Received: 30 January 2022 | Revised: 8 May 2022 | Accepted: 16 May 2022

DOI: 10.1002/amp2.10129

JOURNAL OF ADVANCED MANUFACTURING AND PROCESSING

WILEY

RESEARCH ARTICLE

Special Issue

Smart connected worker edge platform for smart manufacturing: Part 1—Architecture and platform design

Yoon G. Kim[1] | Richard P. Donovan[1] | Yutian Ren[2] | Shijie Bian[3] | Tongzi Wu[4] | Shweta Purawat[5] | Anthony J. Manzo[6] | Ilkay Altintas[5] | Bingbing Li[7] | Guann-Pyng Li[1]

[1]Calit2, University of California Irvine, Irvine, California, USA

[2]EECS, University of California Irvine, Irvine, California, USA

[3]Department of Mathematics, University of California Los Angeles, Los Angeles, California, USA

[4]Autonomy Research Center for STEAHM, California State University Northridge, Los Angeles, California, USA

[5]San Diego Supercomputer Center, University of California San Diego, La Jolla, California, USA

[6]Engineering and Technology Group, The Aerospace Corporation, El Segundo, California, USA

[7]Manufacturing Systems Engineering and Management, California State University Northridge, Los Angeles, California, USA

Correspondence

Guann-Pyng Li, Calit2, University of California Irvine, Irvine, California, USA.
Email: gpli@calit2.uci.edu

Funding information

Office of Energy Efficiency and Renewable Energy, Grant/Award Number: DE-EE0007613

Abstract

The challenge of sustainably producing goods and services for healthy living on a healthy planet requires simultaneous consideration of economic, societal, and environmental dimensions in manufacturing. Enabling technology for data driven manufacturing paradigms like Smart Manufacturing (a.k.a. Industry 4.0) serve as the technological backbone from which sustainable approaches to manufacturing can be implemented. Unfortunately, these technologies are typically associated with broader and deeper factory automation that is often too expensive and complex for the small and medium sized manufacturers (SMMs) that comprise the majority of manufacturing business in the USA and for whom their most valuable asset are the people whose jobs automation while replace. This paper describes an edge intelligent platform to integrate internet-of-things technologies with computing hardware, software, computational workflows for machine learning, and data ingestion, enabling SMMs to transition into smart manufacturing paradigms by leveraging the intelligence of their people. The platform leverages consumer grade electronics and sensors (affordable and portable), customized software with open source software packages (accessible), and existing communication network infrastructures (scalable). The software systems are implemented via Kubernetes orchestration of Docker containerization to ensure scalability and programmability. The platform is adaptive via computational workflow engines that produce information from data by processing with low-cost edge computing devices while efficiently accessing resources of cloud servers as needed. The proposed edge platform connects workers to technological resources that provide computational intelligence (i.e., silicon-based sensing and computation for data collection and contextualization) to enable decision making at the edge of advanced manufacturing.

KEYWORDS
artificial intelligence, industry 4.0, smart manufacturing

1 | INTRODUCTION

Artificial intelligence (AI), edge intelligence, and other similar terms of art typically associated with Smart Manufacturing are commonly understood to imply a high degree of autonomous information technology (IT) functionality with respect to data acquisition, information production (i.e., contextualization), encode/decode for transmission/acquisition and activation with respect to engaging operation technologies (OT) (e.g., motors, conveyors, and ovens). While the emerging IT synergies between sensing, computing and transmission have made information theoretic[1] Smart Manufacturing paradigms connecting IT and OT increasingly more common, it remains the case that human workflows are central to the operations of most manufacturing which is largely comprised of SMMs whose most valuable asset are their people. For these SMMs, capital costs, complications of deploying advanced methods and issues of legacy manufacturing systems have precluded broad implementation of Smart Manufacturing. In this paper we describe methods and technologies that have been specifically developed to connect the intelligence of smart manufacturing (i.e., sensors acquiring data and computing for contextualization via artificial intelligence) with the intelligence of people (five senses for data acquisition and computing for contextualization via biological intelligence) in order to address these challenges.

The Smart Connected Worker Edge Platform (SCW-EP) for Smart Manufacturing addresses these challenges through Affordable, Scalable, Accessible, and Portable (ASAP) methods combining low-cost hardware with scalable hardware/software system architectures that utilize causal inference to couple human workflows to manufacturing equipment operational states and enables real time adaptation. In this paper, we describe the system architecture through which this vision is implemented. Core functionality of the ASAP approach for connecting the intelligence of workers to machine/edge intelligence coupling IT and OT in Smart Manufacturing paradigms includes intelligent (AI enabled) multi-agent models for non-intrusive workflow monitoring which is deployed for real-time machine state monitoring and fault detection. The ASAP approaches described in this paper support the edge intelligence (machine and human) needed for capturing data for information production (conceptualization) via real-time human machine interaction monitoring and deployment of scientific workflows to produce affordable AI (i.e., machine learning) for advanced manufacturing.

In Section 2, System Architecture (SA), System Requirements Specification (SRS) and the SRS methodology are described. The SA is comprised of a software stack and hardware specifications that enable SCW's access to advanced simulation and data acquisition at the edge of advanced manufacturing at reasonable costs. In addition, a methodology for extending the existing architecture to other use cases is presented. Section 3 presents the development of a multi-agent modeling approach to non-intrusive human workflow monitoring. This subsystem provides information (contextualized data) to inform human centered methods for establishing causal relations between the actions of people and the physical assets of a manufacturing plan. In Section 4, methods for establishing the state configuration of manufacturing systems are presented. This information provides the intelligence about machine operations needed to establish the causal relationships between human actions and the physical assets of the manufacturing plan. In Section 5, we describe how the elements that acquire data are connected to produce the correlated human-machine information needed for intelligent decision making at the edge of advanced manufacturing. These human-machine interactions are monitored and contextualized via AI to establish correlations in real time. Section 6 describes the methodologies through which the complexity of developing these event driven AI algorithms is managed via scientific workflow engines.

Results and discussion of these development efforts are presented in Section 7. Application of information production associated with human-machine interactions in 3D printing with extensions to Augmented Reality are presented as well as implementation of the multi-agent methods for correlation of human workflows, energy consumption and anomaly detection. The methods for adaptive information production with AI algorithms are described with respect to creating reduced order models of electromagnetic fields associated with WiFi signals that are essential to transmission/acquisition of encoded/decoded information between the edge and the cloud. In-process information production via application of scientific workflows for AI algorithm generation are presented with respect to laser processing techniques typically associated with surface roughness from laser refraction (i.e., speckle) images. Conclusion and the opportunities for future work are briefly discussed in Section 8. The SCW-EP is focused on the essential role that people can play in the production and use of information at the edge of advanced manufacturing. Future research directions will continue to exploit emerging technologies that enable broader implementation of ML to augment the native intelligence of people in advanced manufacturing contexts.

2 | SYSTEM ARCHITECTURE AND SYSTEM REQUIREMENTS SPECIFICATION

2.1 | Objectives of smart connected worker edge platform and architecture design

The proposed edge intelligent platform is designed to integrate IoT technologies with computing hardware, software, computational workflows, and data acquisition. The goal of the platform is to optimize manufacturing workflows for increasing energy productivity as well as other important sustainability dimensions such as operational safety and cybersecurity.

The system architecture design of the platform begins with generation of the SRS that provides requirements for developing a system or platform. An important feature of the SRS is the Functional Requirements, which define the functions of the system. The requirements are concerned with the identification of system behavior[2] with respect to inputs and outputs.

The requirements are derived from a process that includes: (1) development of a comprehensive list of use cases and actors, (2) production of system reference diagrams, and (3) specification of functional requirements definitions.[3]

Details of these steps as applied to the Smart Connected Worker platform are provided in the following sections.

2.2 | Use cases and actors of SRS

To create a list of functional requirements for SCW systems, the proposed system is analyzed to identify all use cases and the associated actors involved in the manufacturing workflow and processes. Use cases are the description of actions or steps explaining the interactions between a person (operator, or supervisor) and machine (component, module, subsystem, or system). An actor is a role played by a person, or machine. Sources include the use cases and associated actors of participating organizations, Aerospace Corporation, CSU Northridge, General Mills, Honeywell, and UC Irvine. Not all use cases are included in the functional requirements as some of them are unique due to the type of their industries. A set of use cases in common among the major use cases is identified to create a list of functional requirements. A selection of use cases and associated actors are listed in Table 1 to illustrate the process.

From this example shown in Table 1, the actors are Operator/Worker, Control Panel of Machines, Camera, Video Processing Module (A), Edge Intelligence Unit, AC Power Meter (Smart Meter), User Terminal/Headset, Sensor, Data Services (A), and Data Services (B).

