Design, Deployment, and Evaluation of an Industrial AIoT System for Quality Control at HP Factories

DUC VAN LE*, HP-NTU Digital Manufacturing Corporate Lab, Nanyang Technological University
JOY QIPING YANG*†, National University of Singapore, Singapore
SIYUAN ZHOU, HP-NTU Digital Manufacturing Corporate Lab, Nanyang Technological University
DAREN HO, HP Inc., Singapore
RUI TAN, School of Computer Science and Engineering, Nanyang Technological University, Singapore

Enabled by the increasingly available embedded hardware accelerators, the capability of executing advanced machine learning models at the edge of the Internet of Things (IoT) triggers interest of applying Artificial Intelligence of Things (AIoT) systems for industrial applications. The *in situ* inference and decision made based on the sensor data allow the industrial system to address a variety of heterogeneous, local-area non-trivial problems in the last hop of the IoT networks. Such a scheme avoids the wireless bandwidth bottleneck and unreliability issues, as well as the cumbersome cloud. However, the literature still lacks presentations of industrial AIoT system developments that provide insights into the challenges and offer lessons for the relevant research and industry communities. In light of this, we present the design, deployment, and evaluation of an industrial AIoT system for improving the quality control of HP Inc.'s ink cartridge manufacturing lines. While our development has obtained promising results, we also discuss the lessons learned from the whole course of the work, which could be useful to the developments of other industrial AIoT systems for quality control in manufacturing.

CCS Concepts: • **Computer systems organization → Sensor networks**; • **Computing methodologies → Machine learning algorithms**.

Additional Key Words and Phrases: Industrial AIoT, Quality Control, Smart Manufacturing

ACM Reference Format:
Duc Van Le, Joy Qiping Yang, Siyuan Zhou, Daren Ho, and Rui Tan. 2023. Design, Deployment, and Evaluation of an Industrial AIoT System for Quality Control at HP Factories. *ACM Trans. Sensor Netw.* 1, 1, Article 1 (August 2023), 19 pages. https://doi.org/XXXXXXX

1 INTRODUCTION

The recent advances of machine learning (ML) in dealing with sophisticated data patterns and the increasingly available embedded hardware for accelerating ML trigger the interest of studying and implementing industrial Artificial Intelligence of Things (AIoT) [6] that integrates artificial

*The first two authors contributed equally to this research.
†This work was completed while Joy Qiping Yang was with HP-NTU Digital Manufacturing Corporate Lab, Nanyang Technological University.

A preliminary version of this work appeared in The 18th Annual IEEE International Conference on Sensing, Communication and Networking (SECON) held in Virtual Event, 6-9 July 2021 [18].
Authors' addresses: Duc Van Le, Siyuan Zhou, HP-NTU Digital Manufacturing Corporate Lab, Nanyang Technological University; Joy Qiping Yang, National University of Singapore; Daren Ho, HP Inc.; Rui Tan, School of Computer Science and Engineering, Nanyang Technological University.

1550-4859/2023/8-ART1 $15.00
https://doi.org/XXXXXXX

intelligence (AI) with the Internet of Things (IoT) edge. The AIoT systems have distributed, *in situ* inference and decision capabilities to avoid the handicaps encountered when transmitting data to remote central servers for decision making. However, there is no one-size-fits-all AIoT system that can be used for all industrial applications. The designs and implementations of the AIoT systems in general need to be highly customized based on the specific objectives, operational procedures, and practical constraints of the industrial processes. Many task-specific designs such as the configuration and training of the used ML models still require substantial work to achieve the objectives. The main challenges often come from the deviations of the real-world conditions from the assumptions made by the relevant research. Specifically, the relevant research in general needs a set of clearly defined assumptions to render a satisfactory level of rigor in addressing a specific problem while isolating other problems, but real-world tasks in industrial practices face many coupled problems. Therefore, the design of a working industrial AIoT system requires holistic considerations with many inputs from the domain experts and technicians.

Despite the heterogeneity of industrial AIoT systems, the systematic description of an effort that designs and implements an AIoT system for a specific industrial application can provide insights into understanding the potential challenges that would be faced by other AIoT system designs. In this technical note, we present our recent work of designing an AIoT-based quality control (QC) system that provides an essential function to maintain high-quality products in the manufacturing systems. Specifically, the system aims at improving the QC of the ink cartridge manufacturing lines at the factories of HP Inc. (referred to as HP for short in this technical note). This development includes the key elements of AIoT, including sensing, data analytics, design and deployment of embedded ML models at the IoT computing edge, as well as decision support with associated reasoning for machine health prognostics. We present the motivation, the details of our system design, and the experiences learned from this work that can be useful to the design and implementation of other industrial AIoT systems.

Our target application is HP's ink extraction testing (IET), which is a destructive and accelerated testing on randomly selected samples of the manufactured ink cartridges. It is the final QC procedure which aims at detecting any defective batch in which the ink cartridges' performance deviates from the specification. In particular, the IET machine (referred to as *tester* for short in this technical note) extracts the ink from the tested cartridge at a prescribed rate, which is much faster than those on printers, and records the liquid pressure of the ink throughout the course. The *profile curve* of the liquid pressure versus the volume of the extracted ink provides rich information regarding the performance of the tested ink cartridge. Thus, the match between the recorded profile and a preset template profile is the main criterion to pass the test. The alarms due to detected mismatch are further classified manually by trained technicians. Depending on the manual classification results, further QC actions will be taken. Although IET is critical to all HP's ink cartridge manufacturing lines, the factories' current IET procedure faces two main challenges as follows.

First, it is desirable to solidify the technicians' experience-based approach of manually classifying alarms as a computable classifier for the purpose of QC consistency and knowledge transfer. However, the pressure profiles exhibit a significant degree of variability and the technicians' manual classification incorporates extensive domain knowledge regarding the internals of the ink cartridges, which may be descriptive and not quantifiable. The attempt of converting the manual classification approach into a computable rule-based classifier results in many questions of how to properly define the features, configure the rules, and set the thresholds.

Second, the operations of the tester inevitably introduce uncertainties that result in false alarms. For example, from the technicians' experiences, formation of air bubbles in the tester's ink tubes is one of the major factors causing false alarms, because a bubble with a sufficiently large volume affects the liquid pressure measurement. Performing a tube flush before each test can largely resolve

the issue, but it significantly reduces the testing throughput. From the historical records, the overall alarm rate of the deployed testers is about 30 times of the defect rate of the manufactured ink cartridges, suggesting most alarms are false. For quality assurance, upon any alarm, the factories' current practice is to flush the tester's tube and perform the destructive test on an additional ink cartridge sample to reconfirm the technician's manual classification result. Thus, it is desirable to have an approach that can reliably identify the false alarms and avoid the unnecessary tests.

To address the above two challenges, we designed and implemented an AIoT system that classifies the tester's alarms into product-induced (i.e., true alarms) and tester-induced (i.e., false alarms). The primary design goal is to achieve high recall and precision in identifying the product-induced and tester-induced alarms. Specifically, our AIoT system has four main components. First, the *ML-based profile classifier* captures the product engineers' experiences in classifying the alarms. Second, we develop a heuristic-based anomaly detection (AD) approach that classifies the pressure profiles based on domain knowledge on the patterns contained in the profiles. Third, based on a key observation that the air bubbles are often formed at the joint of the tester's ink tubes, we deploy a *smart camera* at the joint and design convolutional neural network (CNN) and computer vision algorithms that run on the camera to detect and estimate the presence and volume of air bubbles. Fourth, we develop a tester assessment approach that applies statistical learning to estimate the probability that a tester is faulty based on the historical alarm classification results. The outcome supports the decision process of whether maintenance activities should be performed for the concerned tester.

We have deployed our AIoT system in HP's manufacturing lines. Through controlled experiments, our heuristic-based AD approach achieves a recall of 95.2% in detecting the defective ink cartridges. Moreover, the smart camera can correctly detect the presence of air bubbles in 94% of the testing images. In summary, this technical note presents the design and evaluation processes of the AIoT system, discusses the key experiences and lessons learned from the whole course of the work, which can be useful to the developments of other industrial AIoT systems.

