QUILT: Quality Inference from Living Digital Twins in IoT-Enabled Manufacturing Systems

Sujit Rokka Chhetri, Sina Faezi, *Arquimedes Canedo, Mohammad Abdullah Al Faruque

University of California, Irvine

*Siemens Corporation Corporate Technology, Princeton, NJ, USA
(schhetri,sfaezi,alfaruqu)@uci.edu, *arquimedes.canedo@siemens.com

ABSTRACT

A *Digital twin* is the virtual replica of a physical system. *Digital twins* are useful because they provide models and data for design, production, operation, diagnostics, and prognostics of machines and products. Traditionally, building a *digital twin* requires many built-in sensors to monitor various physical phenomena associated with cyber-physical systems such as vibration, energy consumption, etc. However, many legacy manufacturing systems do not have multi-physics sensors built-in by default. Moreover, it might not be feasible to intrusively place sensors in these systems after they are manufactured. To bring the advantages of digitalization to legacy manufacturing systems, this paper contributes with an Internet-of-Things (IoT) based methodology to build *digital twins using an indirect medium such as side-channels*, which can localize anomalous faults and infer the quality of the products being manufactured while keeping itself up-to-date. We achieve this by exploring and utilizing the side-channels (emissions such as acoustics, power, magnetic, etc.) of the system that unintentionally reveal the cyber and physical state of the system. To validate our methodology, in this paper, we focus on building a digital twin model of a Fused-Deposition Modeling (FDM) based Cartesian additive manufacturing system. The proposed methodology achieves 83.09% accuracy in anomaly localization. To the best of our knowledge, this is the first work demonstrating the possibility of modeling and maintaining a living *digital twin* of a manufacturing system by extracting information from the side-channels using low-end IoT sensors.

CCS CONCEPTS

• **Computer systems organization** → **Embedded and Cyber-Physical Systems**; • **Information systems** → *Information systems applications; Clustering*;

KEYWORDS

Digital Twin, Smart Manufacturing, Internet-of-Things, Machine Learning

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

IoT'DI'2019, Montreal, Canada

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6283-2/19/04...\$15.00

<https://doi.org/10.1145/3302505.3310085>

1 INTRODUCTION

A *digital twin* is the virtual representation of a physical system (its *physical twin*) [1]. The concept of the *digital twin* was first used by NASA to describe a digital replica of physical systems in space maintained for diagnosis and prognosis. Digital twin consists of large historical context and performance data and utilizes the direct (through inbuilt sensors) and indirect (through latent variable analysis) sensing to provide the near real-time representation of the physical system. Moreover, it consists of various models (for simulation, monitoring, control, optimization, etc.) in a hierarchical manner (consisting of a representation of the system, process, component, etc.) which can provide the blueprint of the whole system [2]. Since, the digital twin allows the user to monitor, simulate, optimize, and control the entire manufacturing system in the virtual domain it is expected to play an important role in the next industrial revolution (Industry 4.0) [3, 4]. Gartner has listed the *digital twin* as one of the top ten technology trends for 2018 and the years to come [5]. Moreover, organizations like Siemens [6], General Electric [7], NASA [8], and the Air Force [9] are currently building *digital twins* of gas turbines, wind turbines, engines, and airplanes that allow them to manage the assets, optimize the system and fleets, and to monitor the system health and provide prognostics.

In the context of next generation of manufacturing systems, additive manufacturing (also known as 3D printing) has allowed designers to rapidly prototype 3D objects layer by layer and brought about significant disruption in the manufacturing domain [10]. In fact, there are various types of 3D printing technologies [11] that enable designers to create 3D objects from light-sensitive polymers, metal powders, thermoplastic filaments, etc. However, these 3D printing technologies are still susceptible to defects due to the large diversity in structure and properties of printed components [12]. Digital twin models, in this situation, could alleviate the cost of manufacturing by providing tools to simulate and infer quality deviation in the virtual domain. In this work, we narrow down the scope towards Fused-Deposition Modeling (FDM) technology based 3D printers, which print 3D objects using thermoplastic filaments such as acrylonitrile butadiene styrene (ABS) or polylactic acid (PLA).

A key enabler for creating a *digital twin* is the availability of a large number of built-in sensors and their historical data. However, current FDM based additive manufacturing printers lack large number of these sensors [13]. It mainly consists of sensor necessary for basic control (such as a temperature sensor, micro-switch, etc.). Lack of sensor arrays makes it a difficult task for sensing the current system states, which is vital for digital twin models. Moreover, the task of building digital twins becomes even harder once the system

has been manufactured as placement and selection of sensors for direct observation of system states can be challenging [14].

Previously, due to the lack of cheap sensors and high-speed and reliable network, acquiring the data for the *digital twin* was costly. Today, thanks to the availability and affordability of IoT sensors, it is becoming easier and cheaper for system operators to acquire large amounts of data on-the-go from their physical systems [15, 16]. However, for 3D printers that do not have built-in sensors and lack historical data, building the *digital twin* with IoT sensors is still a challenge.

1.1 Research challenges and contribution

This paper is motivated by three important research questions that apply to legacy FDM technology based additive manufacturing system without built-in sensors and access to historical data:

- Is it possible to build a *digital twin* of a FDM based additive manufacturing system by indirectly monitoring the side-channels?
- Can this indirect side-channel based digital twin model faithfully capture the interaction between the environmental factors, process parameters of the system, and the design parameters of the product to explain the impact of such interaction on products?
- Can we retrofit low-end IoT sensors to maintain the digital twin up-to-date and use it to predict the product quality (localize faults and infer tolerance deviation) of the next product being produced?

To address these research challenges, this paper provides a first study on the limits of various sensors modalities (such as acoustic, magnetic, power, vibration, etc.) and their contributions towards building and maintaining a *living digital twin*. The key insight of our work is that manufacturing machines generate *unintended* side-channel emissions that carry valuable information about *the machine itself, the product they are producing, and the environment*. Our methodology uses IoT sensors to capture these side-channel information and build a *living digital twin* (see Section 3.6). Our main contribution is a novel methodology to monitor production machine degradation, build their *living digital twin*, and use this *living digital twin* to provide product quality inference (see Section 3.7) while localizing faults (see Section 3.5).

Figure 1: Mutual information analysis on side-channels.

Motivational case study for multisensor data analysis:

In this work, we analyze the data collected from multiple sensors commonly available in IoT devices. These sensors (such as an accelerometer, Hall-effect magnetic sensors, microphone, etc.) are

commonly used in the state-of-the-art IoT devices. Moreover, recently more and more sensors (such as current, humidity, temperature, etc.) are added in IoT devices. For building the system *digital twin*, we propose to capture the interaction between the cyber-domain data (such as G/M-codes carrying geometry and process information), the physical domain input of the system (such as raw materials and energy), and the environment. G/M-codes consists of G and M code. The *digital twin* of the product (the 3D object being created) is initially described using a Computer-Aided Design (CAD) tools. The CAD tool then produces StereoLithography (STL) files which consist of geometry description of the object in coordinate space. Then a Computer-Aided Manufacturing (CAM) tool takes the STL file and slices it into multiple layers and finds a trace to be followed to print the object in each layer. The output of the CAM layer is the G/M-code. In our experiment, we consider that the *digital twin* of the product is described using the G/M-code. G-codes are responsible for controlling the motion (XYZ-axes and extrusion rate of filaments) of the machine for constructing certain geometry shape. Whereas, M-codes are responsible for controlling the machine parameters (process parameters such as temperature, acceleration of stepper motors, etc.). Recently, researchers in [17, 18] have demonstrated that various signals (such as acoustic, magnetic, etc.) collected from the 3D printer behave as side-channels and reveal information about the cyber-domain. Motivated by these work, we performed a preliminary study to measure the mutual information between the four types of sensor data (*acoustic, magnetic, vibration, and power*) and the G/M-codes in a 3D printer. A variation of angle (0° to 90° with step size of 9°) and speed (700 mm/minute to 3300 mm/minute with step size of 100 mm/minute) for printing line-segments is encoded in G/M-codes (with total of 297 G/M-codes) and the principal components various time and frequency domain features extracted from the side-channels were used to calculate the mutual information.