TABLE 1 Use cases and actors (selected)

ID	Use cases	Actor sending data	Actor receiving data
1	Power on machines	Operator/worker	Control panel of machine
2	Camera reading	Camera	Video Processing Module (A); Edge intelligence unit
3	Energy meter reading	AC power meter (smart meter)	Edge intelligence unit
4	On-site status monitoring	Edge intelligence unit	User terminal/headset
5	Notification of machine stoppage	Camera; sensor	Operator; control system; edge intelligence unit (fault detection)
6	Identification of anomalous machine behavior	Camera; sensor	Operator; control system; edge intelligence unit (CAAH)
7	Energy disaggregation	Camera; AC power meter	Edge intelligence unit (energy disaggregation)
8	Worker action recognition	Camera; AC power meter	Edge intelligence unit (WARM)
9	Process control of surface roughness of metals	Camera; sensor	Edge intelligence unit (Laser)
10	Analysis of operator Utilization	Video processing module (A)	Data services (A)
11	Neural network training	Edge intelligence unit	Data services (B)

2.3 | System reference diagram

By exercising the definitions of use cases and associated actors, a reference diagram for the SCW edge platform was developed and is illustrated in Figure 1. The figure shows the system reference diagram of the SCW edge platform with actors highlighted in blue and green (e.g., Worker, Machine, Edge Intelligence Unit, and subsystems), which illustrate data flows between edge and cloud as an example. On-site subsystems are the Manufacturing Facility, Sensor, and Meter Subsystems (which reside on factory premises at the edge), whereas the off-site subsystems are Data Services (A)/(B) Subsystems and Remote User Unit in the cloud (which can be accessed remotely through a wide area network [WAN].) The Manufacturing Facility subsystem includes Worker and Machine actors, with cameras and sensors integrated into the Sensor subsystems to augment the intelligence of workers. The Meter subsystem includes Smart Meters (AC power meters). The data acquired from the sensors and meters are fed into an Edge Intelligence Unit in which data is processed to provide information for monitoring workflows and data streams to improve manufacturing sustainability (e.g., increasing energy productivity.) The Video Processing Module (A) with Data Services (A) is an example of manufacturing operation management systems. An instance of Data Service (B) is a cloud-computing machine on remote servers.

2.4 | Defining functional requirements

It is important to have a clear system architecture to understand data flow between actors and subsystems. Based on the use cases and actors identified along with the system reference diagram, the functional requirements most germane to Smart Manufacturing are defined as follows:

- Operator shall be able to power on machines by utilizing Control Panel;
- Control Panel shall receive a power-on command from Operator;

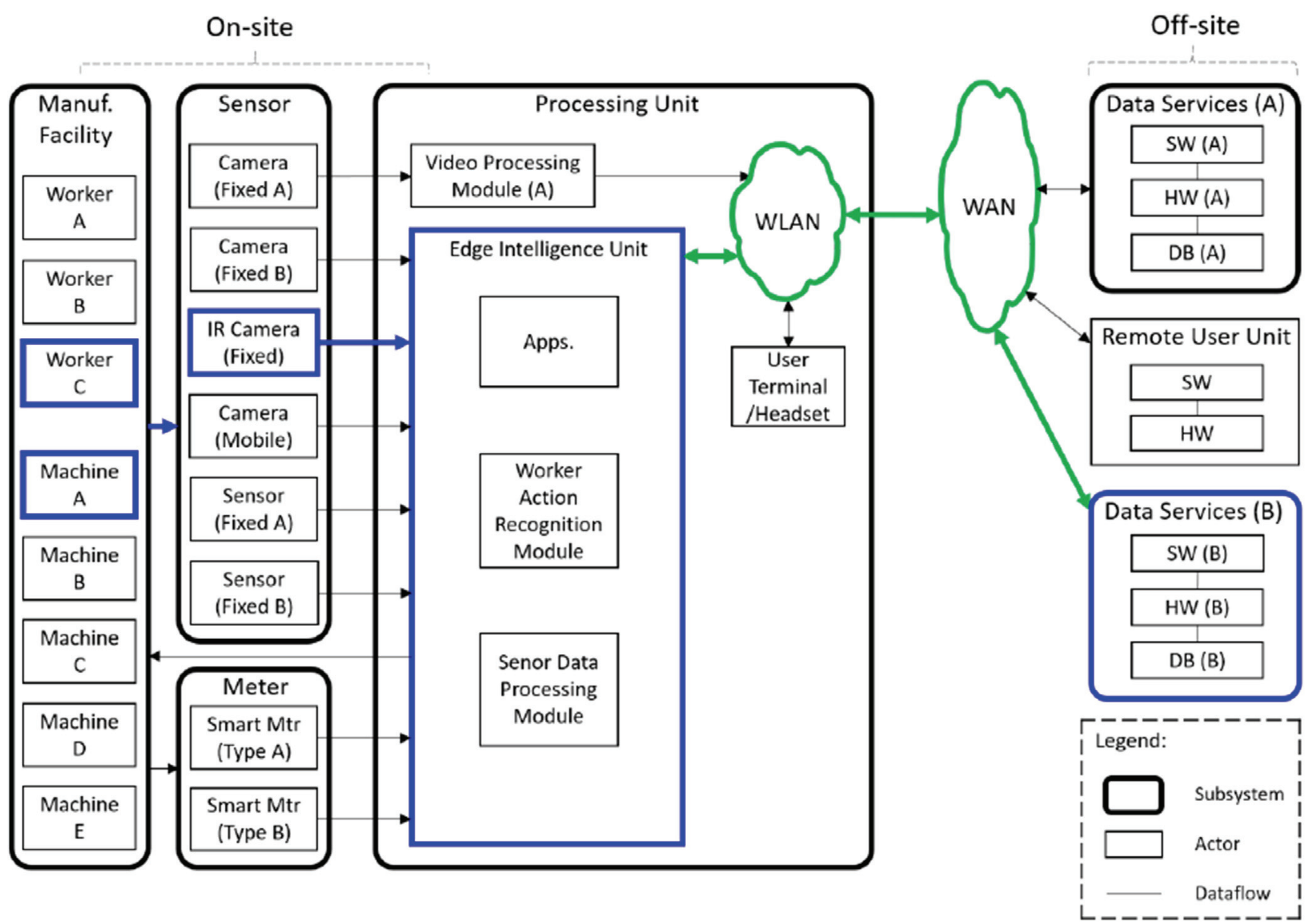

FIGURE 1 System reference diagram.

- Camera shall read the status of Operator and Machine and send it to Video Processing Unit (A) and Sensor Data Processing Module in Edge Intelligence Unit;
- Video processing unit (A) and Edge Intelligence Unit shall receive vision data from Camera;
- AC power meter (smart meter) shall read voltage and current usage of machines at Manufacturing Facility subsystem and send the data to Edge Intelligence Unit;
- Edge intelligence unit shall receive meter readings from AC Power Meter;
- Edge intelligence unit shall read cameras and sensors and send the data to user terminal/headset for on-site status monitoring;
- User terminal/headset shall display on-site status received from Edge intelligence unit;
- Camera or sensor shall read the status of machine and send it to operator, control system, or Edge intelligence unit;
- Operator, control system, or Edge intelligence unit shall receive the data from camera or/and sensor for notifying machine stoppage and identifying anomalous machine behavior;
- Camera and AC power meter shall read the status of machine and send it to Edge intelligence unit;
- Edge intelligence unit shall receive the data from camera and AC power meter for energy disaggregation and recognizing worker's action;
- Edge intelligence unit shall receive the data from camera and sensor for the process control of surface roughness of metals;
- Data services (A) shall receive data from video processing module (A) through WAN for analyzing worker utilization; and
- Edge intelligence unit shall read camera and sensor; and send the data to data services (B) through WAN for training neural networks (NNs).

In the later sections below, we utilize the reference diagram and functional requirements to design each building blocks' architecture and scientific workflows.

The SCW platform is designed to integrate IoT technologies with customized computing hardware and software. Edge Intelligence Unit and Data Services (B) in the reference diagram is the core blocks that have been developed and customized to achieve this integration. The software of the two blocks consists of the software that we developed, open-source software packages including Linux distribution, Kepler scientific workflow system, and Docker container orchestration. The computing hardware of the Edge Intelligence Unit and Sensor subsystem includes consumer grade electronics and sensors.