The remainder of this technical note is organized as follows. §2 reviews related work. §3 presents the background about IET and overviews our AIoT system. §4, §5, and §6 present the designs of ML-based profile classifiers, heuristic-based AD approach, and smart camera, respectively. §7 presents deployment and evaluation of the system integrating the components in §4, §5, and §6. §8 presents the statistical learning-based tester assessment. §9 discusses the experiences and learned lessons. §10 concludes this technical note.

2 RELATED WORK

Challenges in deploying ML and AIoT in Industries: Industrial AIoT is the combination of AI and industrial IoT to improve the level of automation in analyzing and creating useful insights from the industrial sensor data [12]. Deploying an industrial AIoT system often faces challenges of making decision on the design and implementation of IoT hardware infrastructures (e.g., edge, fog, and cloud) and software components (e.g., ML models) based on the specific objectives and practical constraints of the industrial processes. A number of studies [1, 2, 7–9] have investigated practical challenges and provided some insights on deploying industrial AIoT systems. Alkhabbas *et al.* [1] conduct a survey that distributes a questionnaire containing 14 questions about the deployment decisions of IoT systems. Their findings based on the responses of 66 IoT system designers from 18 countries show that the reliability, performance, security, and cost are the four main factors affecting the designer's decisions on deploying IoT systems. The studies [2, 7–9] discuss practical challenges and lessons learned from deploying ML algorithms for various applications. For instance, with experiences in designing analytics platforms at Twitter, Lin and Ryaboy [9] observe that at the first step, the data scientists often spend many efforts in understanding and cleansing the

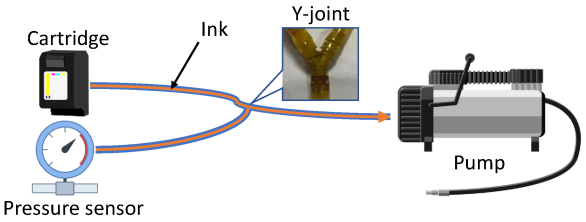


Fig. 1. Illustration of testing a cartridge in IET machines.

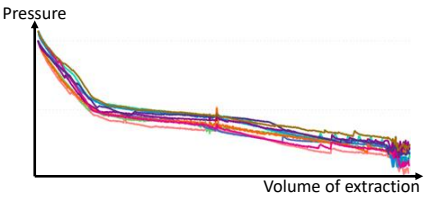


Fig. 2. Samples of measured profiles.

collected data before they can design ML models. Budd *et al.* [2] identify that the lacking of training data labels is a key challenge of designing ML models for medical image analysis. As presented in [7], practical ML systems often employ simple ML models such as random forests, decision trees, and shallow neural networks to shorten the deployment time and gain better interpretability. For instance, Haldar *et al.* [7] report that in the process of applying deep ML models for AirBnB search, after several unsuccessful attempts with complex neural networks, they finally deployed a simple neural network model to simplify the deployment process while providing reasonably good performance. In addition, Hazelwood *et al.* [8] discuss several key factors that drive the decisions on designing ML models for data center infrastructures at Facebook. Similar to the above studies, this technical note presents our experiences and lessons learned from the design and implementation of an industrial AIoT system. As our work considers different specific objectives, operational procedures, and practical constraints, this technical note provides new insights.

QC in production processes: QC is a set of procedures for determining whether a product meets a predefined set of quality criteria or the customer's requirements [16]. It also provides the information to determine the need for corrective actions in the manufacturing process. AIoT technologies have been adopted to improve QC of manufacturing lines. For instance, at Siemens' electronics plant in Amberg, Germany [14], various ML models and edge computing are used to design a predictive model-based QC framework for testing the quality of printed circuit boards (PCBs). The framework helps improve the recall in detecting defective PCBs and reduce testing overheads. In this technical note, we present the work to develop an industrial AIoT system for improving the QC of the ink cartridge manufacturing lines at the HP's factories.

Our prior work [18] has presented the design of the first three components of the developed AIoT system, i.e., ML-based profile classifiers, heuristic-based AD approach, and smart camera. Based on [18], we make the following new contributions in this paper. First, §4.2 presents a new profile classification approach based on ensemble learning and §4.3 presents a new set of experiments driven by historical data to evaluate all the ML-based profile classifiers incorporated with resampling for addressing the data imbalance issue. Second, §8 presents the fourth newly designed component of statistical learning-based tester assessment approach and the related evaluation.

3 BACKGROUND, MOTIVATION, & SYSTEM OVERVIEW

In this section, we present the background of the ink extraction testing (IET) and discuss its current problems in practice. Then, we overview the design of our AIoT system for improving the IET.

3.1 IET Background and Problem Statement

As discussed in §1, the IET is the final QC process of the ink cartridge manufacturing. Specifically, a number of randomly selected ink cartridge samples are tested using the tester. The tester can run six ink cartridges simultaneously. Fig. 1 illustrates how the tubes connect a tested ink cartridge, a stepper motor pump, and a pressure sensor. A transparent plastic Y-joint is used to join the tubes.

A workstation computer of the tester controls the stepper motor pump to extract ink from the ink cartridge at a steady volume rate for a certain time duration. Meanwhile, a liquid pressure sensor continuously measures the pressure in the tube and reports the readings to the workstation computer. The resulting curve of the measured liquid pressure versus the volume of the extracted ink is a profile of the tested ink cartridge. The ink cartridges of different models have distinct profiles. Fig. 2 shows profile samples of a certain ink cartridge model.

The tester adopts a *bound-based detector* to assess a measured profile against a *template profile* with an upper bound and a lower bound. The template profile is defined based on the specification of the ink cartridge. The bound-based detector classifies a profile *normal* if the profile completely lies within the belt area between the two bounds; otherwise, the tester classifies the profile *abnormal*. To achieve high recall in capturing defective cartridges, the factories' current practice is to impose stringent bounds. As a result, the tester generates alarms frequently. As mentioned in §1, many alarms are actually false. This is because that the pressure measurements can be noisy and biased.

Specifically, the pressure sensing is subject to both endogenous and exogenous noises. Endogenous noises are mainly from the thermal noises of the pressure sensor and the random control errors of the stepper motor pump. Exogenous noises are mainly caused by vibrations and blockage of the ink tubes. The vibration is caused by the movements of nearby human operators and bulky manufacturing machines, while the blockage is caused by the hardening ink residue trapped within the tube. In addition, the tester is subject to the following biases. An improper manual insertion of the tested ink cartridge onto the tester may cause loss of back pressure of the cartridge and deviation from the template profile. An air bubble formed in the tester's ink tubes with a sufficiently large volume can also affect the pressure sensing.

In the current protocol of the factories, the alarm-triggering profiles will be further classified manually by the technicians into false positives (i.e., tester-induced) and true positives (i.e., product-induced). The manual classifications are based on the technicians' knowledge received during training and also their own experiences. As such, the classification results may lack high confidence and consistency. To ensure that there is no doubt regarding the QC result of a tested batch, the technicians may need to perform maintenance of the tester and conduct destructive tests with additional samples. A common maintenance performed is to flush the tubes with water to purge out ink and air bubbles at the end of every test. However, the frequent maintenance reduces the IET throughput significantly; the additional destructive tests increase the cost. Therefore, it is desirable to develop a system that can reliably and consistently classify the alarms generated by the bound-based detector, such that all or part of the unnecessary tester maintenance and additional destructive tests can be avoided.

3.2 AIoT System Overview

In this work, we follow the *progressive system development methodology* to design and implement an AIoT system to replace the factories' current practice of manually classifying the alarm-triggering profiles into normal and abnormal profiles. During the whole course of designing our AIoT system, we have developed four main components as follows.

(1) ML-based profile classifiers: We design and train several ML-based classifiers to classify the profiles. The training processes are based on historical profiles labeled by the product engineers. Specifically, we design multiple classifiers based on supervised, semi-supervised, and unsupervised ML models. Each classifier takes different features as input to classify a profile. Ensemble methods are also used to integrate the results of the multiple classifiers.

