

Toward Cloud-Assisted Industrial IoT Platform for Large-Scale Continuous Condition Monitoring

This paper presents a new architecture for networks deployed in process control systems, which includes device and process characteristics. The architecture, which is based on an IIoT system, allows to perform fault detection and prediction.

By GANG WANG¹, MARK NIXON, AND MIKE BOUDREAU

ABSTRACT | Process industries cover a wide set of industries, in which the processes are controlled by a combination of distributed control systems (DCSs) and programmable logic controllers (PLCs). These control systems utilize various measurements such as pressure, flow, and temperature to determine the state of the process and then use field devices such as valves and other actuating devices to manipulate the process. Since there are many different types of field devices and since each device is calibrated to its specific installation, when monitoring devices, it is important to be able to transfer not only the device measurement and diagnostics but also characteristics about the device and the process in which it is installed. The current monitoring architecture, however, creates challenges for continuous monitoring and analysis of diagnostic data. In this paper, we present the design of an Industrial Internet-of-Things (IIoT) system for supporting large-scale and continuous device condition monitoring and analysis in process control systems. The system design seamlessly integrates existing infrastructure [e.g., highway addressable remote transducer (HART) and WirelessHART networks, and DeltaV DCS] and newly developed hardware/software components (e.g., one-way data diode and IIoT cellular architecture) together for control network data collection and

streaming of the collected device diagnostic parameters to a private cloud to perform streaming data analytics designed for fault identification and prediction. A prototype system has been developed and supported by Emerson Automation Solutions and deployed in the field for design validation and long-term performance evaluation. To the best of our knowledge, this is the first ever publicly reported effort on IIoT system design for process automation applications. The design can be readily extended for condition monitoring and analysis of many other industrial facilities and processes.

KEYWORDS | Continuous condition monitoring, industrial control, industrial IIoT, process monitoring.

I. INTRODUCTION

The potential for the industrial Internet includes many opportunities throughout the process industry that includes oil and gas production, chemical, specialty chemical, petrochemical, refining, pharmaceutical, food and beverage, power, cement, water and wastewater pulp and paper, and steel plants. For many of these industries, a shift in quality or productivity of 1%–2% can produce significant benefits through energy savings, reduction in waste, reduction in lost manufacturing time, improved safety, and reduced impact on the environment. To realize these potential benefits, the industry has adopted Industrial Internet-of-Things (IIoT) methods [1].

In most of the existing process control systems (PCSs), process measurements are periodically collected and communicated to gateways, controllers, and workstations or clouds [2]. As an example, the Oil and Gas industry

Manuscript received March 21, 2019; revised April 19, 2019; accepted April 24, 2019. Date of publication May 20, 2019; date of current version May 28, 2019. This work was supported by Emerson Automation Solutions. (Corresponding author: Gang Wang.)

The authors are with Emerson Automation Solutions, Round Rock, TX 78681 USA (e-mail: gang.wang@uconn.edu; mark.nixon@emerson.com; mike.boudreaux@emerson.com).

Digital Object Identifier 10.1109/JPROC.2019.2914021

collects data on flows, pressures, and temperatures of oil, gas, water, and other materials. The industry also collects data on the condition of machinery and equipment across the entire field installation. In the past, much of this data collection was performed manually [3]. The industry is now shifting toward the use of intelligent devices that provide this monitoring. These devices are capable of being networked together, allowing for a centralized location to collect and aggregate data [4]. These devices also include advanced diagnostics that can diagnose the health of the device and, in many cases, the health of the process to which the device is connected. It is not uncommon for the devices to include diagnostics that can detect plugged lines, burner flame instability, agitator loss, wet gas, orifice wear, leaks, and cavitations. These devices provide information on how well they are operating and when they need maintenance. Many of these installations utilize wireless highway addressable remote transducer (WirelessHART) [5].

Reliability is another area where the IIoT has made progress [6]. Reliability applications include well integrity monitoring, energy loss monitoring such as steam trap monitoring, pump health monitoring, and valve and equipment monitoring. In many cases, these data are being transferred to the cloud where the data can be stored and further processed. Cloud monitoring is driven by some unique features [7]. The cloud provides low-cost, highly scalable cloud-based storage, and processing capability; it offers innovative, lower cost deployments of sensor technology generally enabled by wireless capability and more out-of-the-box connectivity solutions to the cloud. In addition, cloud monitoring can connect experts to data wherever they are in a sustainable way.

The current monitoring architecture creates challenges for continuous monitoring and analysis of production and diagnostic data due to the lack of measurements, the limited network bandwidth, the labor-intensive and time-consuming data collection process, the lack of data trending and advanced analytics for accurate fault identification and prediction, the lack of details about the specific devices, and the lack of details about the process into which the instrument and actuator are installed.

The lack of measurements and connectivity is being addressed through standards such as HART protocol and WirelessHART [8], Open Platform Communications–Unified Architecture (OPC-UA) [9], and data-distribution service (DDS) [10]. In many cases, the connectivity is addressed by connecting edge gateways that connect devices already installed into the process to higher level applications that perform streaming and detailed diagnostics. In other cases, separate networks are being installed that are used to transfer the data directly to the cloud services.

IIoT consists of millions of devices designed and manufactured by hundreds of vendors. IIoT devices are not homogeneous with regard to the hardware platform [11]. The heterogeneous IIoT is a combination of

processors with different computing capabilities in different application scenarios and a few standardized communication mechanisms. The interoperability problems might happen between different devices [12]. Details about the devices themselves are well described through standards such as field device integration (FDI) [5], however, gaining access to these from outside of the asset monitoring systems has been problematic. In this paper, techniques will be described for extracting device description (DD) metadata from the devices and streaming it along with the data.

Historical data about the health of the device and the process in which the device is installed have been missing largely because there was no convenient way to capture these data and no low-cost way to store the data and metadata.

The last part of this relates to the process in which the devices are installed. It is important to be able to capture not only data about the device but also details about the process. For example, when monitoring the health of equipment such as heat exchangers, pumps, and valves, it is important to track the flows, temperatures, pressures, and other factors where the equipment is installed. In the long term, it is important to be able to plot the degradation of the equipment and to be able to evaluate how the equipment responds to load disturbances and changes in the overall process operation.

In this paper, we present the design of an IIoT system for supporting large-scale and continuous device condition monitoring and analysis in PCSs. One of the scientific contributions of this system design is that it seamlessly integrates existing infrastructure [e.g., HART and WirelessHART networks and DeltaV distributed control system (DeltaV DCS)] and newly developed hardware/software components (e.g., one-way data diode and IoT cellular architecture) together for control network data collection and streaming of the collected device diagnostic parameters to a private cloud to perform data analytics designed for fault identification and prediction. Also, the proposed prototype system has been validated and supported by Emerson Automation Solutions for long-term performance.