3 | NON-INTRUSIVE WORKFLOW MONITORING: MULTI-AGENT MODELING

Several methods have been proposed to monitor the material flow, process status, workers' location, or machines' status, including using Radio Frequency Identification (RFID), Wireless Sensor Networks (WSN), and directly reading Programmable Logic Controller (PLC) data. These approaches require significant sensor installation, comprehensive wireless communication infrastructure and professional IT personnel, which is a challenge for SMMs. It is especially difficult to design a unified interface for accessing PLC data due to the variety of machines in different manufacturing sectors or legacy machines lacking modern interfaces. In addition, current supervisory control and data acquisition (SCADA) systems are capable of terminating ongoing process when detecting anomalies but lose the external information associated with the anomalies source. For example, a conveyor can be stopped due to accumulation of waste materials, but SCADA cannot capture the root cause of the accumulation. Therefore, an anomaly replay system that captures information from outside the control system boundary, outside-in information, is needed to assist the identification of the cause of anomalies at the edge.

To overcome these issues, a non-intrusive industrial workflow monitoring system has been developed for operation at the edge to provide real-time machine and worker states without the need for accessing PLCs. The scalable software architecture can be extended to accommodate a flexible number of machines and workers as needed. Anomaly identification is achieved by leveraging a worker's intelligence to monitor the workflow and identify anomalous events from the feedback of existing control systems in combination with their knowledge and experience.

Several outside-in sensors at the edge are selected to capture data regarding manufacturing processes. For this application, outside-in sensors are sensors that capture information from outside of normal control system (e.g., PLCs) architecture in order to independently capture worker-machine interactions as well as system emissions (e.g., vibration associated acoustics, or motor electromagnetic signals). Advanced machine learning (ML) methods are applied to perceive the status of workers and machines. In addition, the high-level software architecture is modeled with multi-agent systems, where each self-contained software agent can provide autonomy to achieve certain tasks and multiple software agents interact to achieve more complex functions to provide scalable and easily deployable solutions. To capture workers' feedback, we designed a connected worker technology that virtually connects workers through their body gestures to avoid any physically connected wearable devices. This virtual connection can directly transmit simple messages from workers while not interrupting workers' normal operation.

Conventional manufacturing workflow and process monitoring has been widely investigated, and with pervasive IoT devices the real-time data collection of industrial workflows have become possible. In 2009, Lu et al. implemented a Wireless Sensor Network with torque transducer and tachometer as a sensor node to analyze motor energy signature in a non-invasive way.[4] In 2012, Hou et al. applied various sensors with different sensing modalities (current transducer, vibration sensors and accelerometers) as WSN nodes to monitor conditions of motor and other tools.[5] In 2014, Hu et al. investigated the use of RFID to monitor the operator's location and product's location along an assembly line and with this information the state of production can be inferred without the equipment's state detection.[6] In 2020, Qian et al. utilized RFID tags and Near-Field Communication (NFC) to capture the arrivals of assemblies, tools, and operators to do workflow and process optimization.[7] Beyond including external sensing technologies, several studies attempted to directly read PLC data through standard protocol to capture machines' status. In 2010, Vijayaraghavan et al. proposed to read PLC data through MTconnect (ANSI/MTC1.4-2018) protocol and to use a single meter to monitor a single machine's energy usage profile which is then correlated with the machine events from PLC data.[8] In 2017, Wu et al. investigated a SMM case by using more than 50 sensor nodes including accelerometers, current transducer, and PLC data to monitor pump status (on/off) and do prognosis.[9]

This section presents technical details of the software system modeling including data acquisition, data processing, and components for actionable intelligence generation. The software system model is scalable for edge computing platform.

Our approach utilizes the Prometheus methodology[10] illustrated in Figure 2 that describes the modeling of the entire software system with six agents with specific proactive and reactive functionalities. The system includes an Orchestrator Agent, two ML agents, a Correlation Agent, and two Manufacturing Application Agents, along with their interactions through predefined message formats or databases. The system is adaptable to a variety of scenarios that could include ML agents and process specific manufacturing agents. Agent modeling provides easily scalable solutions to monitor various numbers of workers and machines by duplicating these agents while changing certain parameters.

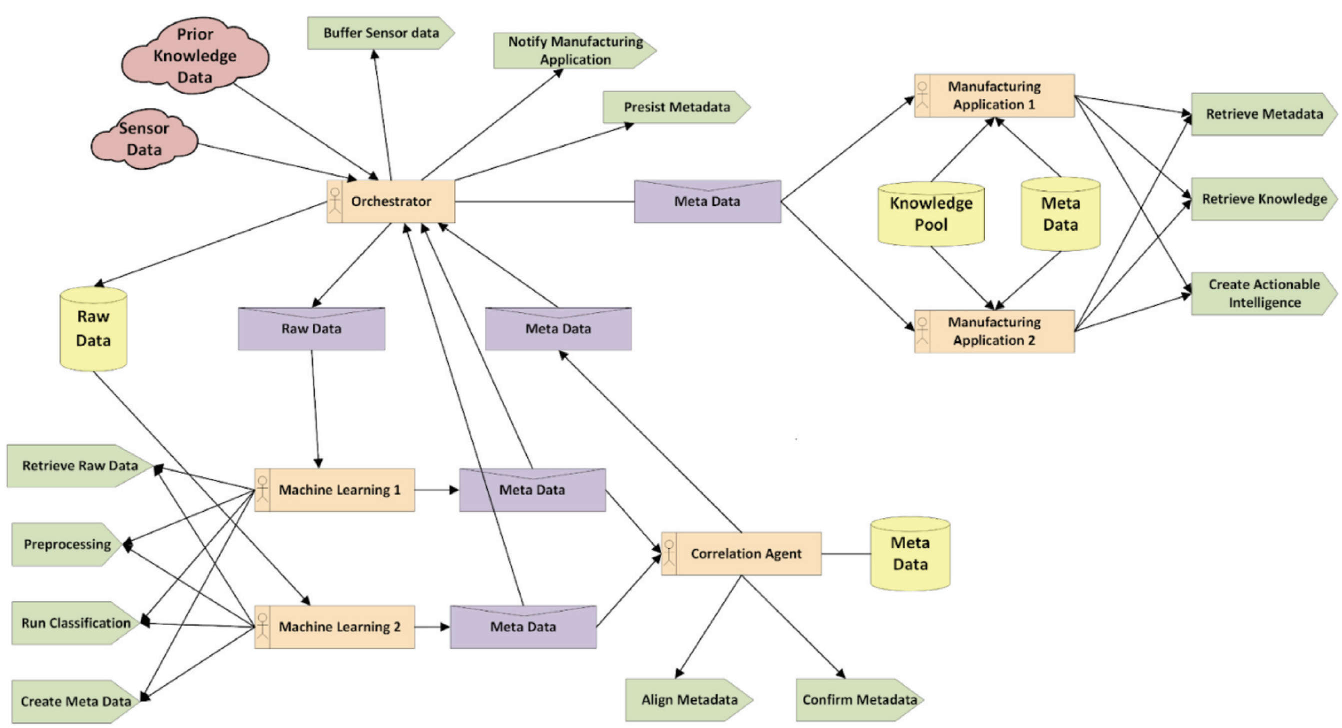

FIGURE 2 An overview of the software agent model with the actions of each agent and the interactions between agents through defined messages.

3.1 | Software agent model

The Orchestrator Agent serves as the adhesive for handling data streaming, data transmission, data persistence, and agent notification. It can process outside-in sensor data streaming and inside-out internal metadata generated through the processing pipelines and notify corresponding agents depending on the data type. Specifically, the Orchestrator Agent monitors the available streaming sensor data, transduces them into proper formats based on the needs of ML agents, and transmits them to the ML agents. Simultaneously, the raw sensor data is stored into a database or local file system depending on the data type. In addition, the orchestrator listens to the ML agent and following the Correlation Agent for the metadata messaging, forwards the metadata message to alert the Manufacturing Application Agent and stores them into a database. In addition, available prior knowledge regarding the manufacturing processes, such as Standard Operating Procedure (SOP) and work schedule, is captured by the orchestrator to be transmitted to other agents. The Orchestrator Agent bridges the physical environments, the backend data processing, and the frontend applications to provide necessary message passing and isolated software environments to reduce interference, enhance programmability and ensure scalability (e.g., agents can reside in an isolated environment, such as a Docker container, and can be upgraded only considering the defined message formats without the need to interrupt other agents.)

The ML Agent 1 processes sensor data with advanced NNs to identify workers actions. It first needs to retrieve the raw sensor data from the orchestrator, preprocess raw data to remove improper samples or noise, run classification leveraging computational efficiency of ML, and create metadata based on the classification results. Next, the agent notifies the Correlation Agent and the Orchestrator Agent by sending the metadata to them. The ML Agent 2 carries out the same functions but focuses on different types of data or the same type of data but from different sensors. In some cases, there can be more ML agents processing data of different aspects of a manufacturing process as long as the data are causally correlated.

