength in the stochastic view generation step. This sugmentation mimics label missingness and acquisition perturbations in streams. By considering these scenarios in the framework, it improves the robustness of the learned representation via contrastive learning.

2) Encoder and Projection Head: We use a two-layer MLP encoder $f_\theta : \mathbb{R}^d \to \mathbb{R}^{128}$ to produce a representation

$$\mathbf{h} = f_\theta\big(t(\tilde{\mathbf{x}})\big). \tag{3}$$

The encoder output dimension is set to 128, following common practice in contrastive learning frameworks [4], which balances representation capacity and computational efficiency. The non-linear projection head $g_\phi : \mathbb{R}^{128} \to \mathbb{R}^P$ further maps $\mathbf{h}$ into a P dimensional latent space for contrastive training, where P is a tunable hyperparameter used only during pretraining and discarded afterward. To ensure that similarity is measured only by angular distance and not by vector magnitude, the output is further normalized by the ℓ_2 norm:

$$\mathbf{z} = g_\phi(\mathbf{h}), \qquad \bar{\mathbf{z}} = \frac{\mathbf{z}}{\|\mathbf{z}\|_2}. \tag{4}$$

This normalization step projects $\mathbf{z}$ onto the unit hypersphere, which is standard in contrastive learning frameworks [4], and facilitates stable training with cosine similarity. As observed by SimCLR [4], the contrastive objective on a small non-linear head improves the quality of the representation $\mathbf{h}$ used for downstream tasks.

3) Contrastive Loss Function: When only a single class has been observed, we optimize the unsupervised NT-Xent contrastive loss [21] to exploit invariances across augmented views of the same sample. Once both normal and abnormal classes appear, we switch to a margin-based triplet loss [22] in the encoder space to explicitly enforce inter-class separability. These two losses are grouped as subequations of Eq. (5).

$$\ell_{i,j} = -\log \frac{\exp\big(s(\bar{\mathbf{z}}_i, \bar{\mathbf{z}}_j)/\tau\big)}{\sum_{k=1}^{2\mathcal{N}} \mathbb{1}[k \neq i] \exp\big(s(\bar{\mathbf{z}}_i, \bar{\mathbf{z}}_k)/\tau\big)}. \tag{5a}$$

$$\mathcal{L}_{\mathrm{CL}} = \frac{1}{2\mathcal{U}} \sum_{u=1}^{\mathcal{U}} (\ell_{2u-1,2u} + \ell_{2u,2u-1}). \tag{5b}$$

$$\mathcal{L}_{\mathrm{tri}}^{(i)} = \max\big\{0,\ \|\mathbf{h}_i - \mathbf{h}_p\|_2^2 - \|\mathbf{h}_i - \mathbf{h}_n\|_2^2 + m\big\}. \tag{5c}$$

$$\mathcal{L}_{\mathrm{tri}} = \frac{1}{|\mathcal{T}|} \sum_{(i,p,n)\in\mathcal{T}} \mathcal{L}_{\mathrm{tri}}^{(i)}. \tag{5d}$$

Eq. 5a computes the log-softmax contrastive objective. Given a positive pair (i, j), the cosine similarity $s(\bar{\mathbf{z}}_i, \bar{\mathbf{z}}_j)$ is contrasted against similarities with all other $2\mathcal{N} - 1$ negatives in the batch. The temperature $\tau > 0$ scales the logits.

Eq. 5b averages $\ell_{i,j}$ across $\mathcal{U}$ unsupervised pairs, ensuring that different augmented views of the same instance are pulled together while pushing apart different instances. This stage is suitable for normal-only initialization, as it does not require class labels.

Eq. 5c compares an anchor i with a positive p (i.e., same class) and a negative n (i.e., different class). Triplets are formed across classes in the encoder space $\mathbf{h}$: for an anchor i of class $C \in \{0, 1\}$, the positive p is sampled from the same class and the negative n from the other class, i.e., $i, p \in D_C$ and $n \in D_C^c$. By enforcing a margin $m > 0$, the encoder learns to keep same-class representations closer than different-class ones by at least m in Euclidean space.

Eq. 5d averages across all sampled triplets $\mathcal{T}$. $\mathcal{T}$ is triplet participating in loss summation, which consists of i, p and n. This stage becomes active once abnormal classes appear, refining the representation space to achieve class-level separability [22].

So the loss function for our contrastive learning presentation can be written as:

$$\mathcal{L} = \begin{cases} \mathcal{L}_{\mathrm{CL}}, & |\mathcal{C}_{\mathrm{obs}}| < 2, \\ \mathcal{L}_{\mathrm{tri}}, & |\mathcal{C}_{\mathrm{obs}}| \geq 2, \end{cases} \tag{6}$$

where C_{obs} is the set of classes observed up to the current time instance.