The remainder of this paper is organized as follows. Section II provides the background information. Section III provides a detailed system design, including architecture design, data-driven approach, and analytics in the cloud. Section IV describes the security. Section V walks through the proof of concept and initial field test results and concludes with information on moving forward. We conclude this paper and discuss future works in Section VII.

II. BACKGROUND

A. SOA of Industrial IoT

IIoT aims to connect different devices (or “things”) over the network. As a key technology in integrating heterogeneous systems or devices, service-oriented architecture (SOA) can be applied to support IIoT in

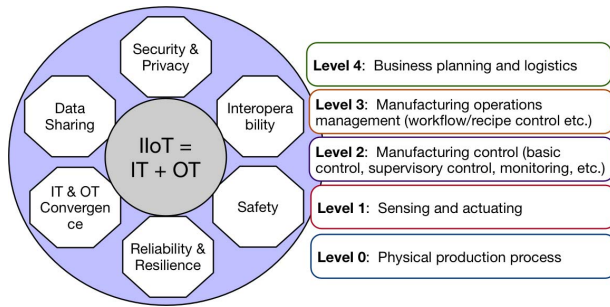


Fig. 1. Implementation of IIoT with fusion of OT and IT.

layered structures [13]. From the technology perspective, the design of an IIoT architecture needs to consider extensibility, scalability, modularity, and interoperability among heterogeneous devices [14].

At the computing domains in the industry, IIoT is commonly grouped into operational technology (OT) and information technology (IT), shaped by different requirements and environments [15]. At the core of IIoT are several shared challenges (as shown in Fig. 1), such as data sharing, interoperability, reliability, and resilience.

An SOA of IIoT includes the following four functional layers [16].

- 1) *Physical Layer (Level 0)*: This layer provides the physical production process, which is directly related to production hardware.
- 2) *Sensing Layer (Level 1)*: This layer is integrated with existing hardware (radio frequency identification (RFID), sensors, actuators, etc.) to sense/control the physical world and acquire data.
- 3) *Control Layer (Level 2)*: This layer provides basic networking support and data transfer over a wireless or wired network, as well as manufacturing control, such as basic control, supervisory control, and monitoring.
- 4) *Operational Layer (Level 3)*: This layer manages services that satisfy user needs. It provides the IIoT with a cost-efficient platform, where the hardware and software platforms can be reused.
- 5) *Interface Layer (Level 4)*: This layer provides interaction methods to users and other applications via business planning and logistics.

However, communication, synchronization with the physical production process, determinism, and real-time operations are challenges that must be resolved in IIoT scenarios. One key challenging task is to perform continuous condition monitoring (CCM) so that the control system can obtain real-time data and provide feedback for the actuating devices.

B. Continuous Condition Monitoring

The continuous and uninterrupted characteristics of the industrial process have historically challenged traditional maintenance practices. CCM has been one of

the main industrial challenges in the past decade [17]. A highly sophisticated methodology for condition monitoring has evolved in the past decade with respect to techniques, digital instruments, and computer chips [18]. Historically, preventive maintenance (PM) and predictive maintenance (PdM) [19] have required maintenance and inspection on a regular basis without prior knowledge of “normal” system conditions. Various sensors, such as vibration, thermal, current, voltage, and power, can be applied to achieve PdM via continuous monitoring [20]. However, these approaches are costly and most often require supervisory control via a DCS to record and analyze data.

The key idea of CCM is to monitor the health condition (or status) of the devices (including sensing and actuating devices) at each time step in terms of a continuous metric based on the available input data. CCM differs from traditional methods such as setting thresholds or differentiating distinct health states as various classes [21]. Instead, CCM monitors the health state of the devices in a continuous manner, which enables operational units to have smoother decision-making systems in the condition-based maintenance. The input data are typically a set of selected features that are extracted from nonintrusively sensed and captured signals [17]. These signals such as force, vibration, and acoustic emission can be captured and recorded using various sensors mounted on the machinery systems, and then, these signals are synchronized to the analytical system (e.g., cloud) to make the decision.

III. SYSTEM DESIGN

This section provides a detailed system design for cloud-assisted CCM in the large-scale IIoT. We first present the key components and these corresponding functionalities of each component and then introduce the data-driven approach for monitoring data processing.

A. System Overview

There are many industrial systems deployed today, which are connected to enterprise systems providing very significant operational benefits. These deployments extract data from a mixture of sensors, actuators, logic components, and databases (DBs) allowing them to interconnect and perform the functions requested by their individual users. The difference between these existing installations and what is described here is that with a more open IIoT approach, these industrial systems can connect in standardized ways that require very little or no configuration to support advanced data processing and cloud-based advanced historical and predictive analytics. The advanced cloud services can then be used to drive optimized decision-making and operational efficiencies and facilitate the collaboration between autonomous industrial control systems.

The best way to approach this overall architecture is through a three-tier architecture. These three tiers include

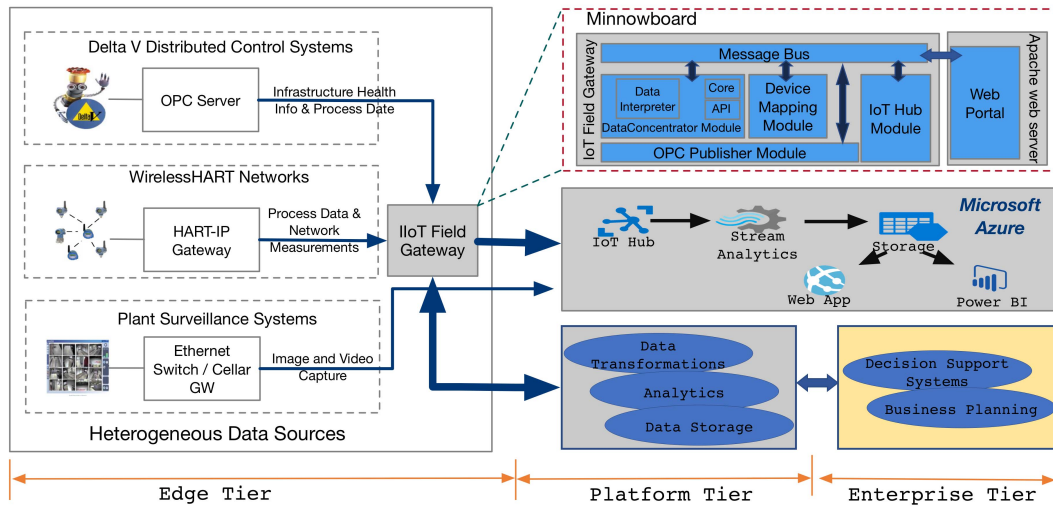


Fig. 2. Three-tier architecture, which consists of edge tier, platform tier, and enterprise tier. The edge tier via the IIoT gateway connects to the cloud and platform tier, and the enterprise tier accesses data from the platform tier.

an edge tier, a platform tier, and an enterprise tier. This architecture is illustrated in Fig. 2.