The Correlation Agent leverages well-established causality knowledge to correlate the detection results from ML agents. The causality knowledge can be simply understood as cause and effect relationships, where causes happen earlier than effects. By knowing the effect and the time interval (known as response time) between causes and effects, the cause can be inferred and vice versa. In manufacturing scenarios, especially for legacy machines without automation control, one of the causal relationships between workers and machines is that the worker interactions towards machines will cause the machine to change states, guided by the machine manual and SOP. In the simple case of one cause and one effect, the two ML agents capture information from the cause and effect side respectively. The Correlation Agent retrieves the metadata from the two ML agents, aligns them to the causal event timelines based on their own timestamps indicating the data capture time of physical worlds and the response time, and conducts cross confirmation of the two metadata. The function of alignment persists metadata if its counterpart is not ready. There are two cases: in the case of using cause to confirm effect, the decision needs to be made after waiting for certain response time to receive metadata from effect side; in the case of using effect to confirm cause, when the metadata from effect side is received, the decision is made by tracing back the metadata from the cause side saved in the buffer. After the alignment, the cross confirmation is made by using the prior knowledge of well-established casualties to enhance the fidelity and reliability of the detection results. New metadata is generated based on the confirmation and is sent to the orchestrator to notify the manufacturing applications.

3.2 | Manufacturing application agents

The Manufacturing Application Agents provide interfaces and actionable intelligence to workers or supervisors based on the metadata and knowledge database. Generally speaking, they are less computationally intensive compared to the ML agents and provide only high-level abstraction of acquired sensor data. One instance of the application agent is the non-intrusive workflow monitoring application. The agent can extract the real-time machine and worker states from the metadata and compare them to the work schedule(s) (e.g., multiple manufacturing workflows) saved in the knowledge database. This comparison can simply provide yes or no answers in response to the user query or can provide actual status of each machine and worker. In addition, users can query for historical workflow status and associated information such as energy consumption by interactive queries. This is achieved by accessing not only the metadata database but also the raw data database with computations of relational variables. Moreover, the user can opt to display the raw data or some intermediate metadata during the processing pipelines on the interface.

A second instance of the Manufacturing Application Agent is an anomaly detection function leveraging human cognition and intelligence. Workers can perceive and identify abnormal events during the manufacturing workflow execution based on accumulated experience and knowledge. When a worker observes an anomaly, the worker can execute a predefined gesture towards a camera sensor to trigger the system to collect and store the past data within a certain period from sensors, PLCs, or SCADA if available. According to user queries, the collected data, such as video snapshots, can be replayed one or multiple times at the user interface to help workers or supervisors analyze the causes. The collected data can subsequently be analyzed by unsupervised ML to uncover common features of repetitive abnormal events which possibly indicates the cause of the anomaly.

Privacy concerns in workplace environments are extremely important. The proposed system addresses these concerns by obscuring biometric identification as needed and by abstracting human features (e.g., skeletal representations) prior to storage for ML training purposes.

4 | REAL-TIME MACHINE STATE MONITORING AND FAULT DETECTION

Real-time fault detection and analysis to prevent device malfunction are indispensable elements of smart manufacturing systems. Typically, conventional devices lack the mechanism and functionality to report malfunctions to operators which is challenging because automatic feedback requires sophisticated coordination and data transmission between devices of different communication protocols. Overcoming these challenges requires that operators continuously monitor, detect, and report machine malfunctions manually, resulting in a waste of human labor. In comparison, the constantly evolving ML algorithms provide the alternative solution of detecting machine states and faults automatically without human intervention. With the SCW-PA, training of ML models normally performed on expensive cloud servers (at significant expense) can be done on lower cost edge platforms which can reduce the cost of computing. To address this, we propose an affordable and easily deployable edge-computing platform that utilizes ML algorithms (i.e., computer vision ML models) for the real-time identification of machine states and the detection of possible fault scenarios.

We used the YOLO[11] object detection ML model to locate the positions of the machine's (a 3D printer) major component. By passing the coordinates through a filtering algorithm, the most likely machine state is predicted, while possible fault scenarios are identified by examining the range of the components' locations. YOLO is a fully open-sourced ML model that supports Linux development environments and is therefore affordable and adaptable to other use cases. YOLO's well-developed API allows the user to train and test their customized ML models without significantly changing the model structure or hyperparameters and is therefore programmable. As long as the labeling and bounding box drawing is correct, the proposed platform can be easily applied to machines of all kinds and therefore, it is scalable.

Existing computer vision (specifically, object detection) ML models consist of two dominant types of architecture: the regional-based convolutional neural network (R-CNN)[12] and the Single Shot Detector.[13] The R-CNN combines selective search with traditional convolutional neural network (CNN) models to detect and segment objects of various sizes and features. The single-shot detector replaces regional proposals with predefined bounding boxes, thus speeding up the inference process but sacrificing accuracy. Since real-time monitoring is the goal of the proposed system, inference speed is prioritized over the accuracy, and the Single Shot Detector object detection model (i.e., YOLO) is adopted.

Object detection ML in particular has been broadly used for constructing smart manufacturing systems. Zhou et al.[14] proposed a hybrid deep NN model to identify the status of physical manufacturing environments and enable synchronization to virtual representations in real-time. To support the real-time analysis of manufacturing systems' multimedia inputs, Lasek[15] developed a real-time simulation framework that discussed the impacts of applying ML methods for the detection, analysis, and simulation of manufacturing components. We intend to further explore the potentials of object detection models by focusing on the real-time identification of machine states and fault scenarios. The primary components of the SCW platform for Real-Time Machine State Monitoring and Fault Detection are Preprocessing, Model Training, Detection/Analysis and Implementation as described below.

4.1 | Components for Real-Time machine state monitoring and fault detection

4.1.1 | Preprocessing

Before utilizing the ML model, image data needs to be collected from real-world working scenarios for training. Specifically, the image collection should center around the machine of interest and record as much detail of the machine components as possible. After collecting the images, the images are manually labeled with bounding boxes around major components and assigned corresponding labels. For our purposes, this approach is relatively laborious. It is important to note that it only requires one round of this labeling process before being fully functional and represents an opportunity for further developing methods through which this can be done automatically.

4.1.2 | Model training

The input to the YOLO model consists of the following elements: the digital image focused on the machine of interest captured by a camera, the ground truth bounding boxes, and the ground truth class labels assigned to each bounding box. Each bounding box is defined by its center horizontal and vertical coordinates and its width and height. The input image is then resized to 416×416 pixels for standardization and is split into 19×19 grid cells, each of which contains three predefined anchor-bounding boxes. Subsequently, the anchor boxes and the one-hot encoded class labels are passed into the deep CNN structure of YOLO V3, through which the feature maps are obtained. The bounding box closest to the ground truth is filtered out using Intersection over Union (IoU) during training. The final output of the YOLO model consists of the following elements: the processed test image with predicted bounding boxes, the corresponding predicted class labels, the confidence scores, and the coordinates of the bounding boxes.

4.1.3 | Detection and analysis

With the output from the YOLO model indicating the coordinates of each major component of the machine, a filtering algorithm is implemented to obtain the machine's status: the algorithm checks the coordinates of each component according to a predefined set of boundary regions. When the algorithm detects that a combination of machine components' regions meets the specific criteria for machine state transition, it will automatically filter out the most likely machine state and output the result. Similarly, if the algorithm detects that a machine component's coordinates exceed the region of normality, it will identify the scenario as faulty and record it.

4.1.4 | Implementation

In order to fully exploit the functionality of the proposed model, users need to collect their machine specific set of training images that capture unique features of target machines and major elements with high quality and minimal occlusion during relevant machine states. Subsequently, the user utilizes open-source labeling tools (such as Microsoft VoTT[16]) to create the ground truth bounding boxes and labels for the target machine's major components. The resulting dataset can be fed into a pre-installed YOLO model for training, and the criterion for machine state transitions can be set in the filtering algorithm. Note that in order to train the model using edge computing, the user can store the data and model required for training on a cloud platform (e.g., Google Colab or Nautilus GPU clusters) and utilize GPU resources at the edge to train the model. Finally, the user can insert the ML model's trained weights into the corresponding directory of the SCW edge computing devices to enable the functionality. If new datasets of the same target machine are created and made available, the user may retrieve the previously trained model weights, and perform additional training to update and refine the parameters.