(2) Heuristic-based anomaly detection: The ML-based classifiers face challenges of limited and imbalanced training dataset. Thus, we also develop a heuristic approach which considers the profile classification as an anomaly detection (AD) problem. The profiles of good ink cartridges,

Table 1. Accuracy of ML-based classifiers over 134 historical profile samples. Each table entry includes average and standard deviation of accuracy results over 10 sub-datasets.

Evaluation metrics	Classifiers				Ensemble	
	CNN	DT	MVAE	k-means	Veto	Majority
Accuracy	0.9 ± 0.03	0.89 ± 0.07	0.62 ± 0.34	0.28 ± 0.12	0.92 ± 0.002	0.91 ± 0.04
Abnormal recall	0.97 ± 0.03	0.94 ± 0.07	0.64 ± 0.40	0.22 ± 0.13	1.0 ± 0.0	0.97 ± 0.03
Abnormal precision	0.92 ± 0.004	0.94 ± 0.03	0.85 ± 0.28	0.9 ± 0.30	0.92 ± 0.002	0.93 ± 0.02
Normal recall	0.0 ± 0.0	0.3 ± 0.45	0.4 ± 0.48	1.0 ± 0.0	0.0 ± 0.0	0.1 ± 0.3
Normal precision	0.0 ± 0.0	0.225 ± 0.39	0.035 ± 0.044	0.096 ± 0.01	0.0 ± 0.0	0.1 ± 0.03

albeit measured in the presence of noises and biases, should be detected normal; the profiles of defective cartridges should be detected abnormal.

(3) Smart camera: From the technicians' experiences, formation of an air bubble at the Y-joint of the ink tubes can affect the pressure measurement, which likely leads to false alarms. We design a smart camera system to monitor the Y-joint. It runs a CNN to detect air bubble and a computer vision algorithm to estimate the volume of the bubbles. The results are used to assist the profile classifier or the AD algorithm in deciding the nature of any alarm generated by the tester.

(4) Statistical learning-based tester assessment: We develop a tester assessment approach that leverages statistical learning to estimate the probability that a tester is faulty based on the historical alarm classification results. The estimated probability can support making decisions on whether maintenance activities should be performed for the concerned tester. With the assessment support, more false alarms can be prevented proactively.

All computing for the profile classification and bubble detection is executed on a Raspberry Pi single-board computer deployed close to the sensors generating data. Specifically, the Pi is connected directly with the camera and tester to receive the captured images and measured pressure profiles.

4 ML-BASED PRESSURE PROFILE CLASSIFIERS

This section presents the design of the ML-based profile classifiers. It also evaluates the performance of the designed classifiers on the historical data samples.

4.1 Preparation of Design Data

We receive a dataset containing 550,508 pressure profiles of 723 ink cartridge models collected from the testers deployed in HP's factories in 18 months. The dataset includes the profile labels which are generated by the tester using the bound-based detector. Specifically, the bound-based detector classifies about 2% of profiles abnormal. However, the actual defect rate of the manufactured ink cartridges is about 0.07% only. This result suggests that most abnormal profile labels generated by the bound-based detector are inaccurate. We work with HP's product engineers and domain experts to manually relabel the abnormal profiles in the dataset. However, the relabeling is tedious and time-consuming. We can only confirm 134 abnormal profiles. Eventually, we have a dataset consisting of about 530,000 profiles with reliable "normal" labels, merely 134 profiles with reliable "abnormal" labels, and about 110,000 profiles that were classified abnormal by the bound-based detector but unlabeled after the relabeling process. This renders the training dataset imbalanced with limited data with abnormal labels. The difficulty of the labeling process will be further discussed in §9.

4.2 Design of ML-based Classifiers

As discussed in §1, each ML approach addresses a specific problem based on a set of assumptions, but real-world tasks often face a mix of many problems. In practice, it is often more efficient to

Table 2. Accuracy of ML-based classifiers with under-sampling and over-sampling over historical profiles.

Metrics	Under-sampling			Over-sample		
	CNN	DT	MVAE	CNN	DT	MVAE
Accuracy	0.68 ± 0.09	0.65 ± 0.15	0.51 ± 0.15	0.84 ± 0.09	0.89 ± 0.04	0.47 ± 0.19
Abnormal recall	0.67 ± 0.11	0.64 ± 0.17	0.48 ± 0.16	0.86 ± 0.10	0.94 ± 0.05	0.46 ± 0.23
Abnormal precision	0.98 ± 0.03	0.98 ± 0.03	0.98 ± 0.03	0.96 ± 0.04	0.94 ± 0.03	0.95 ± 0.06
Normal recall	0.80 ± 0.4	0.8 ± 0.4	0.9 ± 0.3	0.60 ± 0.48	0.3 ± 0.45	0.6 ± 0.48
Normal precision	0.15 ± 0.09	0.15 ± 0.09	0.13 ± 0.07	0.26 ± 0.23	0.13 ± 0.20	0.07 ± 0.07

try multiple ML approaches than relying on a single approach unless we clearly know that the conditions of the task well match the assumptions of the single approach. As such, we have tried four ML-based profile classifiers which are the CNN-based, decision tree (DT)-based, multimodal variational autoencoder (MVAE)-based, and k-means-based classifiers. The detailed design of these four ML-based classifiers can be found in our prior publication [18]. In addition, as an ensemble of multiple ML-based classifiers is often more accurate than any single member classifier [13], we also try the ensembles of the four classifiers with distinct combination rules. Specifically, we adopt a widely used ensemble method called *bagging* [13], which combines the results of the four ML-based classifiers to yield the final result. We implement two variants of the bagging method including *veto* and *majority*. With a primary focus on achieving high recall in capturing defective products, the veto approach considers the profile as abnormal if any of four classifiers outputs abnormal. The majority approach yields the majority of the classifiers' results as the final result.

4.3 Evaluation based on Historical Data

We evaluate the performance of four ML classifiers and two ensemble approaches using the historical profile samples with reliable labels (cf. §4.1). Specifically, we follow the 10-fold cross-validation procedure to train the CNN-based, DT-based, and MVAE-based classifiers. This procedure is often used to evaluate the ML models on small datasets. Specifically, the training dataset is equally divided into 10 groups with the same ratio between the abnormal and normal profile samples.

We use the overall classification accuracy, recall, and precision in detecting the abnormal and normal profiles as the evaluation metrics. Table 1 shows the evaluation metrics of the four ML-based classifiers and two ensemble methods on the 134 training samples. From Table 1, the CNN-based classifier exhibits the highest average accuracy of 0.9 and abnormal recall rate of 0.97 among the four classifiers. The DT-based classifier has the highest average abnormal precision of 0.94. Both two ensemble methods (i.e., veto and majority) always achieve higher classification accuracy than each individual classifier. Moreover, the veto method has the highest average recall in detecting the abnormal profiles. However, as presented in §7.2, these trained ML-classifiers cannot achieve the accuracy level of at least 90% on the testing samples that we collect from the controlled experiments in the deployment phase of our system.

The main reason causing the inferior accuracy performance of the ML-based classifiers is that we can only label 134 historical training samples with a majority of normal samples. The imbalanced training dataset and limited training samples pose substantial challenges for the classifiers to achieve high accuracy. In general, ML techniques such as resampling [11] and few-shot learning [17] can be used to mitigate these problems. Therefore, we adopt two common resampling methods which are under-sampling and over-sampling to create a balanced dataset for training our developed ML-based classifiers. Specifically, the under-sampling method reduces the number of samples in the majority classes, while the over-sampling method duplicates samples from the minority classes. As a result, a balanced training dataset can be achieved.

Table 2 presents the accuracy results of the supervised (i.e., CNN-based, DT-based) and semi-supervised (i.e., MAVE-based) classifiers with the under-sampling and over-sampling methods. From Table 2, two resampling methods do not help improve the accuracy of the developed supervised and semi-supervised ML-based classifiers. They even lead to the low overall accuracy on the training samples. Moreover, the resampling can be used to create a more balanced dataset only. However, it cannot help expand the training data distribution to cover unobserved/unlabelled abnormal profile samples. On the other hand, although the few-shot learning can build accurate ML models with limited training samples based on prior knowledge about the data structure and learning process, we have limited knowledge about the dynamics of the pressure-volume profiles.