Figure 1 shows that various data collected from different sensors. The percentage mutual information in the y-axis represents how much of the total Shannon entropy of the G/M-code ($\log_2(297) \approx 8.21 \text{ bits}$) can be explained by the analog emissions. We have assumed the distribution of G/M-codes to be uniform for calculating the Shannon entropy. The figure shows that different sensors have varying level of mutual information with the cyber-domain G/M-codes. This means that they behave as side-channels and provide information about the cyber-domain. However, the data collected from these sensors not only allow us to infer the cyber-domain data, but it also captures the current system status (mechanical degradation, system vibration, effects of environment on the system, etc.). Hence, this work leverages the side-channel data to build a *living digital twin*.

1.2 Related work

Since the concept origination, and the onset of emerging technologies, there have been various efforts to model the *digital twin* of a manufacturing system. Knapp and Mukherjee in [19] provide building blocks for modeling *digital twin* for laser-based directed energy deposition additive manufacturing. They use the *digital twin* to estimate the effects of the process variables on cooling rates, single layer deposit geometry, and other structural features. Debroy

and Zhang in [20] surveyed the state-of-the-art and motivate the need for more building blocks to create *digital twins* of additive manufacturing systems. Boschert and Rosen in [21] highlight the simulation aspect of the *digital twin* and its use in the product life-cycle. Alam and Saddik in [22] provide the reference model for the cloud-based cyber-physical system, with an implementation of Bayesian belief network for dynamically updating system based on current contexts. Schroeder and Steinmetz in [23] provide a methodology to model the attributes related to the *digital twin* for providing easier data exchange mechanism between the *digital twins*. Cerrone and Hochhalter in [24] present finite element models of as-manufactured models to predict the crack path for each specimen. Authors in [25] provide a semantic layer which provides a mechanism to pass control feedback and evolve the build parameters on-the-fly for compensating the tolerance.

In summary, all these work have focused on either building the digital twin using simulation of the first principle based equations [26, 27] or just placing expensive sensors for in-situ [28, 29] process monitoring. There is work that uses low-end sensors for in-situ process monitoring [30–33], however, these work do not consider keeping the model up-to-date, using the indirect side-channels, which is the fundamental requirement for the *digital twin*. To this end, we propose a methodology for building the system *digital twin* and keeping it alive using low-cost sensors available in off-the-shelf IoT devices. We use the fact that some of the physical emissions act as side-channels, revealing information about cyber-domain, and that for every control signals in cyber-domain, there is a corresponding physical fingerprint in the physical domain. As it uses the side-channels, this methodology is different compared to the existing methods. Using the proposed methodology, we may be able to find new emissions (that may not have been considered during design time) that are able to better represent the cyber and physical states of the system during run-time.

Figure 2: Digital twin concept for manufacturing.

2 BACKGROUND

2.1 Concept definition

As briefly explained earlier, in a manufacturing environment, we have Digital and *physical twin* of product and system $DT_{product}$, DT_{system} , $PT_{product}$, and PT_{system} , respectively (see figure 2).

The *digital twin* of the product $DT_{product}$ starts its life-cycle in the design phase, where computer aided design and computer aided manufacturing tools are used to represent the product in the cyber-domain. These product *digital twins* from design phase (with an instance represented using X_i) then go through the production, where the *physical twin* of the system PT_{system} takes raw materials, energy, and the $DT_{product}$ to create its corresponding *physical twins* (with an instance represented using Y_i). The *physical twin* of the system PT_{system} consists of actual physical components that are used for manufacturing. The PT_{system} is influenced by the manufacturing environment in a stochastic manner.

Let $DT_{product} = \{\alpha_1, \alpha_2, \dots, \alpha_m\} : m \in \mathbb{Z}_{>0}, \alpha \in \mathbb{R}$ represent the parameters that define the *digital twin* of the product (such as dimension, surface roughness, mechanical strength, etc.), $PT_{system} = \{\beta_1, \beta_2, \dots, \beta_n\} : n \in \mathbb{Z}_{>0}, \beta \in \mathbb{R}$ represent the parameters of the *physical twin* of the manufacturing system (such as flow rate, acceleration values for motors, nozzle temperature, etc.), and let $E_{system} = \{\gamma_1, \gamma_2, \dots, \gamma_p\} : p \in \mathbb{Z}_{>0}, \gamma \in \mathbb{R}$ represent the environmental factors affecting the manufacturing system (such as temperature, humidity, pressure, etc.). Here m , n and p represent the total product, system and environmental parameters that maybe considered for the modeling purpose. We propose to capture the interaction between these parameters using IoT sensors. Using the data collected from multiple modalities (acoustic, vibration, magnetic, power, etc.), we propose to model a stochastic function $\hat{f}(\cdot)$, that performs three tasks: (1) localize the deviation in the $DT_{product}$ parameter from its *physical twin* $PT_{product}$, (2) make sure that the DT_{system} is up-to-date (alive), and (3) infer the quality deviation for the $DT_{product}$ before creating the $PT_{product}$. Moreover, the $DT_{product}$ may interrogate the DT_{system} to infer the quality deviation due to the current status of the PT_{system} .

2.2 IoT sensor data as side-channels

Manufacturing systems consists of cyber and the physical domain. The computing components in the cyber-domain have processes that communicate with the physical domain. A cross-domain signal that is passed from the cyber-domain to the physical-domain have the possibility of impacting the physical domain characteristics. This phenomenon is more prominent in manufacturing system where the *digital twin* of the product causes the *physical twin* of the system to behave in a certain deterministic manner. However, due to these characteristics there exists physical emissions (such as acoustic, vibration, magnetic, etc.) which also leak information about the *digital twin* of the product. We denote these emissions as side-channels, as they indirectly reveal the information about the cyber-domain interactions due to the particular physical implementation of the system. For building the *digital twin* of the manufacturing system that captures the interaction between the product *physical twin*, the environment, and the system's *physical twin*, these side-channels play a crucial role in providing the necessary information. As show in [34, 35], there are various components of the system that reveal information about its internal states through the side-channels. In this paper, we propose to utilize those indirect side-channel information for fault localization, quality inference and for updating the digital twin models. In this work, we analyze four such analog emissions which potentially behave as

side-channels. Let $s_a(t)$, $s_v(t)$, $s_p(t)$, and $s_m(t)$ represent acoustic, vibration, power and magnetic emissions from the manufacturing system. Then we define each of these signals as:

$$s_a(t) = \hat{\delta}_a(\alpha_i, \beta_j) + \gamma_k : i \leq m, j \leq n, k \leq p \quad (1)$$

$$s_v(t) = \hat{\delta}_v(\alpha_i, \beta_j) + \gamma_k : i \leq m, j \leq n, k \leq p \quad (2)$$

$$s_p(t) = \hat{\delta}_p(\alpha_i, \beta_j) + \gamma_k : i \leq m, j \leq n, k \leq p \quad (3)$$

$$s_m(t) = \hat{\delta}_m(\alpha_i, \beta_j) + \gamma_k : i \leq m, j \leq n, k \leq p \quad (4)$$

Equations 1-4 represent the analog emissions as a result of the deterministic function $\hat{\delta}(\cdot)$ which is influenced by the *digital twin* parameters of the product, $DT_{product}$, and the *physical twin* parameters of the System PT_{system} , and the non-deterministic environmental parameters E_{system} . Moreover, for each of the analog emissions, the total number of parameters (α, β, γ) may not be same. Traditionally, non-trivial simulation based approach such as finite element analysis is used to model the deterministic part and explore relation between the $DT_{product}$, $PT_{product}$, and PT_{system} . However, the PT_{system} parameters vary over time, and E_{system} parameters affect the $PT_{product}$ in a stochastic manner. Hence, we explore the possibility of using IoT sensors to model and maintain a live DT_{system} for product quality inference.