5 | REAL-TIME HUMAN–MACHINE INTERACTION MONITORING

For smart manufacturing systems, monitoring human-machine interaction provides critical information with regard to sustainable operations. For example, understanding the relationship between human workflows and machine states enables analysis of energy consumption that would typically require individual metering of manufacturing equipment. By capturing human machine interaction (i.e., human workflows) and coupling it with edge computing, energy disaggregation (i.e., identifying individual machine energy profiles from aggregated metering) becomes an affordable method for reducing energy usage and increasing energy productivity of individual machines at SMMs.

The SCW platform leverages the computing power of ML models embedded on edge computing devices to replace the human labor that is conventionally required for the real-time monitoring and documentation of human-machine interactions. Monitoring the interactions is essential for performing energy disaggregation for the optimization of energy usage of manufacturing machines.

For real-time human–machine interaction we utilize the advanced CRAFT[17] text detection model to identify the output of machines by recognizing the displayed texts on the interactive panels and a novel color identification algorithm on the button regions to monitor and identify the input of operators. Text detection and recognition, a crucial part of Computer Vision, has already been extensively explored by researchers. There have also been efforts in extending text recognition models to user-friendly real-world applications. For example, Naiemi et al.[18] proposed a pipeline framework that unified text detection and recognition that could be used to create vision assistance systems. Beyond recognizing English language letters, Wang et al.[19] overcame the difficulty imposed by the complicated nature of Chinese characters and used deep-coupled alignments to improve detection accuracy. Finger detection is also a popular topic for both Computer Vision and real-world applications. Karam et al.[20] adopted depth sensors to detect finger clicks accurately without being restricted by fingers' motions or relative positions. Caputo et al.[21] created a novel and comprehensive benchmark for evaluating finger detection models related to gesture detection and recognition. However, little work has focused on the potential application of finger detection and recognition to empower manufacturing systems. Therefore, our proposed work intends to exploit the powers of finger detection in a novel way to improve human-machine interaction monitoring.

The primary components of the SCW platform for Real-Time Human-Machine Interaction Monitoring and Fault Detection are Text Detection/recognition and Finger Position Identification as described Section 5.1.

5.1 | Components for real-time human–machine interaction monitoring and fault detection

5.1.1 | Text detection and recognition

In order to monitor the human-machine interaction, we choose the visual perspective of the operator as illustrated in Figure 3. Videos and images are captured from a camera positioned on the head of the operator.

During a valid process of human-machine interaction, we mainly focus on the interactive panels. The first step would be to locate the text regions of interest. We used OpenCV's detecting contours to identify all text display regions on the input image and then applied a K Means clustering algorithm on the area of the text display regions to filter out the regions of interest. Subsequently, the CRAFT text detection and recognition algorithm was applied to the selected regions, and the corresponding text outputs were collected. This novel way of first filtering out the text regions instead of performing text recognition globally on the entire image effectively reduces the effect of background noise on the text detection model and allow the more flexible recognition and output of selected text regions. Finally, the identified texts were collected, each assigned a corresponding text display region ID label.

5.1.2 | Finger position identification

With the text regions located using OpenCV, we move on to locate the corresponding press button regions. By carefully observing the relative positions of the press buttons regarding the text display regions, coordinate-transformation matrices can be calculated and applied to derive the location of the press buttons on the interactive panel. Subsequently, we analyze the color changes in the press button regions. Since the color of the buttons demonstrate changes when covered by a human finger, we leveraged OpenCV's functionality of extracting RGB color values from selected regions to monitor press button region RGB values changes. Through this novel way of detecting color changes instead of applying an entire cumbersome finger detection model, we significantly reduce the size and complexity of the proposed framework and allow for easier distribution and scalability.

When the panels are malfunctioning, protocol-based methods could be an alternative solution. We experimented

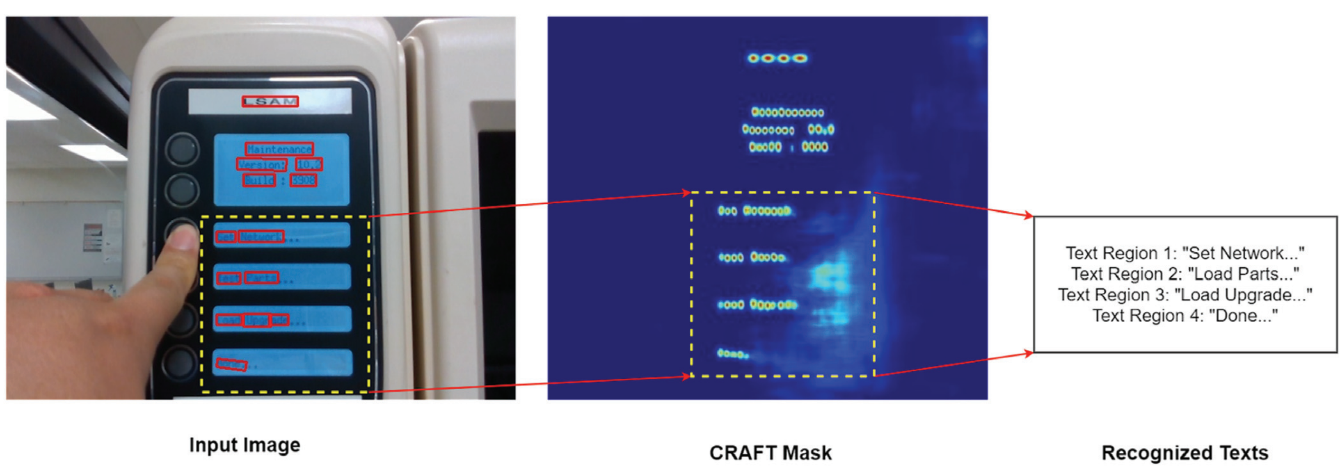

FIGURE 3 Text detection and recognition of 3D printer control panel.

with MTConnect (ANSI/MTC1.4-2018) and Tera Term terminal emulators, both of which are industry-level protocols that enable the inter-communication of devices.

6 | AFFORDABLE AI FOR ADVANCED MANUFACTURING THROUGH SCIENTIFIC WORKFLOWS

The manufacturing sector is at the cusp of a grand transformation where AI and IoT are likely to seamlessly integrate and assist humans to boost precision of actions, improve the pace of diagnostic root-cause analysis, and reduce variance of manufactured output while improving operational safety manifolds.[22]

During the last decade, most AI driven industry benefits have been harnessed by large-scale organizations. Small and medium scale industries are particularly facing challenges in adapting AI for their operational benefit. This gap is rapidly increasing due to budget limitations, cost of retraining workforce, and high capital demands of compute resources necessary to deploy and operate AI techniques locally in a manufacturing environment. AI in manufacturing can lead to a more productive, cost-effective, and safer work environment. It can boost operational efficiency by reducing cost-per-unit produced and lower demand pressures on raw resources consumed by manufacturing industries.[23]

Scientific Workflows are directed acyclic graphs (like flowcharts) that enable integration of different software codes in a modular manner which can be scaled on demand using the in-built compute and data parallelism. These containerized workflows act as automated tools that can be rapidly deployed and can be executed on a variety of hardware devices making them suitable for edge computing platforms. Scientific Workflows can act as an abstraction layer to hide complex data and AI technologies in the backend. This makes adoption of AI technology easy to use by workers without need of technical expertise. The Kepler scientific workflow system[24,25] is open source software making it affordable for SMMs. Kepler scientific workflows are designed in a modular manner to increase adaptability and reuse of cutting edge Deep Learning applications to eventually deliver actionable intelligence on the manufacturing floor.

6.1 | Objectives of scientific workflows

The goal is to build an affordable AI system for small and medium scale industries, so operational teams across the globe can leverage enhanced intelligence of AI-in-the-loop manufacturing. The design goal is to create an AI powered edge-computing framework that is maintainable, upgradable, intuitive to use, energy efficient and can adapt to diverse manufacturing environments.

For SMMs, we designed a workflow-driven advanced manufacturing architecture to provide affordable, scalable, accessible, and portable platforms. We demonstrate the implementation of the proposed framework with Kepler workflows that train AI models on high performance computing (HPC) in the Cloud and provide intelligence for improved decision making at the edge.

6.2 | Multi-layered scientific workflows architecture

The architecture is driven by scientific workflows to bring AI to the edge devices in a manner that is maintainable, upgradable, and efficient in data requirements for decision-making. At the core of the architecture is the Kepler scientific workflow system, which serves to encapsulate and integrate AI tools in a simple, intuitive, and easy to use interface layer. Kepler scientific workflows are designed in a modular manner to increase adaptability and reuse of cutting edge deep learning applications to deliver actionable intelligence on the manufacturing floor.