4) Classification and Evaluation: The downstream task in our setting is binary anomaly detection of AI performance states, where the objective is to distinguish between normal inference (i.e., the AI model delivers predictions within the expected performance range) and abnormal or degraded inference (i.e., the AI model exhibits performance deterioration due to hazardous conditions) in streaming industrial data. After pretraining, the projection head g_ϕ is discarded because it is introduced only to facilitate contrastive learning by shaping the representation space; it does not contribute to the downstream prediction. Instead, a linear classifier is trained on the frozen encoder representation $\mathbf{h}$ to perform the anomaly detection task. This setup evaluates whether the learned representations are sufficiently robust and separable to support reliable classification under limited labeled data. The base learner output $p(y{=}C \mid \mathbf{h}) \in \mathbf{P}^f(\hat{y}_t \mid D_t), C = \{0, 1\}$ thus become the input to the next Section B, where it is further integrated with active learning for efficient annotation. We adopt a multinomial logistic regression (softmax) linear classifier [23] for anomaly detection:

$$\begin{gathered} p(y{=}C \mid \mathbf{h}) = \frac{\exp(\mathbf{w}_C^\top \mathbf{h} + b_C)}{\sum_{r=0}^{1} \exp(\mathbf{w}_r^\top \mathbf{h} + b_r)}, \quad C = \{0, 1\}, \\ \min_{\mathbf{W},\mathbf{b}} \ \mathbb{E}\left[-\sum_{C=0}^{1} y_C \log p(y{=}C \mid \mathbf{h}) \right], \end{gathered} \tag{7}$$

where $\mathbf{W} = [\mathbf{w}_0, \mathbf{w}_1]$ is the weight matrix of the classifier, $\mathbf{b} = [b_0, b_1]$ is the bias vector, and y_C is the one-hot target.

B. Ensemble Active Learning Model

After we obtain the predicted probability $\mathbf{P}^f(\hat{y} \mid D_t)$ through the proposed staged contrastive learning framework, the next challenge is to effectively utilize limited annotation resources in the streaming setting. Although the representation space already improves class separability and robustness, accurate prediction of AI model performance still depends on selectively labeling informative samples under class imbalance and budget constraints. To address this, we integrate an ensemble active learning framework [5] that builds on predicted probability $\mathbf{P}^f$ as the input feature representation

Algorithm 1 Implementation of ConEAL

Input: set of agents $AG_1, AG_2, \ldots, AG_Q$, f_0, D_0, B, S
Initialize: $t = 0$, remaining budget = B, $S = 0$
while remaining budget > 0 **do**
  if $a_t = 1$ **then**
    Acquire y_t, $D_{t+1} = D_t \cup (\mathbf{x}_t, y_t)$
    $S = S + 1$
    B = B - 1
    if $S = po$ **then**
      Train the base learner $f_{t+1} \leftarrow D_{t+1}$
      $S = 0$
    end if
  else
    $D_{t+1} = D_t, f_{t+1} = f_t$
  end if
  Update $AG_1, AG_2, \ldots, AG_Q$, $t = t + 1$
end while
Output: D_{t+1}, f_{t+1}

and adaptively queries samples from the data stream for annotation.

As summarized in Algorithm 1, ConEAL follows the idea of [5] by integrating multiple acquisition agents and adaptively weighting their contributions. The base learner f_0 is first trained by initial dataset D_0. At time t, the incoming sample $\mathbf{x}_t$ is evaluated by a base learner f_t trained with dataset D_t to obtain the predicted label and class probabilities. Each agent AG_i then provides a sample annotation decision to update the prediction model, and these decisions are combined into a weighted ensemble vote. If the sample is selected for annotation, its ground-truth label y_t is acquired, and a reward r_t is computed. The ensemble weights $\alpha_{i,t}$ are updated online using the Exp4.P-EWMA strategy [5], balancing short-term performance with long-term exploration . This design enables ConEAL to leverage the strengths of different agents, improve label efficiency, and achieve higher cumulative rewards than individual acquisition strategies.

1) Reward Function: The reward function is designed to better classify potential valuable samples. At time t, the reward r_t will only be received when ConEAL decides to annotate the samples. Otherwise, the reward will be zero.

We design a bounded, composite reward that integrates multiple components tailored to streaming active learning: (i) it supplies a low-variance global signal by evaluating predictive quality (e.g., Macro-F1 score) on a validation set consisting of the most recent $\mathcal{W}$ annotated samples; (ii) it explicitly incentivizes the discovery of samples from previously unseen classes—particularly first-time abnormal states in the binary classification of normal vs. abnormal inference—thereby preventing the acquisition strategy from collapsing into the majority (i.e., normal) class; (iii) it discourages redundant queries by penalizing instances that the model already predicts with high confidence or correctly, so that annotation resources are concentrated on error-prone regions near the decision boundary; and (iv) it counteracts class imbalance by granting a small bonus to abnormal samples when the estimated abnormal rate is low, improving recall and boundary calibration. This composite design provides a reward signal that is both stable and responsive, making it well-suited for budget-constrained, streaming active learning in industrial AI settings.