2.3 Metric for quality measurement

The *digital twin* can be used for various purposes. However, one of the most fundamental uses of *digital twin* is in predicting the Key Performance Indicators (KPIs). Although the ultimate goal of the *digital twin* will be in predicting a variety of KPIs [36], in this paper, we select quality as one of the KPIs. We will demonstrate that by maintaining a living *digital twin* we can infer the possible deviation in quality of the product. One of the quality metrics that is used is the dimension (Q_d) of the product.

Figure 3: Digital twin modeling methodology.

3 BUILDING THE DIGITAL TWIN

As mentioned earlier, we need to build the *digital twin* from the IoT sensor data to perform three tasks: run-time localization of faults, regularly update of the system *digital twin*, and infer the quality of the product *digital twin*. Hence, in this paper, the digital twin model consists of algorithms and models associated with

fault localization, fingerprint generation, and quality inference. For run-time localization, we propose to create and maintain an active fingerprint library of the individual IoT sensor data corresponding the $DT_{product}$ parameters. This fingerprint also captures the PT_{system} and E_{system} parameters during run-time. Then for localizing the faults, the deviation of the run-time IoT sensor data is compared with the fingerprint. For updating the *digital twin*, a voting scheme is used to check if the majority of the fingerprints are deviating corresponding to few fingerprints. To infer the deviation in quality, we have proposed to estimate a function $Q_d = \hat{f}(\alpha, \beta, \gamma, s_a(t), s_v(t), s_m(t), s_p(t))$, where the Q_d is a function of $DT_{product}$, PT_{system} and E_{system} parameters, and the IoT sensor data. The propose methodology is shown in figure 3. The various components of the proposed methodology is explained as follows:

3.1 $DT_{product}$ parsing

For generating the fingerprint of the $DT_{product}$ from the IoT sensor data, first of all it is parsed to its corresponding parameters ($\alpha_1, \alpha_2, \dots, \alpha_m$). The parsed values will depend on the type of manufacturing system. In the experimental section, we will present the parsing for an additive manufacturing system that uses G/M-codes. These codes are the instruction that carries the process (machine specific parameters, such as temperature, acceleration values for motors, etc.) and product information (for example the geometry description). The parsing will break down the individual parameters from the product *digital twin*.

3.2 Feature extraction

For generating the fingerprint from the analog emissions, in this paper various time domain features such as *Energy*, *Energy Entropy*, *Peak to Peak* features (highest peaks, peak widths, peak prominence, etc.), *Root Mean Square* values, *Skewness*, *Standard Deviation*, *Zero Crossing Rate*, *Kurtosis* (114 features in total) and frequency domain features such as *Mean Frequency*, *Median Frequency*, *Signal to Noise Ratio*, *Spectral Entropy*, *Spectral Flux*, *Spectral Roll Off* from short term 50 millisecond time domain windows (also known as Short Term Fourier Transform) and Continuous Wavelet Transform (CWT) (140 in total), 20 Mel-frequency cepstral coefficients (MFCCs), etc., are analyzed from IoT sensor data. All the analog signals are first synchronized by performing up and down sampling and testing the various window size (5 ms to 100 ms) for highest model accuracy (50 ms in our case). These features have been selected by calculating the Gini importance or mean decrease impurity of well-known time and frequency domain features ($\gg 1000$ in total) used for analysis of time-series data [37]. Principal Component Analysis (PCA) is then performed to further reduce the dimension of these features. Let m be the total number of reduced feature set, then all the features are concatenated for n total samples to create a feature matrix $O \in \mathbb{R}^{n \times m}$.

3.3 Synchronize and segment

Before clustering is performed, the features are synchronized and segmented into subgroups based on the parsed $DT_{product}$ parameters ($\alpha_1, \alpha_2, \dots, \alpha_m$). For instance, the features are segmented based

on conditions such as presence or absence of particular component's movement (for example, motors responsible for moving the 3D printer nozzle in X, Y, Z -Axes). By segmenting based on the parsed parameters, the features are reduced into smaller groups. This allows for further reducing the complexity in acquiring the fingerprints. Henceforth, *group* is used to denote the sub-division of the $DT_{product}$ parameters, which are different than the clusters estimated in the subsequent sections.

3.4 Clustering algorithm

For generating the fingerprint of the parsed parameters of $DT_{product}$, a clustering algorithm is used to generate clusters that group the similar features into a single cluster. For analyzing the clustering algorithm and the corresponding fitness of cluster number, the silhouette coefficient is calculated for each sample. It measures the similarity of the feature to its assigned cluster compared to other clusters, with a high value representing its close match to the assigned cluster. It is calculated as follows:

$$silhouette\ coefficient(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (5)$$

where $a(i)$ is the average intra-cluster distance, and $b(i)$ is the mean of the nearest cluster distance (lowest average distance of i with all other points in another cluster where i is not a member). The clustering is carried out for each group of the features for all the analog emissions. The cluster centers, cluster number and the corresponding average silhouette coefficient of all the analog emission is stored in a library, effectively representing the fingerprint for the given parsed $DT_{product}$ parameter.

Algorithm 1: Algorithm for the fingerprint library generation for digital twin.

Input: Features: $O \in \mathbb{R}^{n \times m}$, Groups: G , Channels: Ch
Output: Fingerprint: (G, Ch , Clusters: C_k , Silhouette Scores)

- 1 Initialize $K = 1, 2, \dots, m$
- 2 Initialize Silhouette Score Threshold $SC_{Threshold}$
- 3 Split $O \in \mathbb{R}^{n \times m}$ into Test and Train set
- 4 **foreach** $ch \in Ch$ **do**
- 5 **foreach** $g \in G$ **do**
- 6 **foreach** $i \in K$ **do**
- 7 Estimate i clusters using Train set of Features
- 8 Use $SC_{Threshold}$ to measure accuracy for clustering the Test Set features
- 9 Select cluster number (k) with highest accuracy
- 10 Re-estimate k cluster with all the Features
- 11 Calculate and Store *Silhouette Score* $_{ch_g}$
- 12 **return** Fingerprint: (G, Ch , Clusters: C_k , *Silhouette Score* $_{ch_g}$)

The pseudo-code for generating the cluster and saving the fingerprint is presented in algorithm 1. Features with their corresponding group and channel name are passed as input and the fingerprint in the form of clusters and their corresponding silhouette scores are given as output. First, Line 1 and 2 initialize the cluster numbers and Silhouette Score Threshold for measuring the accuracy of the cluster estimation. Then Line 3 splits the features into test and train set. Normally 80% of the data is used for training and 20% is

used for testing while performing k-fold cross-validation [38] to validate the accuracy. For each cluster number, Line 7 estimates the clusters for the training set. Then, Line 8 measures the accuracy of the estimated cluster with a specified silhouette score threshold for the test set of features. Based on the obtained accuracy in Line 8, Line 9 to 11 select the cluster number, re-estimate the cluster and store the silhouette scores for all the groups and the channels (acoustic, magnetic, power, and vibration signals).