Figure 4. describes the modular multi-layered architecture that includes: (1) A core-computing layer that delivers the necessary hardware capabilities at scale and (2) a front-end layer that sits at the manufacturing site. The data ingestion module provides multi format data processing capability. The next layer provides data wrangling tools for resampling, cleaning, and parametrization of the input signals. The third module enables loss optimization of the ML model based on the tuning method chosen such as stochastic gradient descent, adaptive learning rate optimization, and so forth. Once the fine-tuning of the ML model reaches stabilization, the model can be deployed to the edge devices. A search index is maintained and updated for fast information retrieval. The architecture is robust to changes in hardware due to containerized code deployment.[26] We have carefully designed the system using the open source technologies actively supported by a diverse set of developers. This ensures long-term scalability of our system for low-resource small to medium sized companies.

7 | RESULTS AND DISCUSSION

The development of the Smart Connected Worker system and subsystem architectures is based upon use cases

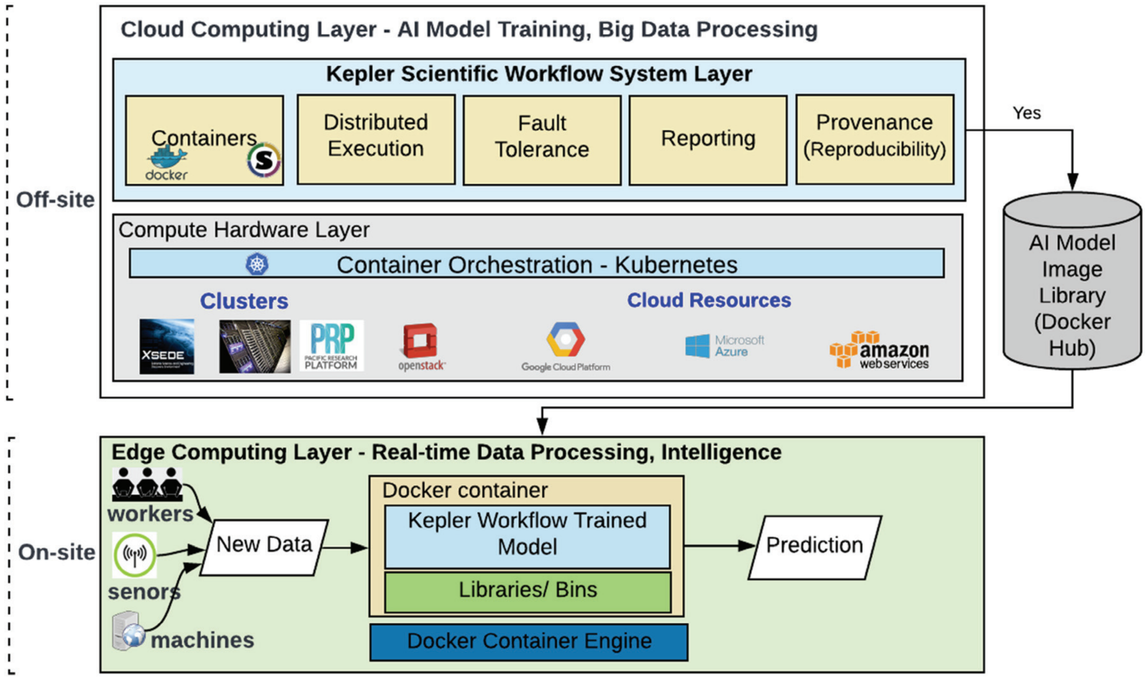

FIGURE 4 Multi-layered scientific workflow architecture.

derived from the collective experience of five CESMII members who participated in the Smart Connected Worker roadmap project. These use cases provide the framing for identification of actors and actions that form the basis of the functional requirements that are used to establish the SCW System Requirements Specification from which the system architecture was developed. The system architecture was used to implement a multi-agent model to capture the causal relationship between human workflows and energy consumption and to leverage advanced manufacturing workers intelligence for effective decision making at the edge of manufacturing. Power meters are used to capture power consumption signals from machines and cameras to capture worker actions. A machine-learning pipeline of skeleton-based action recognition is implemented to recognize worker actions with respect to machine states.[27] Concurrently, an unsupervised method is applied on the power signal to detect the state transition of specific machine components, as demonstrated in Reference [28] The design methodology of the two ML agents leverages the causality underlying worker machine interactions to develop two applications. The first application provides a non-intrusive real time monitoring function of the manufacturing workflow with interactive features. The second application designs a simple messaging system for workers to report the anomalies they observed by a predefined alarm gesture - crossed-arms-above-head. When this worker state is detected, the system is triggered to save the past data for future analysis and for the identification of anomaly causes. The worker action recognition algorithm of the non-intrusive workflow monitoring is composed of a cascaded NN where the first NN is a pose estimation software to extract the skeletal representations of human bodies from raw videos.[27] Compared to raw videos, the skeletal representations that are matrices of ($25 \times 3 \times$ number of people in the scene) need much less bandwidth for data transmission. The video streams we tested for the worker action recognition algorithm and object detection are 640×480 at a rate of 15 frames per second (about 0.7 Mbps using H.264 coding). This requirement is sufficient which fits well within the typical local area network bandwidth. Based upon typical power ratings for CPUs and GPUs (approximately 350 W combined), with the schema that we propose (which can use a single camera to monitor multiple machines and a single computing unit to service multiple cameras) we estimate that the net energy savings (e.g., with regard to machine idle time management) can be in excess of five times the energy cost of sensing and computing (i.e., intelligence).

The system architecture was also utilized for a real-time 3D printing use case as illustrated in Figure 5. For this application a YOLO-based object detection framework takes as input the annotated images collected from real-world working scenarios, trains on cloud platforms, and deploys on edge computing devices for the real-time detection of machine states and fault scenarios.

The proposed method is flexible to the heterogenous environment of SMMs and can be readily scaled to detect general machines by adjusting bounding boxes and assigned labels during local dataset construction. This

JOURNAL OF ADVANCED MANUFACTURING AND PROCESSING — WILEY

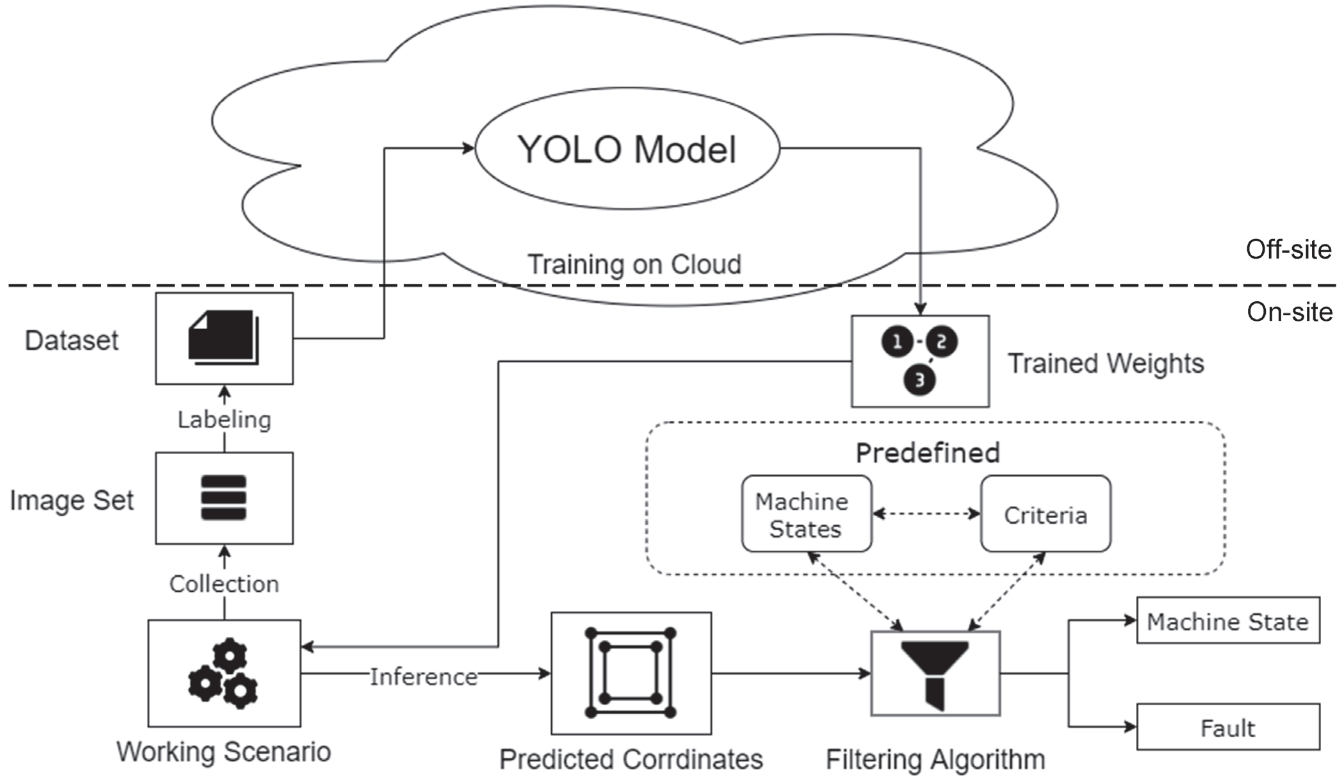

FIGURE 5 Real-time machine state monitoring and fault detection.