5 ANOMALY DETECTION (AD)-BASED PRESSURE PROFILE CLASSIFIERS

As evaluated in §4, the developed ML-based profile classifiers show limitations in achieving high accuracy due to the limited training dataset. In this section, we develop a heuristic approach which treats the profile classification as an AD problem. Specifically, our approach considers the abnormal profiles as outliers which do not follow the expected pattern of the normal profiles. Upon a new profile, a distance-based similarity score between itself and the normal profiles is calculated. The profile is considered abnormal if the score is lower than a threshold. This AD approach provides good interpretability in that it gives information for understanding the classification results. In this section, we present four categories of false alarms and then describe the AD approach.

5.1 Categories of Alarm-Triggering Normal Profiles

As mentioned in §3, the liquid pressure measurements are subject to various biases due to the human operators and the tester deviations. The biases can cause different patterns of the normal profiles that trigger the bound-based detector. From the product engineers' domain knowledge and experiences, the normal profiles can be divided into four categories as follows.

Miss-configuration profiles are caused by setting a wrong reference point by the human operator at the beginning of the test. With the wrong reference point, the measured profiles have a similar pattern to the profiles of good ink cartridges. However, they are shifted beyond the belt area between the two bounds of the template profile which is used by the tester to classify the profiles into normal and abnormal. As a result, these miss-configuration profiles trigger false alarms.

Miss-calibration profiles are caused by configuring a wrong gain to scale the sensor's raw readings to the pressure unit in the calibration process of the pressure sensor.

No-cartridge profiles are collected when the ink cartridges are not inserted properly onto the tester. Without the ink from the cartridge, the motor pump of the tester pulls the air through the tube only. Under this condition, the measured pressure profile is nearly a flat line.

Tube-blocking profiles are measured when the ink tubes are blocked by air bubbles or ink residue. Specifically, the tube-blocking profiles have a liquid pressure drop in the early stage of the extraction due to presence of the air bubbles inside the tube. Then, they quickly increase and recover to the pattern which is similar to a shift-up variation of the normal profile.

5.2 Anomaly Detection

From the technician's experiences, the last phase of the profiles often includes the pressure measurement fluctuations caused by over extraction in which the tester's motor pump still operates when the internal valve of the ink cartridge is already closed. The air gaps traveling through the tube introduce measurement fluctuations that can trigger the bound-based detector. Thus, our AD algorithm excludes such fluctuations from the input profile. Moreover, our experiments in §7 show that the over extraction has a strong correlation with the presence of air bubble in the tube. Thus, we use air bubble as an indicator to determine whether the measurement fluctuations are

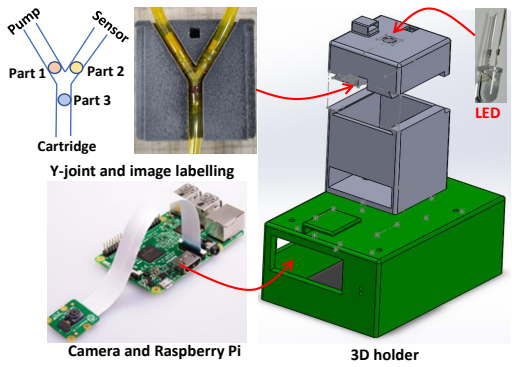


Fig. 3. Smart camera fixed into a 3D-printed holder for Y-joint monitoring.

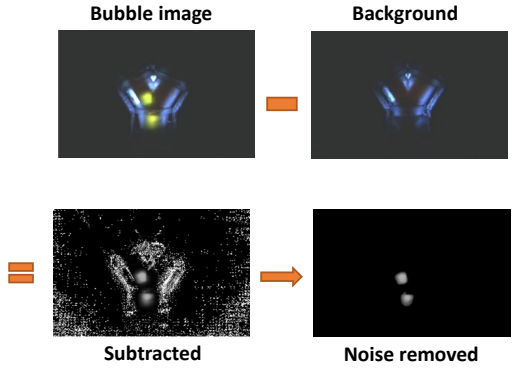


Fig. 4. Workflow of computer vision (CV)-based air bubble size measuring.

caused by over extraction. Lastly, we apply data analytics methods to extract the features of the normal profiles that are used to distinguish the abnormal profiles as outliers. Specifically, we check whether a testing profile belongs to any of the four categories presented in §5.1. If yes, it is normal; otherwise, it is abnormal. The details of the check are as follows.

For the miss-configuration, no-cartridge, and tube-blocking categories, we use the mean subtraction method to normalize the original profile by subtracting its pressure measurements from its average. Dynamic time warping (DTW) distances [3] between all pairs of normalized training profiles in the normal profile category i are calculated. We define γ_i as the detection threshold for category i and $\gamma_i = \mu + 3\sigma$, where μ and σ are the mean and standard deviation of the calculated DTW distances. Upon a new profile, we first calculate the DTW distance between itself with all training profiles of the category i. If the mean of the calculated distances is less than γ_i, the profile is considered normal in the category i.

For the miss-calibration category, we use a scale matching method to extract profile features. Each training profile is equally divided into 10 segments and the maximum among the pressure measurements of each segment is determined. The mean and variance of the maximum over the same segment across all training profiles are calculated. For a new profile, we first determine the maximum of its 10 segments, and then compute their scale with respect to the mean and variance obtained from the training profiles. The profile is considered normal if all scales of its 10 segments fall within a suitable range between each other. If the profile is considered normal by the above scale matching approach, we additionally perform the DTW distance-based AD process to confirm whether the profile is normal.

6 SMART CAMERA SYSTEM

As mentioned earlier, the presence of the air bubbles inside the tester's tube can affect the pressure sensing and is indicative of over extraction. Thus, we design and deploy a smart camera with an embedded image processing pipeline to monitor the air bubbles during the ink extraction.

6.1 Hardware Components

Fig. 3 illustrates our camera system that consists of three main components: the low-cost camera, the edge node, and the light source. For the camera, we select the Raspberry Pi camera module that can capture up to 90 images per second. The captured images are transferred to a Raspberry Pi 4 edge node that runs the CNN and traditional computer vision (CV) algorithms.

An external light source is used to illuminate the ink tube for the camera. To reduce the impact of the tube's vibration on the camera's image sensing, all hardware components and Y-joint are fixed into a custom 3D-printed holder as shown in Fig. 3. We deploy the camera system to monitor the air bubbles at the Y-joint of the tube since the air bubbles are often trapped by the Y-joint.

6.2 Image Processing

We implement a two-step processing pipeline to process the images at the Raspberry Pi. First, the image is fed to a CNN to detect the bubbles in the Y-joint. Specifically, each image is characterized by three labels that indicate the presence of the bubbles in the three tube channels of the Y-joint as shown in Fig. 3. To train the designed CNN, we collected and manually labeled an dataset of 1,494 and 1,455 images with and without the bubbles, respectively. Different from relabeling the pressure profiles, this labeling process is easy because human can easily recognize the bubbles.

Second, we develop a CV-based framework to determine the size of the detected bubbles as shown in Fig. 4. In particular, a previously captured image without air bubble is used as the background. Upon a new image with bubbles, a background subtraction method is used to extract the bubble areas by subtracting the image from the background. Then, the morphological processing is adopted to remove the noises from the extracted bubble areas. Finally, the number of points with the pixel value greater than zero is yielded as the size of the air bubble. The background is updated once a new image without the bubbles is captured.

6.3 Usages of the Smart Camera

We use the camera system to reduce the maintenance overheads and improve the ML-based classifiers or the heuristic-based AD. First, it provides an indicator to determine whether the bubbles are completely removed after performing a water flushing round. As mentioned earlier, the current protocol of the factories performs water flushing to purge out ink and bubbles at the end of every test. This process is labor intensive and usually requires a number of attempts. Thus, to reduce the flushing overheads, the camera system can be used to check whether the air bubbles are completely removed from the tube. Once the tube is clear without bubbles, the flushing process can be stopped. Second, the bubble detection and size measurement functions can be used to avoid the measurement fluctuations during the over extraction period. Specifically, in the last phase of the tests, we stop the pressure measurement when a bubble with a certain size is detected. The bubble presence is also used as an indicator to determine and exclude the over extraction period.