The total reward r_t at time t is defined as a bounded weighted sum:

$$r_t = w_{\mathrm{f1}}\, r_t^{\mathrm{f1}} + w_{\mathrm{pred}}\, r_t^{\mathrm{pred}} + w_{\mathrm{exp}}\, r_t^{\mathrm{exp}} + w_{\mathrm{rare}}\, r_t^{\mathrm{rare}}, \quad (8)$$

where $w_{\mathrm{f1}}, w_{\mathrm{pred}}, w_{\mathrm{exp}}, w_{\mathrm{rare}} \geq 0$ and $w_{\mathrm{f1}} + w_{\mathrm{pred}} + w_{\mathrm{exp}} + w_{\mathrm{rare}} = 1$. The weights are treated as tunable hyperparameters, initialized following prior work and further selected via grid search based on empirical performance in our experiments.

The first term supplies a low-variance, global learning signal by evaluating the model on a validation set including the most recent $\mathcal{W}$ annotated samples using the Macro-F1 score. Because Macro-F1 score simultaneously penalizes false positives and false negatives, it is well-suited for imbalanced binary monitoring, where accuracy alone may be misleading [24]. The second component explicitly incentivizes early discovery of previously unseen class occurrences so that the policy does not collapse onto the normal class in the cold-start phase. Implementation maintains a set D_t of observed classes; when $y_t \notin D_t$, a one-time exploration bonus is issued and the class is added to D_t to avoid repeated triggering.

The third sample-level term directs annotation budget toward regions where the model is currently weak:

$$r_t^{\mathrm{pred}} = \begin{cases} \rho^+, & \hat{y}_t \neq y_t, \\ \rho^-, & \hat{y}_t = y_t, \end{cases} \quad (9)$$

where $\rho^+ > 0$ encourages querying informative mistakes and $\rho^- < 0$ discourages redundant annotations on correct predictions. If the current sample has been already predicted correctly or with high certainty, the term contributes a negative value, signaling a low marginal value for querying; if the prediction is incorrect, it contributes a positive value, reflecting expected information gain. Thus, it rapidly sharpens the decision boundary by prioritizing uncertain or error-prone samples while suppressing repeated labels in saturated regions.

For the last term, we maintain an estimate of the abnormal rate $q^+ \in [0, 1]$. When the estimated abnormal rate q^+ is low, those samples typically carry higher operational value and are underrepresented; this term grants a small bonus if the current instance is abnormal and q^+ falls below a threshold τ. These agents supplement the ensemble and allow the overall policy to balance global utility, local exploration, and robustness to imbalance.

2) Sampling Agents: In addition to the reward-driven ensemble policy, ConEAL incorporates two agents: the Low-Density Exploration Agent and the Reinforced Active Learning Agent. Following the design in [5], the low-density exploration agent (LD-agent) [25] promotes sampling in underrepresented regions of the feature space, and the reinforced active learning agent (RAL-agent) [26] adaptively adjusts decision thresholds based on online rewards from the prediction. These

Algorithm 2 Low-density Based Exploration Agent

Input: $\mathbf{x}_t$, $\mathcal{W}$, L, D_0, δ_L
Calculate $\mathrm{Isf}(\mathbf{x}_t)$
for $j = 1, 2, \ldots, L$ **do**
  if $d(\mathbf{x}_i, \mathbf{x}_j) > \mathrm{MaxDist}(\mathbf{x}_j, \mathcal{W})$ **then**
    $\mathrm{MaxDist}(\mathbf{x}_j, \mathcal{W}) \leftarrow d(\mathbf{x}_i, \mathbf{x}_j)$
  end if
end for
Output: Acquisition probability $p_t = \dfrac{\mathrm{Isf}(\mathbf{x}_t)}{L \cdot \delta_L}$
if $|\mathcal{W}| > L$ **then**
  $\mathcal{W} \leftarrow \mathcal{W} \setminus \{\mathbf{x}_{t-L}\}$
end if
$\mathcal{W} \leftarrow \mathcal{W} \cup \{\mathbf{x}_t\}$

two agents complement uncertainty-based strategies by diversifying acquisition decisions and sharpening class boundaries online, which balance the trade-off between exploration and exploitation. We follow the parameterization in [5], where the LD-agent uses a fixed window size and sparsity ratio and the RAL-agent updates a certainty threshold with a small learning rate, and we further adjust these values empirically for stable convergence in streaming experiments.

3) Low-density Based Agent: This agent adopts a density-based criterion and encourages model to explore the input variable space [27]. It will encourage the model to label data around the cluster boundary such that new clusters will be discovered by annotating samples lying in a sparse region with low density. We apply the idea in [25] to build our agent.

Let $\mathcal{W}$ be a first-in–first-out window that stores the L most recent observations, and let $d(\cdot, \cdot)$ denote a distance metric on the sample space. Define the function $\mathrm{MaxDist}$ by $\mathrm{MaxDist}(\mathbf{x}_j, \mathcal{W}) = \max_{\mathbf{x}_\ell \in \mathcal{W}} d(\mathbf{x}_j, \mathbf{x}_\ell)$, i.e., the largest distance from $\mathbf{x}_j$ to any element in $\mathcal{W}$. To approximate the local density around an incoming sample, we quantify its low-density for a candidate point $\mathbf{x}_i$ as the number of items in $\mathcal{W}$ for which $\mathbf{x}_i$ is farther than their current farthest neighbor, as follows:

$$\mathrm{Isf}(\mathbf{x}_i) = \sum_{\mathbf{x}_j \in \mathcal{W}} \mathbb{I}\{\, \mathrm{MaxDist}(\mathbf{x}_j, \mathcal{W}) < d(\mathbf{x}_i, \mathbf{x}_j) \,\}. \quad (10)$$