The edge tier is where data from the DCS data sources and nodes are collected, aggregated, and transmitted over the L2/L2.5 network to the L3 network. Some level of data translation and aggregation may also be performed at the edge gateway. The edge gateway may also include control applications or control extensions that are used to further process or aggregate data.

The platform tier receives data from the edge tier and is responsible for data storage, workflow processing, plant optimizations, and other applications. If a data diode is not being used, then the platform tier may also write data back through the edge gateway to the control system. Sites may require two edge gateway setups: a general one for one-way data flows and a highly restricted one for two-way communications.

The enterprise tier is where planning and decision support applications are utilized to perform streaming analytics, data mining, and reporting. This tier is also used to support data exploration, searches, and other functions.

The lower layers of the edge tier are comparable with the IEEE 1451 [22] in terms of keeping track of sensor information and product parameters. The IEEE 1451, a family of smart transducer interface standards, defines a set of open, common, network-independent communication interfaces for connecting transducers (sensors or actuators) to microprocessors, instrumentation systems, and control/field networks. The IEEE 1451 standard family provides a set of protocols for wired and wireless distributed monitoring and control applications [23]. The IEEE 1451.0 standard defines a common set of commands for accessing sensors and actuators connected in various physical configurations, such as point-to-point, distributed

multidrop, and wireless configurations to fulfill various application needs [24]. The major difference with the IEEE 1451 is that our system design seamlessly integrates existing infrastructure and newly developed components together to provide a complete data streaming approach to the cloud.

B. Architecture

Data were streamed from field devices such as valves through User Datagram Protocol (UDP) and Advanced Message Queuing Protocol (AMQP) [25] gateways to a cloud-based layer where data were analyzed, aggregated, and stored for further use and visualizations.

To prevent unintentional or unauthorized access to the control system, device data were streamed through a data diode. The AMQP protocol [26] was used to stream data from the field to the cloud via a secure IIoT field gateway (IIoT-FG). The proposed architecture consists of several important components: UDP and AMQP gateways, IIoT-FG, and the cloud.

1) UDP and AMQP Gateways: The UDP and AMQP gateways are used to transfer process measurements, instrument data, and diagnostic data from the field through to the cloud infrastructure. Physically, these gateways may be provided as two separate boxes or virtual machines separated by a data diode. They may also be packaged together as a single physical box.

Along with the ever-growing number of sensors and actuators being deployed in the field, a large number of real-time measurements are being streamed from heterogeneous resources for monitoring and control applications. A simple and unified streaming protocol is thus needed to define these data streams for emitting and retrieving data across platforms. Fig. 3 shows an abstract of streaming

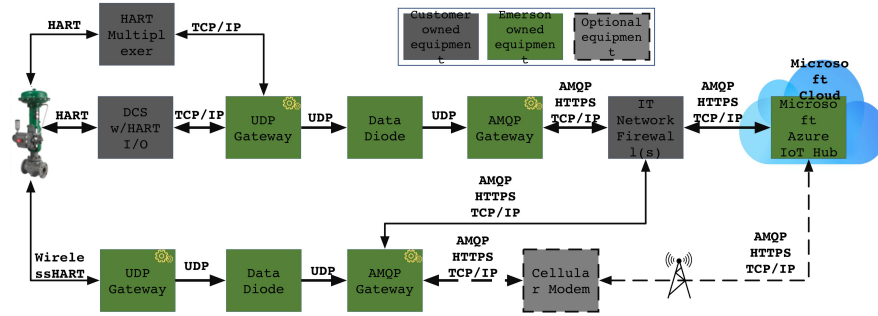


Fig. 3. Streaming data from the field through gateways to the cloud.

data from the field through gateways to the clouds. Given its capability to represent rich data structures in an extendable way, we use JavaScript Object Notation (JSON) [27] objects to define these data streams.

The content streamed through the gateways contains both data and metadata. The metadata itself can be extracted from the DDs and then through a discriminator to produce a data stream that contains values that are much more usable by the end applications. This procedure is described in Section III-C.

2) *Naming Conventions*: Naming conventions distinguish real-time data points from different plant resources. The format is consistent with the one used by DCS vendors. Each data point is assigned a unique tag name. Within one zone, the top name of the tag can be one of three types: module, workstation/controller, and devices. The data points are all defined as paths from the top name. In a plant with multiple zones, the zone name will be prefixed to the top name. To further distinguish different plants, the domain name will be prefixed to the zone name.

3) *Industrial IoT Field Gateway*: For IIoT scenarios in CCM, plenty of real-time data points from heterogeneous plant resources will be collected from a variety data connectors, via hardware devices and software interfaces, which are geographically distributed in the field. The connectors are usually running different communication protocols (e.g., OPC UA and HART-IP). Instead of implementing protocol adapters on each of the connectors to stream the data to the cloud, the system needs a universal IIoT-FG to connect to multiple data connectors to provide protocol adaptation and remote configuration. IIoT-FG has a small form factor, is cheap, and thus can support massive field deployment.

The left part of Fig. 2 shows the heterogeneous data sources. For instance, OPC UA servers are used to connect to DeltaV systems to retrieve periodic measurements from installed modules, controllers, and hardware devices. HART and WirelessHART gateways are used to connect sensor and actuator measurements as well as network health information in a real-time and continu-

ous manner. Wireless packet sniffers, spectrum analyzers, and surveillance cameras are installed in the plant to monitor operation and RF spectrum environments. All these real-time data will be streamed into cloud-based data analytics platforms for advanced modeling, scalable analytics, and real-time visualization and mobile alerting [28].

The top right part of Fig. 2 presents the software architecture of IIoT-FG. The current prototype has the following major software modules: 1) a web portal running on Apache to enable remote configuration; 2) a data concentrator module and an OPC publisher module to interpret HART-IP and OPC UA messages, respectively; 3) a device mapping module to map device UID to the device key used on the analytics platform; and 4) an IoT hub module to stream the data to the data analytics platform by supporting different messaging protocols, such as Hypertext Transfer Protocol (HTTP) [29], Message Queuing Telemetry Transport (MQTT) [30], and AMQP. All these modules communicate with each other through a message bus. Then, operators can remotely access the IIoT-FG via the web portal and configure the target data points and the associated streaming parameters. Process measurements will be streamed into IIoT-FG for protocol adaptation and then forwarded to the data analytics platform according to the messaging protocol.