7 DEPLOYMENT AND EVALUATION EXPERIMENTS

This section presents the deployment of our AIoT system integrating the components presented in §4 and §5 in an HP factory and the results of the evaluation experiments conducted on an operational tester.

7.1 Deployment

We deploy our AIoT system to an operational tester in an HP factory. Specifically, we use Python and several ML libraries including PyTorch, TensorFlow Lite, and Scikit-Learn to implement the ML-based classifiers and AD module running on a Raspberry Pi 4. At the end of each testing round, the tester reports the measured profiles of six tested cartridges to the workstation computer. The profiles triggering alarms are then transferred to the Pi for further classification into normal (i.e., the tester-induced alarm) or abnormal (i.e., the product-induced alarm) profiles. We also deploy six units of the smart cameras to monitor the bubbles at the Y-joints of six tubes connected to six testing modules. The camera periodically captures an image of the Y-joint and transfers it to the Pi at every two seconds during the testing period.

Table 3. Accuracy of ML-based classifiers over 88 profiles collected from controlled experiments.

Metrics	Classifiers				Ensemble	
	CNN	DT	MVAE	k-means	Veto	Majority
Accuracy	0.34	0.51	0.42	0.40	0.65	0.41
Abnormal recall	0.08	0.44	0.38	0.16	0.83	0.17
Abnormal precision	1	0.78	0.67	1	0.72	1
Normal recall	1	0.68	0.52	1	0.2	1
Normal precision	0.30	0.33	0.25	0.32	0.31	0.32

Table 4. Accuracy of AD-based profile classifier over 88 profiles collected from controlled experiments.

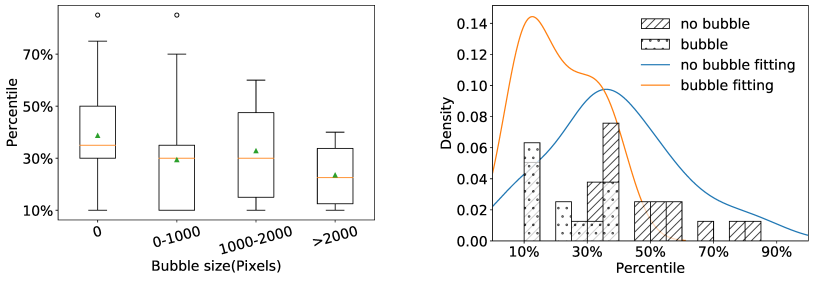
Metrics	Anomaly Detection				Overall	Veto
	Miss-configuration	Miss-calibration	No-cartridge	Tube-blocking		
Accuracy	95.7%	95.7%	100%	100%	96.5%	64.7%
Abnormal recall	95.2%	95.2%	100%	100%	95.2%	82.5%
Abnormal precision	100%	100%	100%	100%	100%	72.2%
Normal recall	100%	100%	100%	100%	100%	20%
Normal precision	70%	72.7%	100%	100%	89.2%	31.2%

7.2 Accuracy of Profile Classification

We perform a set of controlled experiments to evaluate the accuracy of our ML-based classifiers and AD module. We intentionally induce the tester's biases and noises to generate the normal profiles of four categories (cf. §5). Specifically, we create seven miss-configuration profiles by setting an arbitrary reference point in the beginning of tests for seven good ink cartridges. Eight miss-calibration profiles are created by setting a wrong gain parameter to scale the pressure sensor's raw readings to the pressure unit. We also generate six no-cartridge profiles by inserting the ink cartridges improperly such that no ink is extracted under the pressure from the pumps. Moreover, we induce bubbles and ink residue inside the tubes to create four tube-blocking profiles. In summary, we create 25 normal profiles that trigger false alarms. Additionally, we manually induce defects to good ink cartridges by damaging the vent of the cartridges or releasing the pressure into the cartridge to create 15 abnormal profiles. In addition, we run tests for 48 defective cartridges and generate abnormal profiles. As a result, we have 63 abnormal profiles. In summary, our controlled experiments generate a total of 88 profiles whose labels are also confirmed by the domain experts.

We use the overall classification accuracy, recall, and precision in detecting the normal and abnormal profiles as the evaluation metrics. Table 3 shows the evaluation metrics of four ML-based classifiers on the 88 profiles. For the k-means-based classifier, we adopt the settings of $k = 12$ and $k_{th} = 3$. From Table 3, the four classifiers (i.e., CNN, DT, MVAE, and k-means) show the best performance in different metrics. For instance, DT has the highest accuracy and abnormal recall, while CNN and k-means exhibit the best abnormal precision, and normal recall. Moreover, the two ensemble approaches mostly show better accuracy performance. The veto approach has the highest accuracy and abnormal recall.

Table 4 shows the performance of the AD module. The columns headed by miss-configuration, miss-calibration, no-cartridge, and tube-blocking present evaluation metrics of the AD module in detecting the 88 profiles by comparing its similarity score with the normal profiles in each of four category only. The overall column shows the performance results when the scores between the testing profile and the normal profiles in all four categories are used. The AD approach achieves an



(a) Distribution of percentiles. (b) Fitted distributions.

Fig. 5. Impact of air bubbles on pressure measurements. The percentile represents the percentage of historical profiles whose average is lower than the average of the testing profile. In (a), the box, line, triangle, upper and lower whiskers represent middle 50%, median, average, ranges for the bottom 25% and the top 25% of the samples, respectively.

overall accuracy of 96.5% in classifying the testing profiles. Moreover, it always has better accuracy performance, compared with that of the best-performing ML-based classifier, i.e., the veto.

7.3 Performance of Camera System

7.3.1 Accuracy of bubble detection and size measurement. We use 450 captured images in the controlled experiments to evaluate the accuracy of bubble detection by the camera system. The CNN can detect the air bubbles in 450 testing images with an accuracy of 94%. It cannot detect small air bubble in the co-presence of the diluted ink inside the Y-joint. However, the small air bubbles generate little/no impact on the pressure measurements. Moreover, we use 49 images with the air bubbles to evaluate the accuracy of the size measurement by the CV method. We adopt the intersection over union (IoU) as the evaluation metric. In particular, for each image, we calculate the IoU between the detected bubble areas and the ground truth of the bubble areas. The bubble size measurement is considered correct if the calculated IoU is higher than 0.5. Our CV method achieves an accuracy of 79.5% in measuring the sizes of the air bubbles in 49 testing images.

7.3.2 Impact of air bubble on pressure measurement. We use our camera system to capture the top view of the Y-joint at the beginning of the ink extraction for 81 ink cartridges of 6 models over a 7-day operation period of the tester. We perform an analysis on the captured images and the corresponding profiles to study how the bubbles affect pressure measurements. Specifically, we cannot directly compare the 81 pressure profiles with and without bubbles since the profiles of different cartridge models fall in different measurement ranges. Thus, we compare the average of testing profiles with that of profiles of the same cartridge model in our historical dataset. We use the percentage (i.e., percentile) of historical profiles whose average over time is lower than that of the testing profile to characterize the testing profile. Fig. 5(a) shows the box plots of the percentiles of 81 testing profiles which are divided into three groups based on the measured bubble size. The percentiles of the profiles with the bubble size lower than 2,000 pixels have similar average and median. Meanwhile, when the bubble size is greater than 2,000 pixels, the profile percentiles fluctuate in narrower ranges and have lower average. To further investigate the impact of the bubble size on the distribution of the profile percentile, we fit two probability distributions to model the percentiles of the profiles without the bubbles and with the bubble size greater than 2,000 pixels. Fig. 5(b) shows the histograms of the percentiles and the fitted density functions. We can see that the mean percentile of profiles with bubbles is lower than that of the profiles without bubbles.

We also conduct a one-sided Kolmogorov–Smirnov test using testing profiles to check the null hypothesis that the percentile of profile with the bubbles of the size greater than 2,000 pixels is higher than that of the profiles without the bubbles. We obtain a p-value of 0.0273. Thus, the null hypothesis can be rejected. This result implies that the bubbles with large sizes make the pressure measurements statistically lower.