3.5 Anomaly localization algorithm

The proposed digital twin model of the system is utilized to detect and localize anomaly in the product. To do this, a fingerprint library is created using algorithm 1 is used for detecting and localizing the anomalous physical signals corresponding to the $DT_{product}$ while printing. The algorithm for detecting and localizing the deviation from the stored fingerprint is given in algorithm 2.

Algorithm 2: Algorithm for localizing deviation and checking for digital twin update.

Input: Features: $O \in \mathbb{R}^{n \times m}$, $DT_{product}$
Input: Fingerprint(G, Ch , Clusters: C_k , Silhouette Scores SC_{FP})
Output: Segment of $DT_{product}$ with deviation

- 1 Parse $DT_{product}$ into corresponding parameters
- 2 Segment Feature into corresponding group
- 3 **foreach** $ch \in Ch$ **do**
- 4 **foreach** $g \in G$ **do**
- 5 Get cluster labels CL_i for Features O_i by assigning features to the nearest cluster in C_k
- 6 Estimate current silhouette coefficient ($SC_{current}$) for estimated cluster labels
- 7 **foreach** O and CL **do**
- 8 **if** $SC_{current} < SC_{FP} + SC_{Threshold}$ **then**
- 9 Store $DT_{product}$ segment (Seg)
- 10 $DeviationFlag_{ch_g} += 1$
- 11 $\Delta Deviation_{ch_g} = DeviationFlag_{ch_g} / \text{Total } DT_{product}$ **if**
- 12 $\Delta Deviation_{ch_g} > feature_{Threshold}$ **then**
- 13 $\Delta Deviation_g = Deviation_g / \text{Total Group}$
- 14 **if** $\Delta Deviation_g > group_{Threshold}$ **then**
- 15 $Deviation_{ch} += 1$
- 16 $\Delta Deviation_{ch} = Deviation_{ch} / \text{Total Channel}$
- 17 **if** $\Delta Deviation_{ch} > channel_{Threshold}$ **then**
- 18 Use algorithm 1 to update the library
- 19 **return** Seg

Algorithm 2 parses the features of the $DT_{product}$ either run time or after the product's *physical twin* has been created. Then, using the fingerprint library it estimates the new cluster labels for the parsed features in line 5. Using these labels and the features the new silhouette coefficient for the parsed features are calculated in line 6 using Equation 5. If the calculated silhouette coefficient is less than the stored silhouette coefficient $\pm$ threshold $SC_{Threshold}$ then the $DT_{product}$ segment corresponding to the feature is marked as

deviating from the previous fingerprint and returned as containing a possible anomaly. Moreover, G/M-code adds layers to print the 3D object in sequential order. Hence, if a fault is detected at a certain time, it can be correlated to locate its position in the 3D object.

3.6 Digital twin update algorithm

For updating the *digital twin* model, the library of fingerprint for the $DT_{product}$ have to be updated. However, before updating the library, it should be checked if the anomaly in the fingerprint is temporary or it is due to the degradation of the machine over time. In order to update the *digital twin*, line 10 in algorithm 2 keep tracks of all the $DT_{product}$ variables that deviated. Then line 11 checks if more than $feature_{Threshold}$ of the $DT_{product}$ parameters deviated from the previous fingerprint. Then line 14 checks if more than $group_{Threshold}$ of the groups deviated from the previous fingerprint. Finally, line 17 checks if more than $channel_{Threshold}$ of all the channels deviated. If these condition are met then in line 18 the library for the *digital twin* is updated. This threshold for checking the deviation from the fingerprint can be varied for different channels and groups based on the amount of information leaked by each of the side-channels.

3.7 Quality inference model

To infer the quality variation, we estimate a function $Q_d = \hat{f}(\cdot, \theta)$, where θ represents a function parameter that needs to be learned. Specifically, we treat Q_d as a function of analog emissions, product design parameters, process parameters, and environmental parameters. The quality deviation occurs due to the fact that environment (α) affects the PT_{system} process parameters (β). Due to this, when the $DT_{product}$ is sent to the manufacturing system, variations are introduced in the $PT_{product}$. However, when the environment affects (α) the process parameters (β) it changes the physical structure of various components (for example creation of rust, mechanical eroding, etc.). These changes may cause the side-channel analog emissions from the PT_{system} to vary. The relations between various environmental factors, process parameters, and design parameters may be modeled using first principle (using physics-based equations). However, estimating such functions will require rigorous multi-domain analysis of the complex mechanical system, and may not reflect variation introduced when the system is operating. Instead, we propose to use a data-driven modeling approach to estimate the function $Q_d = \hat{f}(\cdot, \theta)$. This function is estimated using a supervised learning algorithm. To do this, for various α , β , and γ values the corresponding emissions needs to be collected. However, for experimental purpose, we assume that the environmental variation affects the β values. Hence, we only vary α and β values and collect the corresponding analog emissions from the side-channels. We extract various time and frequency domain features from these analog emissions and together with α and β , construct a feature matrix $O \in \mathbb{R}^{n \times m}$. Where m represent the total time and frequency domain features concatenated with α and β parameters, and n represents the total samples. Then, we label each of the rows of $O \in \mathbb{R}^{n \times m}$ to its corresponding quality values and use supervised learning algorithm to estimate the function $Q_d = \hat{f}(O \in \mathbb{R}^{n \times m}, \theta)$. More specifically, gradient boosting based regressor [39] is used to estimate the function $\hat{f}(\cdot)$. It uses an

ensemble of decision trees based regression models. This ensemble generates a new tree against the negative gradient of the loss function and combines weak learner to control over-fitting. Hence, they are robust to outliers and outperform many other learning algorithms as demonstrated in [40]. Since regression trees are used as weak learners, we need to estimate various hyper-parameters such as learning rate, number of weak estimators, maximum depth of the weak learners, etc., to improve the capability of the model to generalize. To do this, the collected feature matrix is divided into test and training set. Then, the testing and training accuracy is used to determine the hyper-parameters that best generalize the function. This estimation function is also updated when the *digital twin* update algorithm reaches a consensus that all the fingerprints are outdated.

4 EXPERIMENTAL SETUP

Figure 4: Experimental setup for modeling the digital twin.