Algorithm 2 outlines the procedure for preferring points in sparse regions. When a streaming instance $\mathbf{x}_t$ arrives, the low-density agent first computes the local sparsity score $\mathrm{Isf}(\mathbf{x}_t)$ and converts it into the acquisition probability p_t. After making the decision, both the sliding buffer $\mathcal{W}$ and the cached maximum of pairwise distances $\mathrm{MaxDist}(\cdot, \mathcal{W})$ are refreshed. The window length L and the sparsity fraction δ_L are tunable hyperparameters that govern the aggressiveness of exploration and thus the scale of p_t.

4) Reinforced Active Learning Agent: To refine the classifier's decision boundary, the reinforced active learning agent preferentially queries instances whose class assignment is ambiguous [28], [29]. Following the idea in [26], we cast

Algorithm 3 Reinforced Active Learning Agent

Input: $\mathbf{x}_t$, θ_0, η, $penalty$
if $c(\mathbf{x}_t) < \theta_t$ **then**
  $p_t \leftarrow 1$ {acquire the sample and obtain the reward r_t}
  Update the certainty threshold $\theta_{t+1} \leftarrow \min\Big\{\theta_t\big(1 + \eta(1 - 2^{\,r_t/penalty})\big), 1\Big\}$
else
  $p_t \leftarrow 0$
end if
Output: Acquisition probability p_t

TABLE I
EXPERIMENT FACTORS IN MANUFACTURING INDUSTRIAL INTERNET TESTBED DATA [3]

Factors	Definitions	Levels
A	% of Sensors Contaminated or Failed	3
B	S/N ratio	2
C	Distribution Changes of Input Variables	3
D	Balanceness	3
E	# of Singular Pipelines	3
F	# of Fog Node Failures	3
G	# of Communication Channel Failures	3

the acquisition rule as a reinforcement-learning (RL) control problem: the policy is an adaptive threshold on predictive certainty. An RL controller adjusts a certainty threshold θ by leveraging feedback from past acquisition outcomes. Specifically, when a sample $\mathbf{x}_t$ arrives, the base learner f_t produces $c(\mathbf{x}_t) = \max_k \mathbf{P}^f(\hat{y} = k \mid \mathbf{x}_t)$, and the agent requests a label if $c(\mathbf{x}_t) < \theta_t$; otherwise it skips. If a query is made, a scalar reward r_t is observed, after which the threshold is updated as:

$$\theta_{t+1} = \min\Big\{\ \theta_t\Big(1 + \eta \cdot \big(1 - 2^{\,r_t/penalty}\big)\Big),\ 1\Big\}. \quad (11)$$

Algorithm 3 presents the procedure for the reinforcement-driven active learning agent. When a positive reward is observed, it produces a policy that adapts to the decision boundary learned by the base learner.

IV. CASE STUDY

We evaluate the efficiency and effectiveness of ConEAL based on a dataset from AI inference tasks in a Fog–Cloud MII testbed [3]. The infrastructure consists of five Raspberry Pi fog nodes and a centralized Cloud server that functions both as an orchestrator and as an additional computation node. From the collected runtime and performance metrics, ConEAL derives feature representations and addresses a classification task aimed at identifying whether the system is in a normal or abnormal operational state. The experiment is implemented on a PC with 2.30 GHz Intel Core i7-11800 Processor with 32.0 GB RAM and Windows 11.

A. Data Description

The dataset is generated from the MII testbed [3], where a library of 256 DNN computation pipelines and their parameter-

TABLE II
MANUFACTURING INDUSTRIAL INTERNET TESTBED DATA DESCRIPTION [3]

Data Type	Sample Size	Remark
Normal(Class 0)	4676	If all factors A-E are 0, the AI model performance is labeled as "normal".
Abnormal(Class 1)	291	If any of A-E is 1 or 2, the AI model performance is labeled as "abnormal".
Non-Executable	2323	The AI pipelines cannot be executed thus these samples are excluded from training and evaluation.
Total	7290	

perturbed variants are deployed across Cloud and Fog nodes. Experimental conditions are parameterized by seven factors (A–G) as shown in Table I, which control data hazards (A–D), AI pipeline singularity (E), and cyber-layer failures (F–G). Full combinations yield 7,290 treatment–machine tasks [3].

Tasks affected by Fog node failures (F>0) or communication failures (G>0) are excluded, since inference tasks will not be carried out and AI model performance is not collected. After data pre-processing, we retain 4,967 executable tasks or samples (*E*), on which inference results and runtime metrics are collected. Within *E*, samples are labeled as "normal" (Factors A-E are normal) or "abnormal" (one or more of factors A–E are abnormal), as summarized in Table II. An additional 2,323 Non-Executable (*NE*) cases are excluded from training and evaluation for ConEAL.