7.3.3 Correlation between the bubble presence and over extraction pressure fluctuation. As mentioned in §5, the measured pressure often has fluctuations during the over extraction period. These fluctuations should be excluded from the profiles for better classification performance. However, it is non-trivial to determine the starting point of the fluctuations in the presence of measurement noises. From prior observations, the over extraction often coincides with bubbles in the tubes. Now, we analyze the Pearson correlation between the bubble presence and the over extraction fluctuations. We collect a dataset consisting of 17 profiles and five profiles with and without over extraction, respectively. An image of Y-joint is captured for each profile. The Pearson correlation is 0.7483 over the 22 collected data points. This result implies that there is a strong correlation between the bubble presence and the over extraction fluctuations. Therefore, our AIoT system uses bubble presence to assist the determination of the presence of over extraction fluctuation.

8 STATISTICAL LEARNING-BASED TESTER ASSESSMENT

8.1 Objective and Approach Overview

The main goal of the profile classification approaches (i.e., the ML-based and AD-based classifiers) and the smart camera system is to reliably assess the alarms generated by the tester's bound-based detector. When a normal profile (i.e., the false alarm) and the presence of the air bubbles are detected, the water flushing is performed to purge ink and bubbles out of the tester's tube. However, the water flushing action can only remove pressure measurement errors due to the effects of the air bubbles and ink residue trapped within the tester's tubes. Beyond the above two effects, the tester may malfunction and generate excessive false alarms due to the wear and tear of its components including the motor pump and pressure sensor. Thus, the operator also needs to periodically perform maintenance activities for assessing and repairing the faulty components of the testers. Frequent assessment processes require many manual efforts and reduce the testing throughput of the testers. In this section, we develop an assessment approach that uses the statistics of the historical testing processes to assist determining whether a tester in question is faulty and planned maintenance activities are necessary. The main goal is to reduce unnecessary maintenance overheads and meanwhile capture the faulty testers to reduce false alarms.

As mentioned earlier, each IET can simultaneously test six ink cartridges via individual *pockets* with the same configuration. Note that the setup illustrated in Fig. 1 is for a single pocket only. Since the six ink cartridges in the six pockets are from the same manufacturing line, they have a certain and identical defect rate. Moreover, since these six pockets operate in the same working condition, their pressure sensing measurements are subject to the similar types of the endogenous and exogenous noises. Thus, if all six pockets are not faulty, they should generate similar true and false alarm rates over the long run. Our proposed tester assessment approach monitors statistics of the alarms generated by the six pockets of a tester based on the historical profile classification results. At a specific time, a pocket is considered as an outlier (i.e., a faulty pocket) and requires the corrective maintenance if its statistics metric has large discrepancy from other pockets. For instance, if one pocket generates a number of false alarms more than other five pockets, it should be inspected.

Algorithm 1: Detect faulty pockets of a tester

Data: N: number of tested products on each machine; D: number of IET pockets; $X_i (i = 1, \ldots, D)$: number of defective products from IET pocket i; ε: defect rate threshold.

$\hat{\varepsilon}^{(i)} \leftarrow X_i / N$; ▷ The estimated defect rate $\varepsilon_1 + \varepsilon_2^{(i)}$.

$S \leftarrow \emptyset$; ▷ Set variable for faulty pockets to be reported.

if $\min_i \hat{\varepsilon}^{(i)} < \varepsilon$ **then**

  $j \leftarrow \text{argmin}_i \hat{\varepsilon}^{(i)}$;

  for $k \neq j$ **do**

    if $Z_P(X_k, X_j) \geq 0.001$ **then**

      S.add(k); ▷ If z-statistic is higher than 0.001, report as defective.

else

  $S \leftarrow \{1, \cdots, D\}$; ▷ If all pockets exhibit high defect rate, report all.

Result: S ▷ Set of faulty pockets.

8.2 Proposed Approach

We formulate the task of detecting faulty pockets of a certain tester as a variant of the *hide-and-seek* statistical learning problem [15]. Specifically, we view both product-induced and tester-induced's effects as corruptions to profile data. Denote p as the desired distribution of profile data, q_1 as the distribution of product-induced profile and q_2 as the distribution of tester-induced profile. All three are unknown. Denote ε_1 and ε_2 as the percentages of corruptions associated with q_1 and q_2, respectively. Thus, the observations are from a mixture of the three:

$$P^{(i)} = (1 - \varepsilon_1 - \varepsilon_2^{(i)})p + \varepsilon_1 q_1 + \varepsilon_2^{(i)} q_2^{(i)}. \tag{1}$$

In addition, we have six independent pockets, with possibly different conditions, identified by (i) in Eq. (1). We aim to estimate the outlier ratios $\varepsilon_1 + \varepsilon_2^{(i)}$ and base our decision on detecting discrepancies among pockets through $i = 1, \ldots, 6$. Our approach takes estimations of outlier ratios as input and return YES/NO for the pockets that are likely to be faulty.

The work [4] proposed an approach to address the above problem, which is employed as a baseline approach in our work. Based on the understanding of the limitations of [4], we design an improved approach. To illustrate the drawbacks of [4], we consider the following setting. Assume that $\varepsilon_1 < \delta_1$ is the same over all pockets considering that every pocket takes samples from the same manufacturing line. Given N samples from each $P^{(i)}$, the approach detects whether $\varepsilon_1 + \varepsilon_2^{(i)}$ is significantly higher than the rest. Assume that $\varepsilon_2^{(i)}$ follows a prior distribution, i.e., $D \sim \text{Bin}(n, \alpha)$, where D is the number of pockets with pocket fault rate $\varepsilon_2 > \delta_2$ for a batch size of N samples. In reality, there is a gap between δ_2 and δ_1, i.e., $\delta_2 \gg \delta_1$. We now show that the approach in [4] can fail to detect cases when the majority of the pockets are faulty, i.e., with $\varepsilon_2 > \delta_2$, and the main reason is the use of a biased estimator of defect rate average. In the approach of [4], the null hypothesis is that there is no faulty pocket. Thus, the approach computes the average defect rate from all pockets and make decisions based on the difference of each individual pocket to this estimate being statistically significant. An estimate of average defect rate caused by both the pockets and products can be calculated. Denote $\hat{p}$ as the random variable from averaging over all pockets and p as the actual defect rate of the product. We have $\mathbb{E}(\hat{p}) \geq p + \sum_{i=0}^{n} P(D = i) \left[\frac{i\varepsilon}{n}\right] = p + \frac{\varepsilon}{n} \sum_{k=1}^{n} \binom{n}{k} \alpha^k (1-\alpha)^{n-k} k \overset{*}{=} p + \varepsilon\alpha$, where the derivation step marked by * follows from $\frac{\varepsilon}{n} \sum_{k=1}^{n} \binom{n}{k} \alpha^k (1-\alpha)^{n-k} k = \varepsilon \sum_{k=1}^{n} \binom{n-1}{k-1} \alpha^k (1-\alpha)^{n-k} =$

Table 5. A simulation experiment with common parameters on $D = 6, N = 1000, \varepsilon = 0.04$ and $\varepsilon_1 = 0.02$. $\overline{\text{FPR}}$ stands for average false positive rate, $\overline{\text{FNR}}$ stands for average false negative rate. Both metrics are better if it is a lower value. p-value: 0.001 both.