4.1 IoT Sensors

For analyzing the analog emissions from the side-channels, four acoustic (AT2021 cardioid condenser and a contact microphone with sampling frequency set at 20 kHz, whereas high-end industrial microphones have higher sampling frequency greater than 40 kHz), one vibration (Adafruit triple-axis accelerometer with output data rate ranging only from 1.56 Hz to 800 Hz and measurement range of up to $\pm 8g$, whereas high-end accelerometers have ranges beyond 1 kHz with measurement range around $\pm 50g$), one magnetic (Honeywell's magnetometer HMC5883L with output data rate ranging from 75 Hz to 160 Hz and measurement range between ± 1 to ± 8 gauss, whereas high-end magnetic field sensors have data rate range of more than 1 kHz and measurement range between ± 0.6 to ± 100 gauss, and current (a low range Pico current clamp with measurement range of 10 mA to 20A DC or rms AC with AC sampling frequency up to 20 kHz with measurement accuracy of $\pm(6.0\% \pm 30 \text{ mA})$, whereas high-end sensors have much smaller resolution of less than 5mA in measuring minute current fluctuations) sensors are placed non-intrusively without hampering the normal operation of the system. In our experiment for demonstrating the applicability of the proposed methodology, we have used the above-mentioned sensors which have similar sensor specifications available in IoT devices [41]. The Cartesian FDM based 3D printer selected for the experiment is an Ultimaker 3 [42]. The placement of these sensors is performed by position exploration in Cartesian coordinate. The vibration and magnetic sensors measure signals in X, Y, and Z axis. Hence, in total there are four acoustic, three

vibration, three magnetic, and current sensors. We consider them as 11 separate channels. Analog emissions from the additive manufacturing system (or a 3D printer) are automatically collected using National Instruments Data Acquisition (NI DAQ) system whenever a print command is given to it. The analog emission acquisition was carried out in a lab environment with sound pressure level varying between 60-80 dB. The digital twin models are trained and estimated in a desktop computer with Intel i7-6900K CPU with 3.20 GHz clock frequency, 32 GB of DDR3 RAM, and 12 GB of NVIDIA Titan X GPU. Moreover, the digital twin models are stored and retrieved using pickle operation in Python.

4.2 Digital Twin parameters

The sample G/M-code ($DT_{product}$) consists of maximum six parameters, G/M code specifying whether it is machine instruction or coordinate geometry information, travel feed rate F of the nozzle head, the coordinates in XYZ-Axes each and amount of extrusion E . Various 3D test objects normally used for calibrating the 3D printer are downloaded from the open-source website [43] to extract sample G/M – codes. The parsing algorithm in this case separates the $DT_{product}$ based on presence or absence of 5 of these parameters, G/M, X, Y, Z, and E. Hence, $DT_{product} = \{\alpha_1, \alpha_2, \dots, \alpha_{32}\}$ and there are 32 groups. When the manufacturing system is operating, the environmental parameters $E_{system} = \{\gamma_1, \gamma_2, \dots, \gamma_p\}$ affect the physical twin parameters $PT_{system} = \{\beta_1, \beta_2, \dots, \beta_n\}$. This change in β eventually affects the $DT_{product}$ parameters, which in return affects the quality of the product. For example, environmental parameters such as humidity, temperature, etc., may affect the gearbox of the system, which in return may affect the flow-rate of the manufacturing system. For experimental purpose changing the environment parameters $E_{system} = \{\gamma_1, \gamma_2, \dots, \gamma_p\}$ was not performed, instead we have assumed that this parameters eventually affect the β parameter. Hence, analog emissions ($s_a(t)$, $s_v(t)$, $s_m(t)$, and $s_p(t)$) for various α parameters have been collected for optimal β parameters, and the environmental variability have been simulated by varying the β parameters beyond their optimal values to check if the digital twin model is able to reflect those changes. After collecting the analog emissions, time and frequency domain features are extracted from each of them. Moreover, the $DT_{product} = \{\alpha_1, \alpha_2, \dots, \alpha_{32}\}$ consists of timestamps to segment and synchronize the features. For initial training phase, various G/M-code of the 3D-objects (cube, pyramid, cylinder, etc.) are given to 3D printer and their corresponding analog emissions are collected. From them, we proceed to generate the fingerprint library

for maintaining the *digital twin* of the system. Furthermore, in all the training algorithms K-fold cross-validation has been performed to measure the performance of the models and prevent over-fitting.

Figure 6: Experimental setup for sensor position exploration.

4.3 Sensor position analysis

One of the challenges in IoT sensor-based information extraction is figuring out a non-intrusive position of the sensors. This task may also be machine specific. In our experiment, the 3D printers' external surface is considered for non-intrusively placing the sensors. Total of 28 uniform positions are selected. For each of the positions, vibration, acoustic, and magnetic sensors are placed and data is collected for various $DT_{product}$ parameters. Then a gradient boosted random forest is used to create simple classifier to estimate the accuracy of the model based on various sensor location data. The accuracy of the classifier is given as,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

Where TP stands for total true positives, TN stands for total true negatives, FP stands for total false positives and FN stands for total false negatives. is taken as a metric for determining the placement of the sensors around the 3D printer. $DT_{product}$ parameters selected for estimating the classifier consists of simple G/M-code instructions (such as presence or absence of stepper motors movement in X, Y, and Z-axes).

Accuracy score of IoT sensor data is shown in figure 5, this score shows which of the sensors positions are capable of better classifying the stepper motor movements. It may be noticed that for different positions the classification accuracy is different. Moreover, these accuracy results also correlate the mutual information between the various sensor position and the side-channels themselves. Based on these values, a single position is selected for each of the sensors. However, since four acoustic sensors are used, positions with top four classification accuracy are selected for the sensor placement.

Figure 5: Classification accuracy score for sensors positions.

4.4 Performance of clustering algorithms

Figure 7: Scatter plots of the clusters (plotted with the first two principal components of the features for five clusters for acoustic side-channel).

Various algorithms are explored for creating clusters to generate the fingerprints. Among them are *Mini batch K-means*, *Spectral Clustering*, *Ward*, *Agglomerative Clustering*, *Birch*, and *Gaussian Mixture method*. For each of the clustering algorithm, a varying number of clusters are initialized, and the corresponding silhouette coefficient is calculated for measuring the fitness of the features into these clusters. The average silhouette coefficients of the clustering algorithms for all groups and channels are shown in figure 8, and the corresponding scatter plots of acoustic side-channel for cluster number five is shown in figure 7. It may be noticed that although the *Agglomerative Clustering* has a higher silhouette coefficient value, from the scatter plot, the clusters are not well distributed in the scatter plot. However, the *Birch* clustering algorithm has relatively higher silhouette coefficient with a better spread of the cluster centers. Hence, *Birch* algorithm is selected for generating the clusters for fingerprinting the $DT_{product}$. Furthermore, the number of clusters is also estimated based on the accuracy of the *Birch* algorithm using algorithm 1.

Figure 8: Silhouette coefficient of clustering algorithms.

4.5 Anomaly localization accuracy

For testing the accuracy of the *digital twin* for detecting the anomalous signals that can possibly cause deviation in the quality of the product, specialized test 3D object is designed (see figure 9). We have simulated variability of the environment by varying one of the $PT_{system} = \{\beta_1, \beta_2, \dots, \beta_n\}$ parameters. In our experiment, we have selected flowrate as one of the β parameters. Flowrate should be maintained for uniform deposition of the filament while

printing in fused deposition modeling based 3D printers. However, due to sudden slippage, faulty filament, etc., the flow of the filament may deviate from its nominal value.

Figure 9: Test $DT_{product}$ created using CAD tool for checking anomaly localization capability of the digital twin.