4) *Cloud*: The cloud provides a centralized location to perform analytics, store data, provide access for users, and support capabilities such as notifications. For this

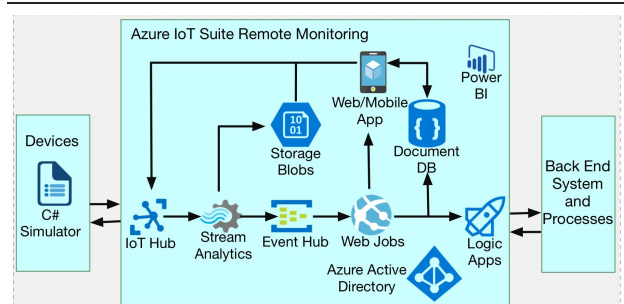


Fig. 4. Azure IoT suite for remote monitoring.

project, Microsoft Azure was used as the cloud platform. Features from the Azure IoT Suite were used to receive data, perform stream analytics and machine learning (ML), archive data, and display data. These features are shown in Fig. 4.

The IoT Hub receives the input data structure from the device network over the AMQP protocol. The IoT then divides the traffic and sends it to Stream Analytics and storage. The Stream Analytics application runs monitoring in-stream and sends the findings as they occur to the Event Hub. The Event Hub sends the events on to Web Jobs, storage for archiving, and to additional logic apps. Users can access the events as well as any archived data online through a web/mobile application. Document DB [31] is used by some of the applications for additional reporting and analytics.

The entire cloud environment is accessed through Azure Active Directory [32]. More complete information on the authorization and authentication is provided in the following sections.

C. Data-Driven Approach

A key design criterion is to reduce setup and configuration as much as possible. To make this happen, we adopt data-driven methods. The data-driven approach combines data from the devices together with DDs to fully describe the data to receiving applications. The data from devices are received as HART-published data. The data descriptions are extracted from the DDs (in the HART standard).

In data-driven approaches, data and functionality are decoupled. The approach can be effectively used in cases where the data are not overly complex. The approach can be applied as part of IIoT projects where thousands of device types hold multiple sets of measurement, diagnostic, and derived data and communicate it as structured data over communication networks to subscribing applications.

In the case of standards such as Foundation Fieldbus, HART, and WirelessHART, devices are described by a DD language [8] that describes devices in a consistent way. Each DD describes the device, methods provided by the device, measurement and device parameters that the device supports, configuration information, and the interactions that the users can perform with that device. The description file for a device is called the DD. A DD file provides a mapping of all parameters and functions of a device in a standardized language.

1) Message-Oriented Communication: The HART protocol supports message-oriented communications. With message-oriented communications, all communication between applications is based on messages that use well-known descriptions. In the case of HART, these descriptions are HART commands. With this communication pattern, it is not necessary for applications to know internal details about each other. Interaction between

applications is accomplished by passing HART commands over a common messaging medium. Several communication styles are supported, including request/response, publish/subscribe, event-based, and simple streaming.

Using messages and commands has several advantages. First, applications can be run in different environments (e.g., one side of the application may be running on a Windows-based host and the other end in an embedded device such as a pressure transmitter). Second, not all devices need to support all commands and services. Third, the device protocol can be easily abstracted from the application logic.

2) Communicating HART Data Using Data-Driven Methods: For this project, HART devices published their data using commands such as commands 1 and 9 through gateways to the Azure platform for condition monitoring and long-term storage. The data were converted from its highly compressed command structure into more easily understandable structures using a data discriminator. The data discriminator took raw packets and, using data descriptions for packet structures and commands and converted the raw data into JSON structures.

3) Messaging Protocol for Data Collection: Typically, a large amount of real-time measurements are required to be collected from heterogeneous plant resources for various monitoring and control applications. It is critical to define a simple and unified streaming protocol to transmit these data streams for cross-platform data emitting and retrieving. Given the capability to represent rich data structures in an extendable way, JSON objects are more suitable for these data streams.

Practically, JSON sends its schema along with every message, which requires a relatively large bandwidth. Since many compatible compression techniques have been reported to achieve good JSON format compression rates, they can be performed on data records to make the streaming protocol more bandwidth-efficient. All data records collected through the streaming protocol will be streamed to the cloud.

D. Analytics in Cloud

Two types of analytics were performed as part of the project. The first, stream analytics, was performed on data as it was streamed from the premises to the cloud services. The second, ML, involved building models offline and then deploying these models online as services. Both techniques are described in the following.

1) Stream Analytics: Azure Stream Analytics (ASA) start with a source of streaming data that is ingested into the Azure Event Hub (AEH), Azure IoT Hub, or from a data store like Azure Blob Storage [33]. To examine the streams, a Stream Analytics job is created that specifies the input source that streams data. The job also specifies a transformation query that defines how to look for data, patterns, and relationships. The transformation

query leverages a structured query language (SQL)-like query language that is used to filter, sort, aggregate, and join streaming data over time. When executing the job, the event ordering options and the duration of time windows when performing aggregation operations can be adjusted.

After analyzing the incoming data, an output for the transformed data is created. As part of setting up the output, actions can be set up to perform the following.

- 1) Send data to a monitored queue to trigger custom workflows downstream.
- 2) Send data to Power Business Intelligence (BI) dashboard for real-time visualization.
- 3) Archive data to other Azure storage services.

In the condition monitoring system created here, Stream Analytics jobs were created that monitored several features such as drive signal limits and travel limits. This is described in more detail in Section V.

2) *Machine Learning*: ML was used in this project to perform linear regression and time-series analysis using partial least-squares regression (PLS regression) [34]. Linear regression models were generated using libraries that existed as part of the Azure ML environment. PLS algorithms were developed in Python. Both linear regression and PLS trained models were executed online as web jobs. ML models and jobs were developed and executed as part of the Azure ML environment [33] [35].

Linear regression is a classic statistical technique used for regression problems to make a prediction for a continuous value from one or more variables or features. This algorithm uses a linear function and optimizes the coefficients that fit best to the training data. If you have only one variable, then you may think of this model as a straight line that best fits the data.

PLS regression is a statistical method that bears some relation to principal components regression; instead of finding hyperplanes of the maximum variance between the response and independent variables, it finds a linear regression model by projecting the predicted variables and the observable variables to a new space. Because both the X and Y data are projected to new spaces, the PLS family of methods are known as bilinear factor models.