Parameters	Baseline		Proposed	
Common $D = 6, N = 1000, \varepsilon = 0.04, \varepsilon_1 = 0.02$	$\overline{\text{FPR}}$	$\overline{\text{FNR}}$	$\overline{\text{FPR}}$	$\overline{\text{FNR}}$
$\varepsilon_2^{(i)} = 0.0, i \in [1,5], \varepsilon_2^{(j)} = 0.03, j \in (5,6]$	1e-4	0.0016	0.15	1.66e-5
$\varepsilon_2^{(i)} = 0.0, i \in [1,4], \varepsilon_2^{(j)} = 0.03, j \in (4,6]$	0.0	0.0392	0.0972	1e-4
$\varepsilon_2^{(i)} = 0.0, i \in [1,3], \varepsilon_2^{(j)} = 0.03, j \in (3,6]$	0.0	0.216	0.0533	5e-4
$\varepsilon_2^{(i)} = 0.0, i \in [1,2], \varepsilon_2^{(j)} = 0.03, j \in (2,6]$	0.0	0.518	0.020	0.0019
$\varepsilon_2^{(i)} = 0.0, i \in [1,1], \varepsilon_2^{(j)} = 0.03, j \in (1,6]$	0.0	0.792	0.0	0.0115

$\varepsilon \sum_{k=0}^{n-1} \binom{n-1}{k} \alpha^{k+1}(1-\alpha)^{n-k-1} = \varepsilon\alpha \sum_{k=0}^{n-1} \binom{n-1}{k} \alpha^{k}(1-\alpha)^{n-1-k} = \varepsilon\alpha$. The problem with testing with the average defect rate is that, while it works fine when the majority of the pockets are functional, it cannot handle the case when half of the pockets are faulty.

In the above cases, the ones with the lower defect rates are more likely to be correct under the approach in [4]. Our improved method works by estimating averages from these subsets rather than the entire population – trading off lower bias with higher variance. Our approach is more robust since it fails only when all pockets are faulty. We now present our algorithm. First, our algorithm makes sure that ε_1 is below some threshold by checking $\hat{\varepsilon}_1 \leq \hat{\varepsilon}_1 + \hat{\varepsilon}_2^{(i)} \leq \varepsilon$ for all i, where ε is the defect rate threshold. If this is not true, our method declares that either all pockets require maintenance or there is some component not functioning in the upstream production line. Otherwise, our algorithm uses one of the pockets that pass the test as our baseline to conduct two sample tests. Algorithm 1 shows the pseudocode of our algorithm. The Line 6 of Algorithm 1 uses the Z_P function for exact Z-pooled test, which is defined as $Z_P(X_1, X_0) = \frac{\hat{p}_1 - \hat{p}_0}{\sqrt{\tilde{p}(1-\tilde{p})(\frac{1}{n_1}+\frac{1}{n_0})}}$, where $\hat{p}_i = \frac{X_i}{N}$ and $\tilde{p} = \frac{X_1+X_0}{2N}$. The exact Z-pooled test in Algorithm 1 adopts a p-value threshold of $\frac{0.005}{5} = 0.001$ to ensure that the type-I (all pockets do not malfunction, but at least one of them reported as defective) error is at most 0.005, where the normalization by 5 is an adjustment for the family-wise error rate.

Our algorithm cannot distinguish between all pockets exhibit error and production line exhibits error without examining other aspects of the process. Compared with the method used in the manufacturing line at present, our algorithm does not require involvement of engineers at the detection phase. Compared with [4], our analysis shows that there exists certain bias-variance trade-off. By exploiting this trade-off, we can obtain an algorithm better suited for our application.

8.3 Evaluation Results

We perform simulation-based evaluation to demonstrate the performance of the baseline algorithm in [4] and our proposed algorithm. Specifically, these algorithms take YES/NO results from the IET or other algorithms that perform the profile-specific analysis. Through these YES/NO responses, these algorithms check the discrepancies between $D = 6$ independent pockets. These YES/NO responses are simulated, which are sampled from a Bernoulli distribution with parameters matching the actual pockets, i.e., the functional pocket generates 2% errors (actual product failure rate) and the malfunctioned pocket generates at least 5% errors. We fix a seed for the pseudorandom number generator and repeat the experiments for 10,000 times and collect the corresponding metrics for Table 5. We use the exact Z-pooled test as our two sample pockets. One of the metrics used is average False Positive Rate (FPR), i.e., the ratio of functioning pocket being flagged as malfunctioned

Table 6. A simulation experiment with common parameters on $D = 6, N = 1000, \varepsilon = 0.04$ and $\varepsilon_1 = 0.02$. $\overline{\text{FPR}}$ stands for average false positive rate, $\overline{\text{FNR}}$ stands for average false negative rate. Both metrics are better if it is a lower value. p-value: 0.8 (baseline) vs 0.001 (proposed)

Parameters	Baseline		Proposed	
Common $D = 6, N = 1000, \varepsilon = 0.04, \varepsilon_1 = 0.02$	$\overline{\text{FPR}}$	$\overline{\text{FNR}}$	$\overline{\text{FPR}}$	$\overline{\text{FNR}}$
$\varepsilon_2^{(i)} = 0.0, i \in [1,5], \varepsilon_2^{(j)} = 0.03, j \in (5,6]$	9.98e-02	0	1.50e-01	1.67e-05
$\varepsilon_2^{(i)} = 0.0, i \in [1,4], \varepsilon_2^{(j)} = 0.03, j \in (4,6]$	7.70e-03	1.17e-04	9.72e-02	1.33e-04
$\varepsilon_2^{(i)} = 0.0, i \in [1,3], \varepsilon_2^{(j)} = 0.03, j \in (3,6]$	2.67e-04	2.48e-03	5.33e-02	4.83e-04
$\varepsilon_2^{(i)} = 0.0, i \in [1,2], \varepsilon_2^{(j)} = 0.03, j \in (2,6]$	1.67e-05	0.0311	2.00e-02	1.87e-03
$\varepsilon_2^{(i)} = 0.0, i \in [1,1], \varepsilon_2^{(j)} = 0.03, j \in (1,6]$	0	0.174	0	0.0115

by the algorithm. It is averaged over these 10,000 repetitions. The other metric False Negative Rate (FNR) is the ratio of malfunctioned pocket being flagged as functioning pocket by the algorithm. Both two metrics are better if they have lower values. Both tables use the same error rates (i.e., 2% and 5%) for functional and malfunctioned pockets and iterate through 5 scenarios: 5, 4, 3, 2, 1 functional pockets in a total of 6 pockets, where the first pocket is always functional and the sets of malfunctioned pockets changes from $\{6\}$ to $\{2, 3, 4, 5, 6\}$. Difference between the two tables is the sample size N, i.e., the number of samples that the algorithm collects before making decisions.

The baseline approach works by computing the average defect rate over all 6 pockets. Based on this estimate, the baseline algorithm conducts an exact Z-pooled test. Therefore, when more than one pocket malfunction, the baseline algorithm overestimates the defect rate, making it harder to detect malfunctioned pockets. As a result, it generates more false negatives. As seen from Table 5, the average FNR of the baseline algorithm becomes higher when the number of malfunctioned pockets increases, and it persists even if we increase the sample size. On the other hand, while the proposed algorithm has a higher FPR in the case of five functional pockets and one malfunctioned pocket, it has lower FNR and better detection rate when the number of pockets is more than two. From Table 6, we can tune the two algorithms to have similar performance. However, the p-value for the baseline algorithm becomes less meaningful.

We can see from the evaluation that both malfunction detection algorithms have their merits. For IET, slightly higher false positive rates are in general acceptable. This is because the maintenance cost of IET is generally cheaper than inspecting/logging extra failure cartridges.

9 EXPERIENCES AND LEARNED LESSONS

As a systematic attempt of developing an industrial AIoT system for improving the QC of ink cartridge manufacturing, our research has generated experiences and learned lessons that the future industrial practices can consider. The experiences and lessons are summarized as follows.

(1) Classifiers vs. heuristics: In the early stage of our system development, we considered the problem of dividing the profiles into normal and abnormal classes as a classification problem. However, the four ML-based classifiers cannot achieve a high accuracy in the deployment. A main reason is the limited and imbalanced training dataset, which is also related to the second challenge that we will discuss shortly. Then, we investigated the characteristics of the normal and abnormal profiles. Specifically, the tester often induces stable biases and noises to the pressure measurement of all tested ink cartridges over a certain period of time. The profiles of defective ink cartridges are rare ones which do not follow the pattern of the profile of good cartridges under the tester-induced noises and biases. Thus, we further designed a heuristic approach that considers the profile classification as an AD problem. Our evaluation results based on the controlled tests

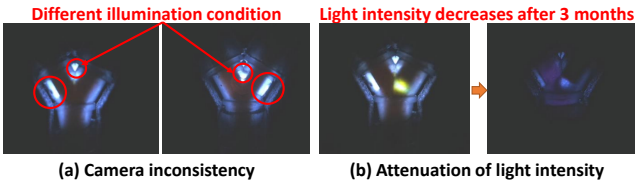


Fig. 6. Impact of sensor condition on data quality.

show that the AD approach outperforms the ML-based profile classifiers. From our experience, the quality of the training data is crucial to the development of effective ML classifiers. It is often very difficult to achieve satisfactory performance if the data is limited or include high-variance noises and biases. In such cases, simpler, heuristic solutions (e.g., AD approach in our case) can be more effective.