This setting naturally defines a binary classification problem. At time t, the input consists of the feature vector $\mathbf{x}_t$ from the incoming sample and the labeled data pool $D_t = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{m_t}, y_{m_t})$. The base learner f_t encodes $\mathbf{x}_1, \ldots, \mathbf{x}_{m_t}$ into a representation space $\mathbf{h}$, which is further used to fine-tune a linear classification head with the corresponding labels $y_1, \ldots, y_{m_t}$. The response variable is the system state label at time t, with $y_t \in \{\textit{normal}, \textit{abnormal}\}$. Then f_t is validated on a separate test set to compute performance metrics, and the reward r_t is derived from these evaluations. The reward then updates agent weights and guides subsequent sampling decisions, ensuring that informative abnormal samples are efficiently acquired under streaming and imbalanced conditions. By prioritizing the most informative samples, the system can achieve reliable anomaly detection with reduced annotation cost, thereby enhancing the resilience of AI inference in Fog–Cloud manufacturing environments.

B. ConEAL Experiments

In this section, we will evaluate performance of the proposed ConEAL method on the dataset described in last section and compare it with other benchmark methods. Due to the imbalance of the dataset, we use Macro-F1 score [30] as an indicator to measure the performance of each method. For a V-class classification problem, the F1 score of class v is defined in a one-vs-rest manner, i.e., by treating class v as the positive class and all other classes as negative. Formally, the F1 score for class v is

$$\text{F1}_{(v)} = \frac{2 \cdot \text{Precision}_{(v)} \cdot \text{Recall}_{(v)}}{\text{Precision}_{(v)} + \text{Recall}_{(v)}}, \tag{12}$$

where

$$\text{Precision}_{(v)} = \frac{TP_{(v)}}{TP_{(v)} + FP_{(v)}}, \tag{13}$$

$$\text{Recall}_{(v)} = \frac{TP_{(v)}}{TP_{(v)} + FN_{(v)}}. \tag{14}$$

Here, $TP_{(v)}$, $FP_{(v)}$, and $FN_{(v)}$ denote the true positives, false positives, and false negatives for class v, respectively, under the one-vs-rest evaluation.

The Macro-F1 score is then obtained by averaging the F1 scores across all classes:

$$\text{Macro-F1} = \frac{1}{V} \sum_{v=1}^{V} \text{F1}_{(v)}. \tag{15}$$

In the binary case considered in this work ($V = 2$), $\text{F1}_{(1)}$ corresponds to the abnormal class and $\text{F1}_{(0)}$ corresponds to the normal class, and the Macro-F1 is calculated as their simple average so it can better reflect the quality of small categories when the categories are unbalanced.

To emulate online prediction and annotation in streaming industrial data, the dataset is first subdivided into three subsets: the initial training set D_0, the streaming training set, and the testing set. At time t, ConEAL adaptively queries labels from the unlabeled stream under a total annotation budget B, and the base learner f_t is updated online. To build the base learner, we use the softmax linear classifier defined in Eq. (7), trained on the frozen encoder representation $\mathbf{h}$ obtained from the two-stage contrastive pretraining. Model performance is evaluated dynamically during online updates. At time t, a validation set is constructed by selecting the most recent $\mathcal{W}$ annotated samples from the labeled pool D_t, which serves as a sliding evaluation window. This mechanism approximates the model's generalization ability without requiring a separate held-out set when online updating. The feature space is derived from raw multivariate time series collected in the testbed and tabular outputs describing the learning performance of deployed AI pipelines. The raw multivariate time series data were converted into statistical descriptors (e.g., mean, standard deviation, skewness, kurtosis). Combined with tabular outputs, these data consist of streaming industrial data. After standardization, missing-value imputation, and outlier filtering, 32 features are retained as model input (i.e., $\mathbf{x}_t$). To preserve the streaming nature of the task, the chronological order of samples (i.e., the execution order of the inference tasks via the AI system) is maintained. The testing set is held out at the task level and remains fixed throughout the experiments, providing unbiased

TABLE III
THE PREDICTION OF AI MODEL PERFORMANCE OVER 10 REPLICATIONS

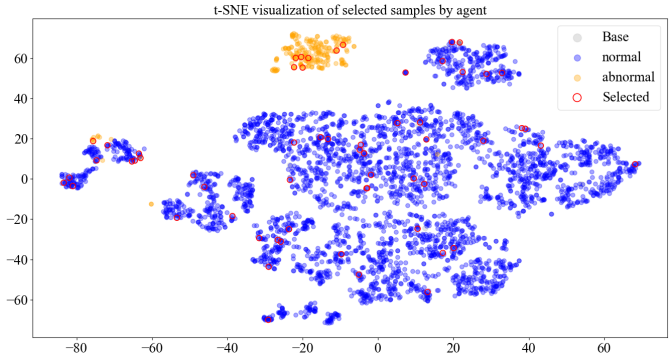
Method	Training Sample Size (Percentage in All Samples)	Average Macro-F1 Score (Standard Error)
RS	87.9(2.21%)	0.93(0.09)
US	90(2.26%)	0.94(0.16)
QBC-PYP	90(2.26%)	0.87(0.14)
DBALStream	90(2.26%)	0.96(0.08)
ConEAL(Proposed)	**51.4(1.29%)**	**0.99(0.11)**
All Training Data	3975(100%)	1

performance metrics for comparison across methods. This setup captures the challenges of imbalanced, temporally correlated data streams that are common in manufacturing quality inspection, and provides a realistic scenario for evaluating ConEAL.