The flow rate, a process specific parameter, is calculated as follows:

$$W * H = A = \frac{Q}{v_{feed}} \quad (7)$$

Where W is the width and H is the height of the line-segment being printed on the XY-plane, Q is the constant volumetric flow rate of the material. Q is estimated based on die swelling ration, pressure drop value and buckling pressure of the filament. And $v_{feed} = \omega_r * R_r$ is the feed velocity of the filament. Where ω_r is the angular velocity of the pinch rollers, and R_r is the radius of the pinch rollers. Then, the pressure drop is calculated as follows:

$$P_{motor} = \frac{1}{2} \Delta P * Q \quad (8)$$

Where, P_{motor} is the pressure applied by the stepper motors, ΔP is the pressure drop. Hence, the pressure applied by the motor needs to be maintained for the constant volumetric flow rate. This pressure needs to be less than buckling pressure which is calculated as follows:

$$P_{cr} = \frac{\pi^2 * E * d_f^2}{16 * L_f^2} \quad (9)$$

Where E is the elastic modulus of the filament, d_f is the diameter of the filament, and L_f is the length of the filament from the roller to the entrance of the liquefier present in the nozzle. A sudden change in the pressure can cause the uniform flow of the filament to be disrupted. For validating the application of the digital twin in anomaly localization, the flow rate is varied outside the optimal range ($< 80\%$ and $> 120\%$) at a specific location (see figure 9) for multiple 3D objects. The anomalous flowrate variation introduced is between 40% and 180% with the step-size of $\pm 10\%$. Then the *digital twin* is tested to see if can accurately classify the deviation in quality as an anomaly at the specific location. This is done by first segmenting the 3D object and assigning labels (1 for anomalous flow rate outside optimal range, and 0 for normal flowrate) to these segments. Then comparing these labels with the results of the algorithm 2.

The *digital twin* consists of fingerprints for the optimal flow rate in its library, and the corresponding clusters of the individual channels. When the object is printed, the corresponding features are passed to the *digital twin*, and the silhouetted coefficients corresponding the $DT_{product}$ is calculated. Based on algorithm 2, the analog emissions in each channel is labeled as either being within the deviation limit or exceeding the deviation limit of the silhouette coefficient. For selecting the optimal threshold for making this

Figure 10: Average receiver operating characteristic (ROC) curve for anomaly localization for various sensors.

decision, initially the threshold is varied and the corresponding accuracy of the detection mechanism is measured.

Figure 11: Accuracy of the digital twin's anomaly localization for each channel.

Based on the highest accuracy acquired for each of the channels, the threshold is set and the corresponding classification accuracy for the segments that have been degraded is calculated. Corresponding to the varying threshold the Receiver Operating Characteristic (ROC) curve for some of the channels is presented in figure 10. The accuracy of each channel in detecting the anomalous flowrate is shown in figure 11. Since the features are time stamped, the corresponding section of the $DT_{product}$ maybe calculated after the digital twin has marked the features to be anomalous. From figure 11, it can be seen that analog emissions from microphone number four are more accurate in detecting the degradation of the flow rate. This is due to the fact that this emission is collected by the contact microphone attached near the extruder's stepper motor. Moreover, the average accuracy across all the channels in detecting the anomalous flowrate is **83.09%**.

4.6 System degradation prediction analysis

For detecting the degradation of the System, and hence the need for updating the digital twin, the flow rate for the entire $DT_{product}$ is varied beyond the optimal range. From equation 7 to 9, it is evident that various mechanical degradation (such as worn out rollers), stepper motor degradation over time, etc., may cause the flow rate to be reduced over time.

To check if the digital twin model gets updated to reflect the current status of the system, we perform two experiments. In the first experiment, the current digital twin with its fingerprint library is used to predict the class labels (True for update and False for do

Figure 12: Three of the $PT_{product}$ created using CAD tool for testing quality inference and update capability of the digital twin ($DT_{product}$ thickness is 4 mm).

not update) for the degraded flowrate (60%). Then based on the result of algorithm 2, the updated (or the old) digital twin is used to predict the class labels again for the same degraded flowrate (60%) to see if the digital twin gets updated again. The result of degradation analysis is presented in Table 1.

Table 1: Degradation test result for the digital twin.

Channel	Old Clusters			New Clusters		
	TNR	FPR	Update	TPR	FNR	Update
Mic_1	0.97423	0.0257	True	0.6040	0.3960	False
Mic_2	0.4962	0.5038	False	0.7460	0.2540	False
Mic_3	0.9705	0.0295	True	0.5731	0.4269	False
Mic_4	0.9867	0.0133	True	0.9798	0.0202	False
Current	0.5924	0.4076	True	0.5545	0.4455	False
Vib_x	0.9324	0.0676	True	0.8681	0.1319	False
Vib_y	0.9695	0.0305	True	0.7224	0.2776	False
Vib_z	0.4602	0.5398	False	0.4400	0.5600	True
Mag_x	0.4791	0.5210	False	0.6718	0.3282	False
Mag_y	0.4382	0.5618	False	0.4344	0.5656	True
Mag_z	0.6267	0.3733	True	0.3669	0.6331	True

Table 1 consists of true negative rate, false positive rate and update decision taken for each channel for the old cluster. When the system degrades, we expect the digital twin to find higher negative labels being generated as the silhouette score will be lower than the average silhouette score stored for all the channels and groups. It can be seen that out of eleven channels four of them had the decision of not updating the cluster, and seven of them opting for updating the clusters. Hence, the clusters are updated by algorithm 2. On the other hand, once the cluster has been updated, the analog emissions are labeled as true, hence we expect to see higher true positive rate and lower false negative rate. In the table 1, it can be seen that only three of the channels gave the decision for updating the clusters again, however, eight of them opted for not updating the cluster. This shows that the digital twin is able to update itself during degradation that causes emissions in multiple side-channels to vary.

It may be noted that the side-channels gave different decisions for updating the digital twin. Out of them, the acoustic sensors

Figure 13: Accuracy of the quality inference model.

and vibration side-channels were mostly able to predict the right decision, whereas the magnetic sensors were mostly wrong in this decision. This also correlates with the accuracy values presented in Figure 11. One anomaly to this is the current sensor data. However, it may be noticed that during both the decision model's true positive and true negative rates are very low compared to the acoustic and magnetic sensors. This means that the current side-channel data is not as reliable for the update decision as the acoustic and the vibration side-channels.

4.7 Quality inference

For checking the accuracy of the *digital twin* in inferring the deviation of quality (Q_d), first of all, gradient boosting based ensemble of regressors is used to estimate the function $Q_d = \hat{f}(\cdot)$ for the optimal flowrate range (80% to 120%). For each flow rate, five test objects (with the thickness of 4 mm for $DT_{product}$) are 3D printed, and for each test object, various segments (see figure 12) are created to measure the thickness using the micrometer. Then all the groups of features lying in these segments are assigned a single thickness value. Initially function $Q_d = \hat{f}(\cdot)$ is estimated using optimal flowrate. Then it is used to infer the thickness of the 3D object for various feature samples with varying flow rates. The accuracy of the DT_{system} quality inference model is measured using mean absolute error value.

The result of the quality inference is shown in figure 13. At first, the mean absolute error value of the inference model trained with optimal flowrate range is measured. It can be seen in the figure that for optimal flow rate ranges, the mean absolute error value is around 0.5 mm. Then, at each consecutive step the flow rate of the PT_{system} is varied with a step size of +10% in the positive direction ($> 120\%$) and at the same time +10% in the negative direction ($< 80\%$). It may be seen that in both directions when the system ages (degrades with an increase or decrease in the flow rate), the DT_{system} has increased mean absolute error without the update. This is intuitive as the DT_{system} has not been updated to the new fingerprints. However, once it has been updated the mean absolute error is lower. It may also be noticed that when the system degraded with flowrates at 160% and 50%, the wrong decision was taken by the algorithm 2 in not updating the quality inference model. Due to this, a large increase in mean absolute error was observed for quality prediction other $DT_{product}$. However, this faulty decision was recovered in the consecutive stages. Moreover, the average mean absolute error

in predicting the quality was 0.59 mm (calculated by averaging the mean absolute errors of the inference model after update decision).