PLS is used to find the fundamental relationships between two matrices (X and Y), i.e., a latent variable approach to modeling the covariance structures in these two spaces. A PLS model attempts to find the multidimensional direction in the X space that explains the maximum multidimensional variance direction in the Y space. PLS regression is particularly suitable when the matrix of predictors has more variables than observations and when there is multicollinearity among X values. By contrast, the standard regression will fail in these cases (unless it is regularized).

In the valve monitoring case, since there were only 9 features used to perform health indications, linear regression models worked well. In other examples that were tested that required over 100 features with significant coupling, PLS models worked better.

E. Continuous Condition Monitoring

IIoT drives many opportunities such as the potential for energy savings, improved quality, and increased throughput. While IIoT is new to many sectors of the economy, a form of it has been used since the 1960s in manufacturing. These early implementations did not use the Internet, which was decades away, but instead relied on plant and enterprise-wide intranets to deliver information from sensors to software and decision makers, where it drove operational improvements [36]. What is new is the IIoT service business model and corresponding solutions. These IIoT solutions start with sensors and deliver information to decision makers, but the infrastructure in between has changed drastically, offering three different and overlapping options for manufacturers. At one end of the service model is an in-plant intranets system that is used as the communications infrastructure to deliver sensor data to plant maintenance and engineering personnel. These raw data are then transformed into actionable information by various software applications licensed and operated by end users and augmented by their experienced domain experts. At the other end is a full outcome-based model where sensor data are delivered to a third-party service provider via the Internet. The service provider then analyzes the data with their own software applications and experts. They not only analyze the data but also send personnel to the manufacturing site to implement operational improvements by repairing or replacing malfunctioning components and equipment. The third model is a hybrid model, where the service provider analyzes sensor data and provides guidance to the plant on appropriate actions, with the plant taking final action. In all cases, the goal is to convert data to actionable information.

To illustrate how condition monitoring works with these models, consider what is in-place in an actual plant. All plants have flow, pressure, temperature, level, and other sensors that are connected to some type of automation system for controlling and monitoring the plant. Additional sensors can be added to support other types of applications such as equipment reliability, energy management, environmental monitoring, and so on. With the traditional model, intranet connections channel this sensor data to other parts of the organization for analysis by applications hosted in-house.

With this traditional model, sensors are added to large compressors at the plant, purely to monitor operating conditions. Data from these sensors are transmitted across the plant intranet and analyzed by in-house compressor experts. These experts can be onsite personnel looking at

just local compressors or remote personnel in a centralized corporate engineering center. The centers can monitor dozens of compressors located across multiple sites. Local plant service people are responsible for any corrective actions.

In the outcome-based method, these same compressor data go over the Internet to a third-party service provider such as Emerson, where experts use models and software to perform analysis using the latest tools. When problems are detected, the service provider takes corrective actions at the plant, providing uninterrupted operation of the compressors.

While the infrastructure described in this paper is ideal for a full outcome-based model, it may be used for the other two models as well. Tools such as analytic models are used to monitor equipment and provide recommended actions. These actions are stored in a DB for later recall. They are accessible through an Azure-hosted web site for users. In addition to being made available as a summary, the actions may also be communicated as notifications.

IV. SAFETY ANALYSIS

IIoT systems must follow a layered approach. With this approach, demilitarization zones (DMZs) separate internal LANs from external-facing servers that have access to the Internet and business networks. The most secure DMZ zones tend to be the control systems, followed by the site and business networks, followed by the servers that have access to the Internet. Firewalls are used to isolate networks. In some cases, it may be necessary to fully air gap the most restricted parts of the system, which means that the control systems are hidden behind a device such as a data diode. This isolation is shown in Fig. 2.

Another aspect of security is gateway authentication. In the condition monitoring described here, the connection between the on-premise gateways and the Azure cloud is formed bottom up, i.e., from the gateway to Azure. For this to work, the Azure side must be configured with a list of gateways that are allowed to connect. Once the gateway has been authenticated, the Azure Service Bus API hands it a token. The final step is for the connection to request a certificate and then use this certificate to encrypt communications.

All users who access the monitoring service must have valid credentials. User identity is validated through the Azure Active Directory.

The final aspect of security for this project was data protection. Two methods are used. In the first method, for each company or site, if required, separate Azure subscriptions are used for complete isolation. Within each Azure subscription, data are restricted using security filters.

V. PROOF OF CONCEPT

A. Overview

In this section, we present the objectives for the proof of concept (PoC) as follows.

- 1) PoC provides an “end-to-end” run of processing incoming telemetry from valves to hot and cold storage, including in-stream processing, as well as archives of all the data.
- 2) It generates alerts on the field data if the data fall outside operating ranges individualized for each valve or if the data exceed established limits.
- 3) PoC offers a way to train and connect to Azure ML models and access from Stream Analytics and custom algorithms.
- 4) It authenticates gateway connections using shared access signature (SAS) tokens.
- 5) It develops an analyst portal that allows experts and users to securely access field data, alerts, reports, and an online website.
- 6) PoC also refines the understanding of architecture and decision points for an IIoT solution.

The gateways are connected to WirelessHART networks. HART data were run through the data discriminator where it was converted to JSON form before being streamed over AMQP to the Azure environment. Once in Azure, the data were processed in an IoT Event Hub and forwarded as events to ASA and other Azure services. The data flow has several critical steps as follows.

- 1) In the field, data are published by WirelessHART devices. The parameters published, publish rates, scaling, and other settings are configured as part of the device configuration.
- 2) After generating the data, it is reformed into JSON form via the data discriminator. The data are then forwarded to the AMQP gateway where these are streamed to the Azure IoT Event Hub.
- 3) The IoT Event Hub converts the data stream into events (the data coming out of the gateways are streamed from the edge to Azure). These events are then passed on to ASA.
- 4) ASA processes each incoming event and performs some checks: 1) it generates alerts if the data fall outside normal operating regions for each valve; 2) it generates alerts if data exceed the derived or engineering limits; and 3) it performs evaluations using Azure ML models.
- 5) After stream analytics, Azure ML models are run for specific models, e.g., linear and PLS models, to generate learning results.
- 6) All events, which include both the raw data from the field devices as well as the stream and ML results, are passed through to archives for long-term storage. Several storage formats are used including blob storage and a Cassandra DB [37]. Cassandra is a distributed storage system for managing very large amounts of structured data spread out across many commodity servers while providing highly available service with no single point of failure.
- 7) An Azure website displays device data and alerts.

During the data transmission, the AMQP gateway performs as follows: 1) it accepts incoming UDP messages from the UDP gateway; 2) it acquires an SAS token from the Azure API app and then uses this token to communicate to AEHs using AMQP; and 3) it buffers up to 24-h worth data in case of loss of the Internet connection.

The core technologies supported as part of the Azure cloud are as follows.