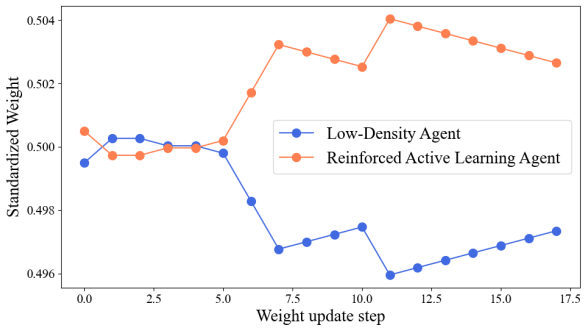
Four benchmark methods are used for comparison with the proposed ConEAL framework. Random Sampling (RS) [31] explores the input space by acquiring samples uniformly at random, without considering model confidence or class distribution; it serves as a naive baseline emphasizing pure exploration. Uncertainty Sampling (US) [32] instead exploits the current model by querying labels for instances where the prediction confidence is lowest (e.g., samples closest to the decision boundary), thereby refining the classifier in regions of ambiguity. DBALStream [25] and QBC-PYP [33] are two state-of-the-art composite active learning strategies that combine exploration and exploitation: DBALStream balances uncertainty-based selection with density-aware exploration of the feature space, while QBC-PYP leverages model diversity to identify informative samples that reduce disagreement among committee members.

Table III shows the average Macro-F1 score and sample sizes of different methods after selecting a subset of samples for annotation, over 10 replications. In each replication, 20% samples are randomly selected from the whole dataset as the testing set for Macro-F1 score evaluation, with the remaining for training. Based on the suggestion in [5], [34], [35] and grid search, we set the following values: weights in the reward function as w_{f1} =0.2, w_{pred} =0.5,w_{exp} =0.15, w_{rare} =0.15, respectively; the first 10 samples based on the time of execution in the training set will be used for model pretraining (i.e., the base dataset) and the sampling budget is set as 80.

As shown in Table III, the proposed ConEAL method provides higher or tied-best testing Macro-F1 scores while requiring fewer labeled samples than other benchmark methods. This improvement can be attributed to the integration of contrastive representation learning with ensemble active learning. In particular, contrastive learning constructs an invariant and discriminative feature space under one-class initialization, allowing the model to incorporate abnormal samples as they emerge and maintain robustness under time-varying distributions. Combined with the ensemble active learning policy, which adaptively balances exploration and exploitation under severe class imbalance, ConEAL is able to identify informative samples more effectively and thus achieve superior label efficiency compared with traditional annotation strategies.



(a) Visualization of the samples selected by agents in ConEAL-EN2



(b) The weight trend during the sampling process

Fig. 3. Results of ConEAL under certain learning scenario.

Notably, ConEAL reaches its peak Macro-F1 well before the annotation budget is exhausted, leaving a substantial portion of the budget unused; in contrast, most baselines require nearly the full budget to approach their best scores, evidencing superior label efficiency. In addition, the standard error of our method is on par with that of DBALStream, and the overall variability of the prediction performance is manageable. RS and single-point uncertainty-based US tend to overly favor the dominant class under severe imbalance.

As shown in Fig. 3a, a t-SNE [36] projection is used to visualize the feature representations of executable tasks, where blue and orange points denote normal and abnormal samples, respectively, and red markers indicate the instances selected for annotation by ConEAL-EN2. The visualization demonstrates that the selected samples are not confined to a single region but are distributed across multiple clusters,

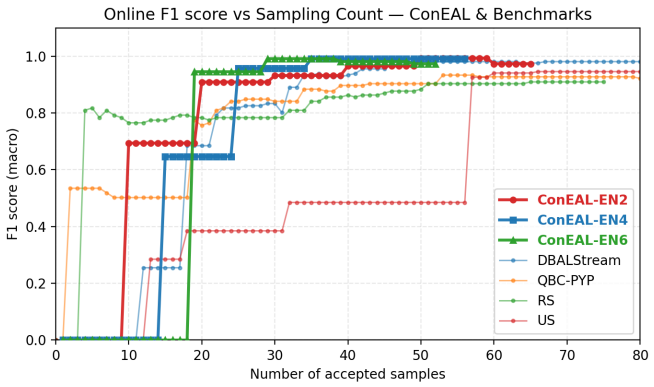


Fig. 4. Online Macro-F1 over accepted queries — ConEAL & benchmarks

reflecting the ability of the agents to explore diverse areas of the feature space. In particular, the sampling density is higher near abnormal clusters, suggesting that ConEAL effectively prioritizes minority-class instances. This behavior highlights ConEAL's capability to balance exploration of the overall data distribution with targeted exploitation near decision boundaries, thereby improving anomaly detection performance under class imbalance.