FIGURE 6 Demonstration of the interactive VR guidance system.

framework achieved exceptional accuracy and inference speed in the real-time monitoring of a 3D printer's machine states, as demonstrated in.[29] Furthermore, the proposed framework is also adaptable since the trained model could be further improved (either at the edge or in the cloud) when new datasets are made available.

The proposed framework of the human-machine interaction monitoring has been exercised to create an interactive VR guidance system for users of the 3D printer as shown in Figure 6. A Hololens attached to the operator's head captures the operator perspective and the proposed model can deliver the finger and text recognition outputs to the VR displayed in front of the operator. By analyzing the predicted output, the interactive real-time guidance system prompts the user for the subsequent instructions and record the user's actions in real-time for monitoring and later analysis.

With the proposed methodology, the user can also refer to the hints and cues displayed on the Hololens to operate the 3D printer correctly and more efficiently. Experimental results with the use case of 3D printer operation proved to benefit both the operators in their operation of the machine and the supervisors in the real-time supervision of the operators' actions. Comparison with human baselines demonstrated the effectiveness and efficiency of the proposed methodology.

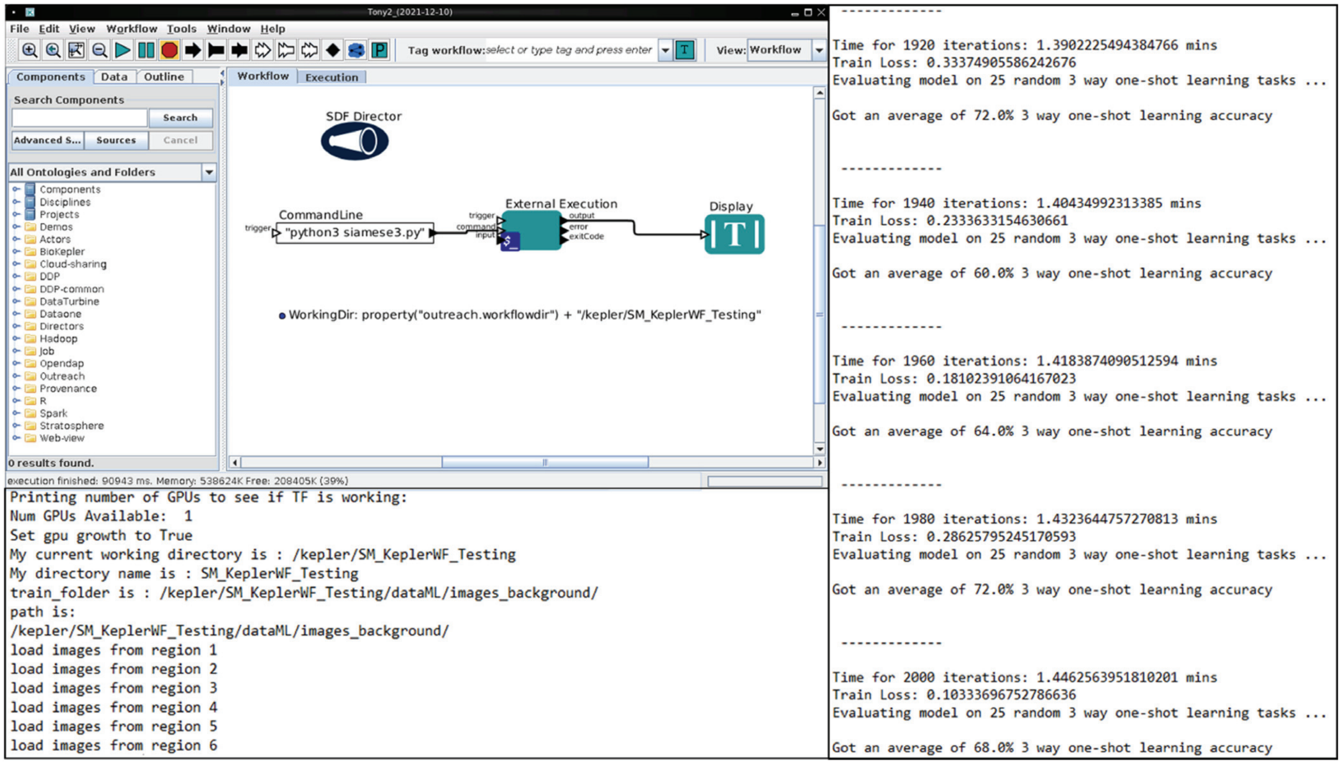

FIGURE 7 Kepler scientific workflow with Siamese neural network.

The multi-layer workflow-driven advanced manufacturing architecture is successfully mapped to the overarching SCW edge platform's off-site and on-site components in Figure 1. We have developed and deployed end-to-end predictive solutions in two distinct manufacturing applications that leverage the scientific workflow-driven manufacturing design framework. The first application predicts WiFi Received Signal Strength Indicator (RSSI) on a manufacturing floor dynamically in real-time.[30] The edge-intelligent Kepler workflow provides a real-time heat map of the WiFi signal distribution spatially across the manufacturing floor. This heat map empowers factory personnel to make data-driven decisions when installing new machines for collecting manufacturing process data to make automated decisions and can be utilized as a computational engine to provide geolocation for simulated human workflows. Behind the scenes, a lightweight Docker container encapsulates the Kepler workflows. This workflow leverages a trained CNN model to predict WiFi antenna location in real-time. The end-to-end training and execution of computationally expensive deep learning models on HPC is orchestrated by Kepler workflows. In the second application, the scientific workflow framework was leveraged to train and execute ML at the edge to classify surface roughness from speckle images generated by real time laser refraction. We built a generalizable scientific workflow which trains a Siamese Neural Network[31] to recognize different surfaces (categories). As shown in Figure 7 the Siamese Neural Network classifies an incoming image to regions by calculating probabilities. The workflow-driven framework enabled us to leverage this NN as a reduced order model in edge computing devices.

These two use cases demonstrated Kepler workflow's capability to orchestrate edge-aware computation and their adaptability to a diverse set of deep learning applications. The modular design makes the framework scalable for a wide range of manufacturing scenarios that aim to combine Computational Physics and ML to boost manufacturing effectiveness.

Readers who would like to find the success factors of the proposed methodology and the deployment case study are encouraged to read *Smart Connected Worker Edge Platform for Smart Manufacturing: Part 2: Implementation and On-site Deployment Case Study*.

8 | CONCLUSION

The system architecture of the proposed edge platform was developed from a methodically derived SRS. The architecture (and the SRS) could also be applicable to other edge to cloud-based platforms for smart manufacturing. For non-intrusive workflow monitoring and anomaly identification, a multi-agent model was implemented to capture the relationship between human workflows and energy consumption. Future directions include exploration of additional case studies of the multi-agent software model (with similar correlation-based ML agents) for novel manufacturing applications in the fields of energy efficiency, cybersecurity, and context awareness. In addition, the multi-agent features of the SCW architecture represent an ideal framework for developing real time intelligent systems that not only adaptively ingests new data features but are also capable of adaptively producing new programming by deploying symbolic AI in combination with workflow engines like Kepler scientific workflows. The SRS and system architecture were used to develop real-time machine state monitoring and fault detection for 3D printing that achieved excellent accuracy and speed in the real-time monitoring of the machine states. Future work in real-time machine state monitoring will require introducing other machine signals, such as energy consumption, to the machine state monitoring and fault detection framework. By analyzing the predicted machine states jointly with the energy consumption via time series data, the proposed framework may be applicable for optimizing energy usage between machines. In addition to improving operational performance the platform of human-machine interaction monitoring was developed to create a VR guidance system for the 3D printer users. Human-machine interaction is a complicated topic that involves far more than just text output and keystroke input. Therefore, future applications involve combining the proposed visual-based methodology with multimedia inputs, such as the machine's digital signal and energy signature, to create more robust and comprehensive monitoring methods. The SRS derived system architecture was also used to establish a scientific workflow platform that has been utilized to create a variety of computational systems that include: (1) prediction of real-time WiFi signal strength across the factory floor for collecting manufacturing process data and (2) classification of 3D printed surface roughness from laser refraction (i.e., speckle) images. In the future, we would like to extend the scientific workflow architecture to support online-learning which can enable end-users, such as operations managers, to update the underlying model's training parameters, daily or weekly, by training on the most recent manufacturing data collected by the edge devices. Further, we would like to support reinforcement learning capability where the underlying ML system continuously learns from manufacturing outcomes and self-adjusts its parameters to maximize the manufacturing performance.