- 1) *Azure Event Hub*: AEH supports communications between the on-premise gateways and Azure-based services.
- 2) *Azure Stream Analytics*: ASA monitors data in real-time as the data are received from devices, sensors, infrastructure, applications, and data. ASA provides out-of-the-box integration with AEH to ingest millions of events per second.
- 3) *Azure ML*: Azure ML performs more extensive analytics on the data. Once models are generated, they are exported as web services and called from ASA.
- 4) *Azure Storage*: A Cassandra column-store DB archives data and supports queries, reporting, and the online website.
- 5) *Azure Web App*: The Azure Web App presents data to analysts for use in valve monitoring and diagnostics.
- 6) *Azure API App*: Azure API Apps host REST-based APIs. The API App issues SAS tokens to on-premise gateways. The SAS tokens authenticate the on-premise gateway with the AEH.
- 7) *Azure Active Directory*: Azure Active Directory provides identity management and authentication for the analyst web portal.

For the analytics applications, several months of field data generated reference data for the valves. These reference data were then used as part of the stream analytics application. The field data were further used by the ML portion of the project through Azure ML.

B. Historical Field Data and Reference Data

Data collected from several sites were used to generate reference data and typical limits for the stream analytics and for the ML models developed as part of the Azure ML model generation. The reference data were generated offline and sent to Azure to be used in the stream analytics. The reference data included operating ranges for each valve derived from the data provided for the study. These data also included upper limits generated from the data aggregated across all of the valves used in the study. The reference data are originated from several manufacturing sites.

C. Stream Analytics

The stream analytics processed data en-route as it was received from each of the manufacturing sites. The processing was performed, as shown in Fig. 5.

Since this was a PoC, a couple of the limits was set lower than would be used once the condition monitoring site was

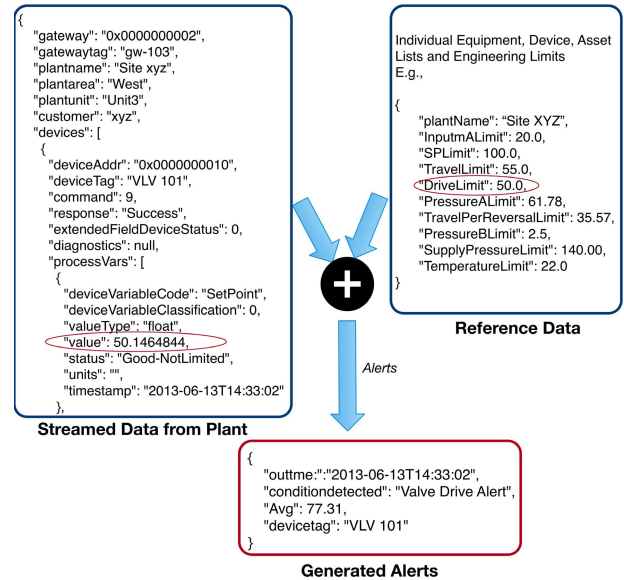


Fig. 5. Stream analytics processing.

set up for actual use. As shown in Fig. 5, using a drive signal level of 50% instead of the typical upper limit of 85% generated more alerts.

The stream analytics used a tumbling average for the input field value. Since it is possible to get a transient measurement that was outside the normal operating region, an average of the past 60-s worth data was used.

D. Machine Learning

There are many valve performance problems that require more extensive models than the simple models used as part of the stream analytics implementation. Examples include:

- 1) increase in valve friction;
- 2) process oscillation due to valve instability (requires additional device measurements, which we had but did not utilize);
- 3) oscillations due to loop tuning;
- 4) supply pressure issues;
- 5) high cycle counts;
- 6) sticking valve/high dead band;
- 7) changes in operating ranges;
- 8) throttling at low levels;
- 9) excessive travel.

For the PoC, linear and PLS models were generated to supply pressure issues and detect high cycle counts and excessive travel. The model generation is shown in Fig. 6.

The models were generated using the data provided offline and deployed online. They were initiated by the stream analytics processor.

E. Visualizing and Interacting With Site Data

After leaving the plant gateways, the data are analyzed instream, archived, and accessed by users for viewing and

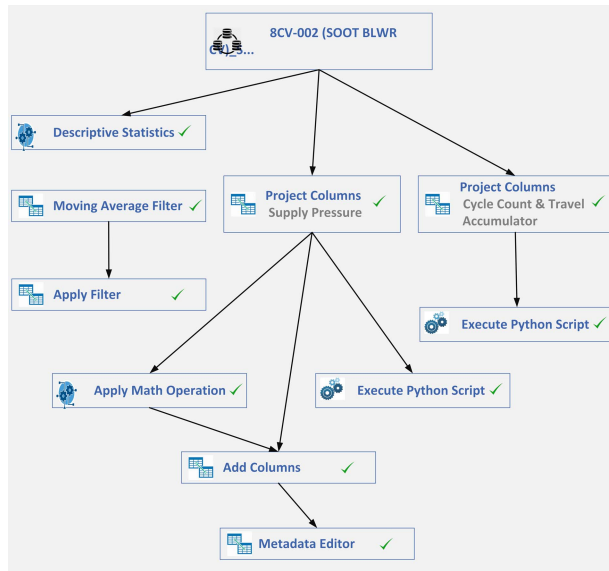


Fig. 6. ML for high cycle counts, excessive travel, and supply pressure issues.

analysis. For this project, the web interface shown in Fig. 7 was created.

The web interface enables the user to select the site and edge gateway to connect, view the top alerts, and browse the data. Since there can be more alerts that will fit on the alert banner, the total number is displayed in the active alert indicator. The right-hand side of the display shows the devices from which data are being collected and the parameters that were provided by the on-premise gateway. The naming, site, device tags, and parameters shown are provided by the on-premise gateway.

Categories were used to indicate different alert levels such as critical, warning, advisory, and log. Critical alerts are normally reserved for situations where action must be taken to avoid endangering the safety of plant equipment and/or personnel or impact on the environment. In some cases, critical alerts may be used to indicate that product quality will be impacted if action is not taken. Warning, advisory, and log alerts are used when the user has more time to respond, or when the outcome is not likely to lead to a condition that is not critical.

Alerts must be displayed with the highest unacknowledged alert at the top of the list. For this project, alerts were presented on an alert banner with the highest priority alert at the left. This is shown in Fig. 7.

VI. DISCUSSION

In this section, we provide some ongoing issues, which combine new technologies in the existing design. As the last mile of CCM, it needs to connect the actual plants to the operational units. The fifth generation (5G) of cellular wireless technology [38] promises to provide a communication platform to advance the communication between IT and OT. Similarly, abstracting metering in a secure way

is an ongoing concern. When multiple independent plants are involved in the large-scale IIoT, immutability and trustworthiness are important to guarantee data authenticity. Blockchain technology offers a mechanism to provide the immutability and trustworthiness among different plants.