In Fig. 3b, we visualize the standardized weights of the Low-Density Exploration Agent and the Reinforced Active Learning Agent evolve differently over the update steps. Initially, both agents start with comparable influence, but with increasing update steps the Reinforced Active Learning Agent gains higher weight, shifting the strategy toward exploitation while maintaining residual exploration [5]. This adaptive balance explains the stable and efficient online annotation achieved by ConEAL. In Fig. 4, ConEAL variants, including using two agents (-EN2), four agents (-EN4), and six agents (-EN6), achieve higher Macro-F1 with fewer labeled samples than other benchmark methods. All ConEAL variants converge to 0.9 or higher within about 25 samples, whereas most benchmark methods converge slower. These results indicate superior sample efficiency of ConEAL for online learning. Therefore, the proposed method not only reduces the annotation efforts but also contributes to the prediction of AI model performance.

In summary, ConEAL performs active learning that explicitly balances exploration and exploitation to select the most informative and representative samples for annotation. This targeted acquisition improves data quality and computational efficiency, addressing an essential challenge in AI model performance prediction to achieve proactive-adaptive resilient AI system.

V. Conclusion

DNNs are widely used to improve manufacturing performance via computation services in MII. However, the performance of DNNs at the inference stage is frequently affected by data quality, computation pipeline, and cyber-physical layer of manufacturing. The capability to predict the AI model performance at the inference stage is the foundation for proactive-adaptive resilient AI models. However, there are significant challenges on extreme imbalanced samples of abnormal AI model performance, time-varying distributions, and one-class initialization. This paper proposed a contrastive learning-based framework for AI model performance anomaly prediction. By combining contrastive representation learning and ensemble active learning, ConEAL can detect subtle abnormal states with limited labeled data. The method effectively balances prediction performance and annotation cost in real-world streaming data scenarios. Experimental results on a resilient AI experiment in a Fog-Cloud testbed demonstrate that ConEAL outperforms existing benchmark methods with faster learning efficiency.

The paper provides a few future research directions. First, we aim to generalize the framework to handle multi-label anomaly prediction tasks [37], where an AI system may be simultaneously affected by multiple abnormal root causes, as defined in Table I. Second, we will investigate the integration of ConEAL with multi-modal, incompatible data streams for online contrastive learning strategies to improve adaptability across changing environments and time-varying distributions [38]. Third, an adaptive adjustment strategy will be explored to ensure uninterupted AI model performance in online settings, based on the prediction of abnormal AI model performance [39].

Acknowledgment

The authors acknowledge the funding support from National Science Foundation, CMMI-2331985.

References

[1] Y. Lu, X. Xu, and L. Wang, "Smart manufacturing process and system automation–a critical review of the standards and enabling technologies," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1443–1457, 2022.

[2] R. Zhao, D. Sahoo, and S. C. Hoi, "Online continual learning for adaptive ai systems," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 7. AAAI Press, 2022, pp. 7452–7460.

[3] Y. Zeng, I. Lourentzou, X. Deng, and R. Jin, "Fair: Facilitating artificial intelligence resilience in manufacturing industrial internet," *arXiv preprint arXiv:2503.01086*, 2025.

[4] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PmLR, 2020, pp. 1597–1607.

[5] Y. Zeng, X. Chen, and R. Jin, "Ensemble active learning by contextual bandits for ai incubation in manufacturing," *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 1, pp. 1–26, 2023.

[6] Y. Zeng, P. Shojaee, S. H. A. Faruqui, A. Alaeddini, and R. Jin, "Contextual bandit guided data farming for deep neural networks in manufacturing industrial internet," in *2022 IEEE 5th International Conference on Industrial Cyber-Physical Systems (ICPS)*. IEEE, 2022, pp. 1–6.

[7] A. K. Kalusivalingam, A. Sharma, N. Patel, and V. Singh, "Enhancing predictive maintenance in manufacturing using machine learning algorithms and iot-driven data analytics," *International Journal of AI and ML*, vol. 1, no. 3, 2020.

[8] R. Gao, J. Krüger, M. Merklein, H. Möhring, and J. Váncza, "Artificial intelligence in manufacturing: State of the art, perspectives, and future directions," *CIRP Annals*, vol. 73, no. 2, pp. 723–749, 2024.

[9] G. Abdiyeva-Aliyeva and M. Hematyar, "Ai-based network security anomaly prediction and detection in future network," in *The International Conference on Artificial Intelligence and Applied Mathematics in Engineering*. Springer, 2022, pp. 149–159.

[10] M. Abououf, S. Singh, R. Mizouni, and H. Otrok, "Explainable ai for event and anomaly detection and classification in healthcare monitoring systems," *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 3446–3457, 2023.

[11] P. Dixit, P. Bhattacharya, S. Tanwar, and R. Gupta, "Anomaly detection in autonomous electric vehicles using ai techniques: A comprehensive survey," *Expert Systems*, vol. 39, no. 5, p. e12754, 2022.

[12] P. Schummer, A. del Rio, J. Serrano, D. Jimenez, G. Sánchez, and Llorente, "Machine learning-based network anomaly detection: Design, implementation, and evaluation," *AI*, vol. 5, no. 4, pp. 2967–2983, 2024.