(2) Curse from data labeling: ML classifier's attractive advances recently are mainly owing to availability of big labeled training data and standardized hardware acceleration. For the tasks that humans are good at, creating big labeled training datasets is feasible. Manual labeling services (e.g., Google's [5]) are now established. However, data labeling is very challenging for developing an industrial AIoT system. Such labeling processes cannot be performed by normal persons based on their instinct and/or basic knowledge. Differently, they require experts' experience and prior knowledge. In our work, relabeling the pressure profiles is highly non-trivial and requires a collaboration with the tester domain experts. In particular, the experts sometimes lack high confidence and consistency for assigning labels for high-variance profiles. This can be solved if they can access meta information about the internals of the tested ink cartridges and tester's parameters. However, this meta information was not collected in the historical database. Even if the meta information is available, frequently referring to the detailed meta information inevitably adds overhead to the relabeling process. Eventually, we can only relabel a limited number of profile samples, which lead to the poor performance of our ML-based profile classifiers. The use of ML classifier in our AIoT system is limited to the bubble detection, which is a task that a normal human can complete after receiving some simple guidance. From this experience, it is reasonable to argue that the success of applying ML classification to an industrial task highly depends on the availability of sufficient labeled data.

(3) System challenges: Sensor inconsistency and deviation pose challenges for the deployment of industrial AIoT systems in practices. In our system, we use a camera to capture images to train the CNN for detecting the air bubbles. A light source was used to provide a stable and sufficient illumination for the camera to capture the training images. Then, the trained CNN was deployed to six sets of cameras. However, the trained CNN did not show the same performance on them. This is because the quality of captured images across six cameras are different due to the deviation in installation and working condition of the cameras and light sources. Fig. 6(a) shows two images captured by two camera sets. We can see that they have different illumination conditions, which affect the performance of the CNN. Moreover, the illumination condition of a certain camera can drift over time due to wear and tear of the light source. Fig. 6(b) presents two images captured by the same camera set at the beginning of the deployment and three months later. The light intensity of the light source is weakened. As a result, the CNN cannot correctly detect the air bubbles in the images captured with weakened lighting conditions. Although the dimming was caused by that the light was kept on all the time, which was then replaced with on-demand switch-on, the long-term wear and tear are inevitable. This calls for new research to obviate negative impacts of sensor inconsistency and deviation on performance of AIoT systems. The method proposed in [10]

may be promising to address the issues. Specifically, we can model the relationship between the images captured by different cameras or under different controlled illumination levels. Then, we can use the modeled relationship to augment the training dataset. As such, the trained CNN can have the capability to deal with different cameras and illumination levels.

10 CONCLUSION

This technical note presented the design, deployment, and evaluation of an industrial AIoT system for improving the quality control of HP Inc.'s ink cartridge manufacturing lines. Specifically, the evaluation results showed that our AIoT system can help improve the accuracy of the HP Inc.'s testers in detecting defective ink cartridges. This technical note also developed a statistical learning-based tester assessment support approach that detects the faulty pockets of a certain tester. The lessons learned and experiences discussed in this technical notes can be useful to the developments of other industrial AIoT systems, especially those for QC purposes.

ACKNOWLEDGMENTS

This study is supported under the RIE2020 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contribution from the industry partner, HP Inc., through the HP-NTU Digital Manufacturing Corporate Lab. We thank Chou Po-Yi, HP Inc. product engineer, for providing expert knowledge on ink cartridge and the operation management team for supporting us to conduct tests and evaluation at an HP Inc.'s manufacturing facility.

REFERENCES

[1] Fahed Alkhabbas, Romina Spalazzese, Maura Cerioli, Maurizio Leotta, and Gianna Reggio. 2020. On the Deployment of IoT Systems: An Industrial Survey. In *Proceedings of the IEEE International Conference on Software Architecture Companion (ICSA-C)*. 17–24.

[2] Samuel Budd, Emma C. Robinson, and Bernhard Kainz. 2021. A survey on active learning and human-in-the-loop deep learning for medical image analysis. *Medical Image Analysis* 71 (2021), 102062.

[3] Xingyu Cai, Tingyang Xu, Jinfeng Yi, Junzhou Huang, and Sanguthevar Rajasekaran. 2019. DTWNet: a dynamic time warping network. *Advances in Neural Information Processing Systems (NeurIPS)* 32 (2019).

[4] Alexander Chakhunashvili and Bo Bergman. 2006. An EWMA Solution to Detect Shifts in a Bernoulli Process in an Out-of-control Environment. *Quality and Reliability Engineering International* 22, 4 (2006), 419–428.

[5] Google. 2021. Human Labeling. https://cloud.google.com/vision/automl/docs/human-labeling

[6] Iman Gosh. August 12, 2020. AIoT: When Artificial Intelligence Meets the Internet of Things. https://bit.ly/3aMVmRb

[7] Malay Haldar, Mustafa Abdool, Prashant Ramanathan, Tao Xu, Shulin Yang, Huizhong Duan, Qing Zhang, Nick Barrow-Williams, Bradley C Turnbull, Brendan M Collins, et al. 2019. Applying deep learning to airbnb search. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & Data Mining*. 1927–1935.

[8] Kim Hazelwood, Sarah Bird, David Brooks, Soumith Chintala, Utku Diril, Dmytro Dzhulgakov, Mohamed Fawzy, Bill Jia, Yangqing Jia, Aditya Kalro, et al. 2018. Applied machine learning at facebook: A datacenter infrastructure perspective. In *Proceedings of the IEEE International Symposium on High Performance Computer Architecture (HPCA)*. 620–629.

[9] Jimmy Lin and Dmitriy Ryaboy. 2013. Scaling big data mining infrastructure: the twitter experience. *ACM SIGKDD Explorations Newsletter* 14, 2 (2013), 6–19.

[10] Wenjie Luo, Zhenyu Yan, Qun Song, and Rui Tan. 2021. PhyAug: Physics-Directed Data Augmentation for Deep Sensing Model Transfer in Cyber-Physical Systems. In *Proceedings of the 20th International Conference on Information Processing in Sensor Networks (co-located with CPS-IoT Week 2021)*. 31–46.

[11] Ajinkya More. 2016. Survey of resampling techniques for improving classification performance in unbalanced datasets. *arXiv preprint arXiv:1608.06048* (2016).

[12] Janakiram MSV. August 12, 2019. Why AIoT Is Emerging As The Future Of Industry 4.0. https://bit.ly/3rJLtuB

[13] David Opitz and Richard Maclin. 1999. Popular ensemble methods: An empirical study. *J. Artif. Intell. Res.* 11 (1999).

[14] Jacqueline Schmitt, Jochen Bönig, Thorbjörn Borggräfe, Gunter Beitinger, and Jochen Deuse. 2020. Predictive model-based quality inspection using Machine Learning and Edge Cloud Computing. *Advanced engineering informatics* 45 (2020).

[15] Ohad Shamir. 2014. Fundamental limits of online and distributed algorithms for statistical learning and estimation. *Advances in Neural Information Processing Systems (NeurIPS)* 27 (2014), 163–171.
[16] Graphic Products Staff. January 01, 2021. Quality Control In Manufacturing. https://bit.ly/2MjlY2J
[17] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. 2020. Generalizing from a Few Examples: A Survey on Few-Shot Learning. *ACM Comput. Surv.* (2020).
[18] Joy Qiping Yang, Siyuan Zhou, Duc Van Le, Daren Ho, and Rui Tan. 2021. Improving Quality Control with Industrial AIoT at HP Factories: Experiences and Learned Lessons. In *Proceedings of the 18th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON).* 1–9.