A. 5G in IIoT

IIoT focuses on the integration between IT and OT [39] and on how smart objects (e.g., smart machines and networked sensors) improve services. IIoT generally implies machine-to-machine (M2M) interactions, either for application monitoring or as part of a self-organized system in a distributed manner [40]. However, many current cellular communication networks, such as the Third Generation Partnership Project (3GPP) do not support efficient machine-type communication (MTC). 5G communication provides several disruptive elements, such as increased data rate, reduced end-to-end latency, and improved coverage. In addition, by providing the integration of heterogeneous access, 5G can serve the role of unified interconnection framework, facilitating seamless connectivity of “things” with the Internet.

Currently, one of the most promising options for IIoT integration within the long-term evolution (LTE) [41]-enabled standard is narrowband IoT (NB-IoT) [42] [43]. NB-IoT is a new 3GPP cellular technology for providing wide-area coverage for IoT, which is designed to achieve excellent coexistence performance in the presence of legacy Global System for Mobile communications (GSM) [44] and LTE technologies. Still, further enhancements of NB-IoT are ongoing in 3GPP new releases. In general, NB-IoT is a step toward building the 5G radio access technology intended for enabling new use cases like efficient MTC. Due to the existence of NB-IoT devices while the network migrates toward 5G, it is important to design 5G access technology to coexist with NB-IoT and its evolution. It is also important to ensure that NB-IoT continues to evolve toward meeting all 5G requirements for IoT, minimizing any need to introduce a new 5G IoT technology.

The main objective is to increase the capacity of the current wireless technologies through self-organizing 5G technology and prepare for future scenarios [1]. In addition, end-to-end communication may require the integration of public cellular networking technologies with private networks, such as picocells or meshed networking topologies. However, the application functions should be applicable to different resources. They cannot rely on specific communication functions directly. Thus, generic communication services are required. Also, the services and data models in legacy systems should be consistent with the new networks so that there is some form of uniform data exchange at the higher networking layers, irrespective of the underlying communication systems.

With the help of end-to-end communication, 5G enables cellular networking technologies to be used in

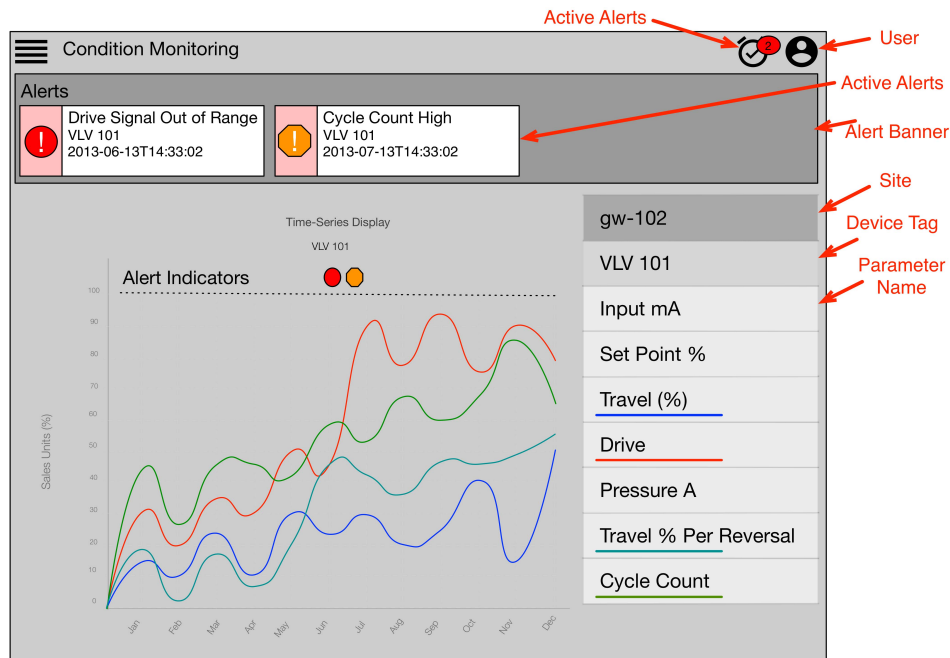


Fig. 7. Condition monitoring web interface.

industrial CCM without sacrificing the latency. Also, 5G's higher speeds enhance the feasibility of real-time condition monitoring. In addition, the IEEE standardization is moving toward industry standards (i.e., WirelessHART) [45].

B. Blockchain in IIoT

In IIoT, different devices will be interconnected together in a distributed manner. IIoT typically requires multiple entities, such as plants, to cooperate to perform some measurements or monitoring. It is particularly difficult to build secure and trusted distributed industrial systems that integrate information and computation from independent administrative plants. Each plant has policies for security and privacy but does not fully trust other plants to enforce them [46]. Integrating information from different plants is important because it enables new services and capabilities. The distributed ledger technology, e.g., Blockchain [47], provides a candidate to keep the consistency of blockchain.

A blockchain allows a number of participants in a restricted or unrestricted peer-to-peer (P2P) network to validate new transactions or blocks of new transactions and append them to the chain of previously validated blocks. The ledger is not owned or controlled by one centralized authority or company; it can be viewed by the participating nodes on the network. The transactions on the blockchain can be verified at any time in the future. Blockchain technology, as a distributed ledger, provides several innovative features, such as distribution, decentralization, trustworthiness, immutability, transparency, and

security [48]. These features together fit the multiple-plant scenario.

Combined with the current industrial infrastructure and communication networks, the blockchain data can be stored in the cloud to guarantee consistency and immutability.

VII. CONCLUSION

CCM is a critical and challenging process to monitor the health state of devices in a continuous manner for a large-scale IIoT platform. This paper presents a complete system design to stream the collected device diagnostic parameters, following the universal data-driven approach, to the cloud for analytics and decision-making. Specifically, the proposed system design seamlessly integrates existing infrastructure and newly developed components together for control network data collection. It then streams the collected device diagnostic parameters to a private cloud. We also provide a validated and deployed PoC to refine the understanding of the proposed prototype system for the IIoT solutions.

Currently, this prototype system has been developed and supported by Emerson Automation Solutions and deployed in the field for design validation and long-term performance evaluation. Future work is focused on three areas. The first is to deploy this prototype for other CCM use cases and evaluate latency, throughput, and path stability. The second area is expanding the scope beyond devices into process equipment, process units, and the process itself. This includes sending the structure of the data, the relationships between equipment, and the data itself.