[13] S. Holly, R. Heel, D. Katic, L. Schoeffl, A. Stiftinger, P. Holzner, T. Kaufmann, B. Haslhofer, D. Schall, C. Heitzinger, and J. Kemnitz, "Autoencoder based anomaly detection and explained fault localization in industrial cooling systems," *arXiv preprint arXiv:2210.08011*, 2022.

[14] Y. Li, G. Qian, X. Jiang, Z. Jiang, W. Wen, S. Zhang, K. Li, and Q. Lao, "Hierarchical-instance contrastive learning for minority detection on imbalanced medical datasets," *IEEE Transactions on Medical Imaging*, vol. 43, no. 1, pp. 416–426, 2023.

[15] V. Sharmanska, L. A. Hendricks, T. Darrell, and N. Quadrianto, "Contrastive examples for addressing the tyranny of the majority," *arXiv preprint arXiv:2004.06524*, 2020.

[16] A. Shen, X. Han, T. Cohn, T. Baldwin, and L. Frermann, "Contrastive learning for fair representations," *arXiv preprint arXiv:2109.10645*, 2021.

[17] Y. Marrakchi, O. Makansi, and T. Brox, "Fighting class imbalance with contrastive learning," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2021, pp. 466–476.

[18] Q. Feng, L. Xie, S. Fang, and T. Lin, "Bacon: Boosting imbalanced semi-supervised learning via balanced feature-level contrastive learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 11, 2024, pp. 11 970–11 978.

[19] Z. Zhong, J. Cui, E. Lo, Z. Li, J. Sun, and J. Jia, "Rebalanced siamese contrastive mining for long-tailed recognition," *arXiv preprint arXiv:2203.11506*, 2022.

[20] M. W. Gardner and S. R. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmospheric environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.

[21] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," *Advances in neural information processing systems*, vol. 29, 2016.

[22] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[23] C. Kwak and A. Clayton-Matthews, "Multinomial logistic regression," *Nursing research*, vol. 51, no. 6, pp. 404–410, 2002.

[24] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC genomics*, vol. 21, no. 1, p. 6, 2020.

[25] D. Ienco, I. Žliobaitė, and B. Pfahringer, "High density-focused uncertainty sampling for active learning over evolving stream data," in *Proceedings of the 3rd international workshop on big data, streams and heterogeneous source mining: algorithms, systems, programming models and applications*. PMLR, 2014, pp. 133–148.

[26] S. Wassermann, T. Cuvelier, and P. Casas, "Ral-improving stream-based active learning by reinforcement learning," in *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD) Workshop on Interactive Adaptive Learning (IAL)*, 2019.

[27] S. Alizadeh-Fanalou, A. Nazarizadeh, F. Alian, P. Faraji, B. Sorori, and M. Khosravi, "Small dense low-density lipoprotein-lowering agents," *Biological chemistry*, vol. 401, no. 10, pp. 1101–1121, 2020.

[28] M. Dodds, J. Guo, T. Löhr, A. Tibo, O. Engkvist, and J. P. Janet, "Sample efficient reinforcement learning with active learning for molecular design," *Chemical Science*, vol. 15, no. 11, pp. 4146–4160, 2024.

[29] M. Fang, Y. Li, and T. Cohn, "Learning how to active learn: A deep reinforcement learning approach," *arXiv preprint arXiv:1708.02383*, 2017.

[30] J. Opitz and S. Burst, "Macro f1 and macro f1," *arXiv preprint arXiv:1911.03347*, 2019.

[31] S. Noor, O. Tajik, and J. Golzar, "Simple random sampling," *International Journal of Education & Language Studies*, vol. 1, no. 2, pp. 78–82, 2022.

[32] D. D. Lewis, "A sequential algorithm for training text classifiers: Corrigendum and additional data," in *Acm Sigir Forum*, vol. 29, no. 2. ACM New York, NY, USA, 1995, pp. 13–19.

[33] C. C. Loy, T. M. Hospedales, T. Xiang, and S. Gong, "Stream-based joint exploration-exploitation active learning," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1560–1567.

[34] J. Eschmann, "Reward function design in reinforcement learning," *Reinforcement learning algorithms: Analysis and Applications*, pp. 25–33, 2021.

[35] G. Hacohen and D. Weinshall, "How to select which active learning strategy is best suited for your specific problem and budget," *Advances in Neural Information Processing Systems*, vol. 36, pp. 13 395–13 407, 2023.

[36] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[37] W. Zhang, D. K. Jha, E. Laftchiev, and D. Nikovski, "Multi-label prediction in time series data using deep neural networks," *arXiv preprint arXiv:2001.10098*, 2020.

[38] S. Mai, Y. Zeng, S. Zheng, and H. Hu, "Hybrid contrastive learning of tri-modal representation for multimodal sentiment analysis," *IEEE Transactions on Affective Computing*, vol. 14, no. 3, pp. 2276–2289, 2022.

[39] Y. Liu, Z. Yang, X. Wu, L. Lan, F. Lin, H. Su, and J. Huang, "Adaptive threshold adjustment strategy based on fuzzy logic control for ground energy storage system in urban rail transit," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 10, pp. 9945–9956, 2021.