4.8 Comparative analysis

Although this paper presents a novel methodology of building a living *digital twin* by using IoT based sensors, there has been a considerable amount of work in quality prediction in additive manufacturing or manufacturing systems in general. In this section, we provide a qualitative comparative study of the various non-exhaustive list of methods compared to the proposed methodology. The result of the comparison is shown in Table 2. It may be observed that there are three general categories of research effort in maintaining quality.

The first is the first principle-based approach (simulation) [26, 45], where quality inference model is based on the process and design parameters. These models although are accurate, they do not account system degradation over time and requires non-trivial formulation of physics-based equations. The second category involves in-situ process monitoring methodologies [44, 47]. These methods monitor the process variation using high-end acoustic and piezoelectric sensors. Compared to these high-end sensor-based methods, our method is able to keep the model updated even using low-end sensor data for fault localization and quality inference¹. The third category involves process monitoring using low-end sensor placement [30–32]. They focus either on specific anomaly detection or quality variation detection. However, these methods do not consider checking the aliveness (up-to-date model) of the model and are mostly limited to anomaly detection. Each of these techniques has their own merit, hence, the proposed methodology is not intended to function independently but in conjunction with various approaches to fully realize the concept of *digital twin*.

5 DISCUSSION

Quality inference: To validate the proposed methodology, the flow rate was used to detect anomalous system behavior and overall system degradation behavior. However, there can be multiple PT_{system} parameters that might affect the quality. However, our methodology can be adjusted over time to consider variation in other PT_{system} parameters over time. We have considered only dimension (thickness of a simple 3D object) as a quality metric.

¹With high-end sensors as theirs, our methodology may achieve higher accuracy in anomaly detection along with the capability of keeping the digital twin most up-to-date at the cost of more computational and resource requirements, which may not be feasible for an IoT paradigm.

Table 2: Comparative analysis of the proposed methodology.

Work/System	Method	Metric	Sensors	Anomaly Detection Accuracy	Checks Model Update	Quality Inference Accuracy
[30]/FFF	Bayesian DP mixture model	Build Failure Detection	Accelerometer, Thermocouple, IR, Borescope	85% (Average F-score)	×	-
[31]/FDM	Functional Qualitative Quantitative Model	Dimension, Surface	Accelerometer, IR, Thermocouple	-	×	~0.3 mm (median RMSPE)
[44]/FDM	Support Vector Machines	Abnormal Extrusion Detection	High-end Acoustic Sensor	95%	×	-
[45]/FDM	First Principle Model of Filament	Dimension	-	-	×	~0.1 mm MAE
[32]/FFF	Online sparse estimation-based classification	Abnormal Extrusion Detection	Accelerometer, IR, Thermocouple	90% (F-score)	×	-
[46]/FDM	Hidden semi-Markov model	Abnormal Extrusion Detection	High-end Acoustic Sensor	91.9% (Accuracy rate)	×	-
[26]/FDM	Theoretical Model	Surface	-	-	×	5.66% MAPE
[29]/SLM	Spectral Convolutional Neural Networks	Build Quality	High-end Acoustic Sensor	79-84%	×	-
[47]/FDM	Heterodyne Technique	Belt Fault Detection	High-end Acoustic & Piezoelectric Sensor	-	×	-
QUILT/FDM	Behavioral Modeling (Random Forest, Clustering)	Dimension	Low-end Acoustic, Accelerometer, Magnetic, Current	83.09% (Classification Score)	✓	0.59 mm MAE

FDM: Fused Deposition Modeling
FFF: Fused Filament Fabrication

SLM: Selective Laser Melting
RMSPE: Root Mean Square Percentage Error

MAE: Mean Absolute Error
MAPE: Mean Absolute Percentage Error

However, for building the full scale DT_{system} , multiple metrics are needed to be considered. We leave this as future work.

More IoT sensors and placement: In this work, sensors with low sampling rate and resolution were used. The number of sensors were limited as well. To improve the accuracy of the *digital twin* techniques such as [13] needs to be incorporated for the development of IoT sensor arrays. We leave this as our future work.

Implementation using IoT device: For building the DT_{system} using IoT devices, further consideration is required for off-the-shelf and wireless IoT devices [13]. Hence, further analysis is required to understand the trade-off between power, time, and performance of the DT_{system} in localizing and inferring the quality variation.

More test cases: One of the limitations of the experimental section was in using a limited number of test 3D objects for inferring the quality and localizing the faults. However, these 3D objects contain structures which provide large possible variation in G/M-code for building the digital twin models. Nonetheless, the digital twin models will be more accurate if large test data are incorporated. We leave this as our future work.

information by fusing these sensors may further justify the proposed methodology. The machine learning algorithm achieves this by carefully selecting the features extracted from the side-channels to build a model to get the highest possible accuracy. However, in our future work, we will dwell deeper in performing mutual information analysis to further clarify the contribution of individual side-channels in indirectly building the digital twin.

Generalizability of the proposed method: As a case study, we presented applicability of the proposed methodology in fused-deposition modeling based additive manufacturing. However, many other technologies also have analog emissions (see Table 3), which may be capable of aiding in the indirect method of building digital twin models. Hence, we hypothesize that the proposed methodology will be able to scale across multiple manufacturing systems.

6 CONCLUSION

This paper presents a novel methodology to build a living *digital twin* of the fused deposition modeling technology based additive manufacturing system by utilizing various retrofitted low-end sensors available in IoT devices to indirectly monitor the system through various side-channels (such as acoustic, vibration, magnetic, and power). Based on these signals, a clustering algorithm is used to generate a fingerprint library, that effectively represents the physical status or the *physical twin* of the system. The *digital twin* is used for localizing the anomalous physical emissions that have the potential of resulting in quality variation. For localizing the error, the *digital twin* achieved an average accuracy of **83.09%**. Moreover, we also presented an algorithm for updating the *digital twin*, and inferring the quality deviation. As a case study the *digital twin* modeling was performed on additive manufacturing system. Compared to the state-of-the-art methods (which do not consider model aliveness), our methodology is able to update itself, infer quality deviation and localize anomalous faults in the additive manufacturing system.

ACKNOWLEDGMENTS

This work was partially supported by NSF CPS grant CNS-1546993.

Table 3: Other additive manufacturing technologies.

Technology	Source of analog emissions		
	Acoustic	Vibration	Power
SLA	Build Platform, Stepper Motor	Sweeper	Motor Controller
SLS	Fabrication Piston	Rollers	Power supply
MJ	Build Tray, Jetting Head	Moving head, Blower, Position belt	Heater, Coil
SLM	Retractable Platform	Leveling Cylinder	Power Supply
EBM	Build Platform	Build Platform	High Voltage Cable

SLA: Stereolithography

SLS: Selective Laser Sintering

MJ: Material Jetting

SLM: Selective Laser Melting

EBM: Electron Beam Melting

Sensor fusion analysis: In the motivation section, we provided mutual information analysis for individual side-channels. We acknowledge that a rigorous analysis of the calculation of mutual