AUTHOR CONTRIBUTIONS

Yoon G. Kim: Conceptualization (lead); data curation (equal); formal analysis (lead); methodology (equal); project administration (equal); software (supporting); supervision (lead); validation (equal); visualization (equal); writing – original draft (lead); writing – review and editing (lead). **Richard P. Donovan:** Conceptualization (lead); data curation (equal); funding acquisition (lead); project administration (lead); supervision (lead); validation (equal); writing – original draft (lead); writing – review and editing (lead). **Yutian Ren:** Conceptualization (equal); data curation (equal); formal analysis (lead); methodology (lead); software (lead); validation (lead); visualization (lead); writing – original draft (lead); writing – review and editing (supporting). **Shijie Bian:** Conceptualization (equal); data curation (equal); formal analysis (lead); methodology (lead); software (lead); validation (lead); visualization (lead); writing – original draft (lead); writing – review and editing (supporting). **Tongzi Wu:** Conceptualization (equal); data curation (equal); formal analysis (lead); methodology (lead); software (lead); validation (lead); visualization (lead); writing – original draft (lead); writing – review and editing (supporting). **Shweta Purawat:** Conceptualization (equal); data curation (equal); formal analysis (equal); methodology (lead); software (lead); validation (equal); visualization (equal); writing – original draft (lead); writing – review and editing (supporting). **Anthony J. Manzo:** Conceptualization (equal); data curation (equal); formal analysis (lead); methodology (lead); software (lead); validation (lead); visualization (lead); writing – original draft (lead); writing – review and editing (supporting). **Ilkay Altintas:** Conceptualization (equal); funding acquisition (lead); project administration (lead); supervision (lead). **Bingbing Li:** Conceptualization (lead); data curation (equal); funding acquisition (lead); project administration (lead); supervision (lead); writing – review and editing (equal). **Guann-Pyng Li:** Conceptualization (lead); data curation (lead); funding acquisition (lead); project administration (lead); supervision (lead); writing – review and editing (equal).

ACKNOWLEDGMENTS

The authors acknowledged the funding from the U.S Department of Energy, Office of Energy Efficiency and Renewable Energy, DE-EE0007613. This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Advanced Manufacturing Office Award Number DE-EE0007613.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

ORCID

Yoon G. Kim https://orcid.org/0000-0003-2844-6525
Shijie Bian https://orcid.org/0000-0003-2814-9854

REFERENCES

[1] C. Shannon, *The mathematical theory of communication*, University of Illinois Press, Urbana **1949**.
[2] P. Loucopoulos, in *Design process improvement: a review of current practice* (Eds: J. Clarkson, C. Eckert), Springer, London **2005**, p. 116.
[3] SG Network System Requirements Specification - Interim Release 3. https://www.energy.gov/gc/downloads/sg-network system-Requirements-specification-interim-release-3 **2010**. (accessed Jan 17 2022).
[4] B. Lu, V. C. Gungor, *IEEE Trans. Ind. Electron.* **2009**, *26*, 4651.
[5] L. Hou, N. W. Bergmann, *IEEE Trans. Instrum. Meas.* **2012**, *61*, 2787.
[6] J. Hu, F. L. Lewis, O. P. Gan, G. H. Phua, L. L. Aw, *IEEE Trans. Ind. Electron.* **2014**, *61*, 7083.
[7] C. Qian, Y. Zhang, C. Jiang, S. Pan, Y. Rong, *Robot. Comput. Integr. Manuf.* **2020**, *61*, 101841. https://doi.org/10.1016/j.rcim.2019.101841
[8] A. Vijayaraghavan, D. Dornfeld, *CIRP Ann.* **2010**, *59*, 21.
[9] D. Wu, S. Liu, L. Zhang, J. Terpenny, R. X. Gao, T. Kurfess, J. A. Guzzo, *J. Manuf. Syst.* **2017**, *43*, 25.
[10] M. Winikoff, L. Padgham, in *Artificial Societies, and Simulated Organizations* (Eds: F. Bergenti, M. P. Gleizes, F. Zambonelli) Springer, Boston, MA, USA **2004**. https://doi.org/10.1007/1-4020-8058-1_14
[11] J. Redmon, S. Divvala, R. Girshick, A. Farhadi. *Proc. of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Las Vegas, NV, USA **2016**, Vol. *2016*, pp. 779–788
[12] *Proc. of 2014 IEEE Conference on Computer Vision and Pattern Recognition*. (Eds: R. Girshick, J. Donahue, T. Darrell, J. Malik), IEEE, Columbus, OH, USA **2014**, pp. 580–587.
[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, A. Berg, *Lect. Notes Comput. Sci*, Columbus, OH, USA **2016**, *9905*, 21.
[14] X. Zhou, X. Xu, W. Liang, Z. Zeng, S. Shimizu, L. T. Yang, Q. Jin, *IEEE Trans. Ind. Inf.* **2022**, *18*, 1337. https://doi.org/10.1109/TII.2021.3061419
[15] P. Lasek, *Proc. Comput. Sci.* **2021**, *192*, 923.
[16] Visual Object Tagging Tool, https://github.com/microsoft/VoTT
[17] Y. Baek, B. Lee, D. Han, S. Yun, H. Lee. *Proc. of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Long Beach, CA, USA **2019**, Vol. *2019*, pp. 9357-9366
[18] F. Naiemi, V. Ghods, H. Khalesi, *Expert Syst. Appl.* **2021**, *170*, 114549.
[19] K. Wang, Y. Yi, Z. Tang, J. Peng, *Appl. Soft Comput.* **2021**, *108*, 107475.
[20] H. Karam, J. Tanaka, *Proc. Manuf.* **2015**, *3*, 5381.
[21] A. Caputo, A. Giachetti, F. Giannini, K. Lupinetti, M. Monti, M. Pegoraro, A. Ranieri, *Comput. Graph.* **2020**, *91*, 232.
[22] H. S. Kang, J. Y. Lee, S. Choi, H. Kim, J. H. Park, J. Y. Son, B. H. Kim, S. D. Noh, *Int. J. Precis. Eng. Manuf.-Green Tech.* **2016**, *3*, 111.
[23] B. Nedelcu, *Database Syst. J.* **2013**, *4*(3), 10.
[24] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, Y. Zhao, *Concurrency Computat.: Pract. Exper.* **2006**, *18*, 1039.
[25] I. Altintas, J. Wang, D. Crawl, W. Li, *Proc. of the 2012 Joint EDBT/ICDT Workshops (EDBT-ICDT '12)*, Association for Computing Machinery, New York **2012**, p. 73. https://doi.org/10.1145/2320765.2320791
[26] D. Merkel, *Linux J.* **2014**, *2014*. https://www.linuxjournal.com/content/docker-lightweight-linux-containers-consistent-development-and-deployment. (accessed Jan 17 2022).
[27] Y. Ren, G. P. Li, *IEEE Trans. Ind. Inf.* **2022**. https://doi.org/10.1109/TII.2022.3150795
[28] Y. Ren, G. P. Li, *J. Manuf. Syst.* **2022**, *62*, 87.
[29] S. Bian, T. Lin, C. Li, Y. Fu, M. Jiang, T. Wu, X. Hang, B. Li, in *Computational Science – ICCS 2021. ICCS 2021. Lecture Notes in Computer Science*, Vol. 12745 (Eds: M. Paszynski, D. Kranzlmüller, V. V. Krzhizhanovskaya, J. J. Dongarra, P. M. Sloot), Springer, Cham **2021**. https://doi.org/10.1007/978-3-030-77970-2_42
[30] E. Olivares, H. Ye, A. Herrero, B. A. Nia, Y. Ren, R. Donovan, F. Flaviis, *Neurocomputing* **2021**, *454*, 405.
[31] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. Lecun, C. Moore, E. Säckinger, R. Shah, *Int. J. Pattern Recognit. Artif. Intell.* **1993**, *07*(04), 669.

How to cite this article: Y. G. Kim, R. P. Donovan, Y. Ren, S. Bian, T. Wu, S. Purawat, A. J. Manzo, I. Altintas, B. Li, G.-P. Li, *J. Adv. Manuf. Process.* **2022**, *4*(4), e10129. https://doi.org/10.1002/amp2.10129