REFERENCES

- [1] E. Negri, L. Fumagalli, *et al.*, "A Review of the Roles of Digital Twin in CPS-based Production Systems," *Procedia Manufacturing*, 2017.
- [2] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary Perspectives on Complex Systems*, Springer, 2017.
- [3] D. C. LLP, "Industry 4.0 and the digital twin: Manufacturing meets its match." <https://goo.gl/6HcBn5>, 2017.
- [4] S. F. T. Group, "Smart factory applications in discrete manufacturing." <https://goo.gl/tT9Sf4>, 2017.
- [5] K. Panetta, "Top Trends in the Gartner Hype Cycle for Emerging Technologies." <https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017/>, 2017.
- [6] R. Schmitt, S. Rose, *et al.*, "Advance: Digital Enterprise - on the way to industrie 4.0." <https://goo.gl/LtL5oy>, 2015.
- [7] G. E. Company, "GE Digital Twin: Analytic engine for the digital power plant." <https://goo.gl/bhQzRF>, 2016.
- [8] E. Glaessgen and D. Stargel, "The digital twin paradigm for future NASA and US Air Force vehicles," in *Structures, Structural Dynamics and Materials Conference*, 2012.
- [9] E. M. Kraft, "The Air Force Digital thread/digital twin-life cycle integration and use of computational and experimental knowledge," in *54th AIAA Aerospace Sciences Meeting*, 2016.
- [10] M. Ashour Pour, S. Zaroni, *et al.*, "Additive manufacturing impacts on a two-level supply chain," *International Journal of Systems Science: Operations & Logistics*, 2019.
- [11] D. Tan, A. Nokhodchi, and M. Maniruzzaman, "3d and 4d printing technologies: innovative process engineering and smart additive manufacturing," *3D and 4D Printing in Biomedical Applications: Process Engineering and Additive Manufacturing*, 2019.
- [12] T. Mukherjee and T. DebRoy, "A digital twin for rapid qualification of 3d printed metallic components," *Applied Materials Today*, 2019.
- [13] F. Baumann, M. Schön, J. Eichhoff, and D. Roller, "Concept development of a sensor array for 3d printer," *Procedia CIRP*, 2016.
- [14] Y. He, J. Guo, *et al.*, "From surveillance to digital twin: Challenges and recent advances of signal processing for industrial internet of things," *Signal Processing Magazine*, 2018.
- [15] A. Banafa, "Iot standardization and implementation challenges," *IEEE.org Newsletter*, 2016.
- [16] A. S. Petrenko, S. A. Petrenko, *et al.*, "The iiot/iot device control model based on narrow-band iot (nb-iiot)," in *Young Researchers in Electrical and Electronic Engineering (EICoNus)*, IEEE, 2018.
- [17] C. Song, F. Lin, *et al.*, "My smartphone knows what you print: Exploring smartphone-based side-channel attacks against 3D Printers," in *Conference on Computer and Communications Security*, ACM, 2016.
- [18] A. Hojjati *et al.*, "Leave your phone at the door: Side channels that reveal factory floor secrets," in *Conference on Computer and Communications Security*, ACM, 2016.
- [19] G. Knapp, T. Mukherjee, *et al.*, "Building blocks for a digital twin of additive manufacturing," *Acta Materialia*, 2017.
- [20] T. DebRoy, W. Zhang, *et al.*, "Building digital twins of 3D printing machines," *Scripta Materialia*, 2017.
- [21] S. Boschert and R. Rosen, "Digital twin-the simulation aspect," in *Mechatronic Futures*, Springer, 2016.
- [22] K. M. Alam and A. El Saddik, "C2PS: A digital twin architecture reference model for the cloud-based cyber-physical systems," *Access*, 2017.
- [23] G. N. Schroeder *et al.*, "Digital twin data modeling with automatonml and a communication methodology for data exchange," *IFAC-PapersOnLine*, 2016.
- [24] A. Cerrone *et al.*, "On the effects of modeling as-manufactured geometry: toward digital twin," *Journal of Aerospace Engineering*, 2014.
- [25] K. Garanger, E. Feron, *et al.*, "Foundations of Intelligent Additive Manufacturing," *arXiv preprint:1705.00960*, 2017.
- [26] L. Di Angelo, P. Di Stefano, and A. Marzola, "Surface quality prediction in fdm additive manufacturing," *The International Journal of Advanced Manufacturing Technology*, 2017.
- [27] A. Boschetto, L. Bottini, and F. Veniali, "Integration of fdm surface quality modeling with process design," *Additive Manufacturing*, 2016.
- [28] S. Shevchik, C. Kenel, C. Leinenbach, and K. Wasmer, "Acoustic emission for in situ quality monitoring in additive manufacturing using spectral convolutional neural networks," *Additive Manufacturing*, 2018.
- [29] K. Wasmer, C. Kenel, *et al.*, "In situ and real-time monitoring of powder-bed am by combining acoustic emission and artificial intelligence," in *International Conference on Additive Manufacturing in Products and Applications*, Springer, 2017.
- [30] P. K. Rao, J. P. Liu, *et al.*, "Online real-time quality monitoring in additive manufacturing processes using heterogeneous sensors," *Journal of Manufacturing Science and Engineering*, 2015.
- [31] H. Sun, P. K. Rao, *et al.*, "Functional quantitative and qualitative models for quality modeling in a fused deposition modeling process," *Transactions on Automation Science and Engineering*, 2018.
- [32] K. Bastani, P. K. Rao, *et al.*, "An online sparse estimation-based classification approach for real-time monitoring in advanced manufacturing processes from heterogeneous sensor data," *IIIE Transactions*, 2016.
- [33] S. Rokka Chhetri, S. Faezi, and M. A. Faruque, "Digital twin of manufacturing systems," *CECS Technical Report CECS TR-17-07*, 2017.
- [34] A. Faruque, M. Abdullah, *et al.*, "Acoustic side-channel attacks on additive manufacturing systems," in *International Conference on Cyber-Physical Systems*, IEEE, 2016.
- [35] S. Faezi, S. R. Chhetri, *et al.*, "Oligo-snoop: A non-invasive side channel attack against dna synthesis machines," *Network and Distributed System Security Symposium (NDSS)*, 2019.
- [36] L. Zhu, C. Johnsson, *et al.*, "Key performance indicators for manufacturing operations management in the process industry," in *Industrial Engineering and Engineering Management (IEEM)*, IEEE, 2017.
- [37] M. Christ *et al.*, "Distributed and parallel time series feature extraction for industrial big data applications," *arXiv preprint:1610.07717*, 2016.
- [38] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation," in *Encyclopedia of database systems*, Springer, 2009.
- [39] J. Elith, J. R. Leathwick, and T. Hastie, "A working guide to boosted regression trees," *Journal of Animal Ecology*, 2008.
- [40] S. R. Chhetri, A. Canedo, *et al.*, "Kcad: kinetic cyber-attack detection method for cyber-physical additive manufacturing systems," in *International Conference on Computer-Aided Design*, ACM, 2016.
- [41] Postscales, "Iot Sensors and Actuators." <https://www.postscales.com/trackers/video/the-internet-of-things-and-sensors-and-actuators/>, 2018.
- [42] Ultimaker, "Ultimaker 3." <https://ultimaker.com/en/products/ultimaker-3>, 2018.
- [43] Thingiverse, "Calibration collection." <https://www.thingiverse.com/glitchpudding/collections/calibration>, 2017.
- [44] H. Wu, Y. Wang, and Z. Yu, "In situ monitoring of fdm machine condition via acoustic emission," *The International Journal of Advanced Manufacturing Technology*, 2016.
- [45] A. Boschetto and L. Bottini, "Accuracy prediction in fused deposition modeling," *The international journal of advanced manufacturing technology*, 2014.
- [46] H. Wu, Z. Yu, and Y. Wang, "Real-time fdm machine condition monitoring and diagnosis based on acoustic emission and hidden semi-markov model," *Journal of Advanced Manufacturing Technology*, 2017.
- [47] J. Yoon, D. He, *et al.*, "A phm approach to additive manufacturing equipment health monitoring, fault diagnosis, and quality control," in *Prognostics and Health Management Society Conference*, Citeseer, 2014.