The third area is to incorporate new technologies (e.g., 5G and blockchain technology) into the current prototype system to provide more exciting features, such as low latency, immutability, and trustworthiness. ■

Acknowledgments

The authors would like to thank the people at Microsoft, Rosemount, and Fisher, who participated in the project.

REFERENCES

- [1] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the Internet of Things and industry 4.0," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 17–27, Mar. 2017.
- [2] Y. Liu, R. Candell, K. Lee, and N. Moayeri, "A simulation framework for industrial wireless networks and process control systems," in *Proc. IEEE World Conf. Factory Commun. Syst. (WFCS)*, May 2016, pp. 1–11.
- [3] P. L. Ingrassia et al., "Data collection in a live mass casualty incident simulation: Automated RFID technology versus manually recorded system," *Eur. J. Emergency Med.*, vol. 19, no. 1, pp. 35–39, 2012.
- [4] P. Priller, A. Aldrian, and T. Ebner, "Case study: From legacy to connectivity migrating industrial devices into the world of smart services," in *Proc. IEEE Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2014, pp. 1–8.
- [5] D. Chen, M. Nixon, and A. Mok, "Why wirelessHART," in *WirelessHART*. Boston, MA, USA: Springer, 2010, pp. 195–199.
- [6] Z. Sheng, C. Mahapatra, C. Zhu, and V. C. M. Leung, "Recent advances in industrial wireless sensor networks toward efficient management in IIoT," *IEEE Access*, vol. 3, pp. 622–637, Jun. 2015.
- [7] A. Colombo et al., *Industrial Cloud-Based Cyber-Physical Systems: The IMC-AESOP Approach*, 2014.
- [8] J. Song et al., "WirelessHART: Applying wireless technology in real-time industrial process control," in *Proc. IEEE Real-Time Embedded Technol. Appl. Symp.*, Apr. 2008, pp. 377–386.
- [9] S. Lehnhoff, S. Rohjans, M. Usler, and W. Mahnke, "OPC unified architecture: A service-oriented architecture for smart grids," in *Proc. 1st Int. Workshop Softw. Eng. Challenges Smart Grid*, Jun. 2012, pp. 1–7.
- [10] G. Pardo-Castellote, "OMG data-distribution service (DDS): Architectural overview," Real-Time Innovations, Sunnyvale CA, USA, Tech. Rep., 2004.
- [11] L. Marin, M. P. Pawlowski, and A. Jara, "Optimized ECC implementation for secure communication between heterogeneous IoT devices," *Sensors*, vol. 15, no. 9, pp. 21478–21499, 2015.
- [12] G. Xiao, J. Guo, L. D. Xu, and Z. Gong, "User interoperability with heterogeneous IoT devices through transformation," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1486–1496, May 2014.
- [13] W. Zhiliang, Y. Yi, W. Lu, and W. Wei, "A SOA based IIoT communication middleware," in *Proc. Int. Conf. Mech. Sci., Electr. Eng. Comput. (MEC)*, Aug. 2011, pp. 2555–2558.
- [14] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IIoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.
- [15] H. Derhamy, "Architectural design principles for industrial Internet of Things," Ph.D. dissertation, Luleå Univ. Technol., Luleå, Sweden, 2018.
- [16] L. Da Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.
- [17] O. Geramifard, J.-X. Xu, J.-H. Zhou, and X. Li, "A physically segmented hidden Markov model approach for continuous tool condition monitoring: Diagnostics and prognostics," *IEEE Trans. Ind. Informat.*, vol. 8, no. 4, pp. 964–973, Nov. 2012.
- [18] R. B. Randall, *Vibration-Based Condition Monitoring: Industrial, Aerospace and Automotive Applications*. Hoboken, NJ, USA: Wiley, 2011.
- [19] Z. Li, K. Wang, and Y. He, "Industry 4.0—Potentials for predictive maintenance," in *Proc. 6th Int. Workshop Adv. Manuf. Autom.*, Paris, France: Atlantis Press, 2016, pp. 42–46.
- [20] B. Lu, D. B. Durocher, and P. Stemper, "Online and nonintrusive continuous motor energy and condition monitoring in process industries," in *Proc. Rec. 54th Annu. Pulp Paper Ind. Tech. Conf.*, Jun. 2008, pp. 18–26.
- [21] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6418–6428, Nov. 2014.
- [22] K. Lee, "IEEE 1451: A standard in support of smart transducer networking," in *Proc. 17th IEEE Instrum. Meas. Technol. Conf.*, vol. 2, May 2000, pp. 525–528.
- [23] *IEEE Standard for a Smart Transducer Interface for Sensors and Actuator—Wireless Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats*, IEEE Standard 1451.5-2007, 2007.
- [24] E. Y. Song and K. Lee, "Understanding IEEE 1451-networked smart transducer interface standard—What is a smart transducer?" *IEEE Instrum. Meas. Mag.*, vol. 11, no. 2, pp. 11–17, Apr. 2008.
- [25] S. Vinoski, "Advanced message queuing protocol," *IEEE Internet Comput.*, vol. 10, no. 6, pp. 87–89, Nov. 2006.
- [26] R. Godfrey, D. Ingham, and R. Schloming, "OASIS advanced message queuing protocol (AMQP) version 1.0; OASIS standard," Tech. Rep.
- [27] D. Crockford, *The Application/JSON Media Type for Javascript Object Notation (JSON)*, document RFC 4627, 2006.
- [28] S. Han, T. Gong, M. Nixon, E. Rotvold, K.-Y. Lam, and K. Ramamritham, "RT-DAP: A real-time data analytics platform for large-scale industrial process monitoring and control," in *Proc. IEEE Int. Conf. Ind. Internet (ICII)*, Oct. 2018, pp. 59–68.
- [29] R. Fielding et al., *Hypertext Transfer Protocol—HTTP/1.1*, document RFC 2616, 1999.
- [30] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-S—A publish/subscribe protocol for wireless sensor networks," in *Proc. 3rd Int. Conf. Commun. Syst. Softw. Middleware Workshops (COMSWARE)*, May 2008, pp. 791–798.
- [31] J. Han, E. Haihong, G. Le, and J. Du, "Survey on NOSQL database," in *Proc. 6th Int. Conf. Pervasive Comput. Appl.*, Oct. 2011, pp. 363–366.
- [32] M. Copeland, J. Soh, A. Puca, M. Manning, and D. Gollob, "Microsoft azure and cloud computing," in *Microsoft Azure*. Springer, 2015, pp. 3–26.
- [33] B. Calder et al., "Windows azure storage: A highly available cloud storage service with strong consistency," in *Proc. 23rd ACM Symp. Oper. Syst. Princ.*, 2011, pp. 143–157.
- [34] P. Geladi and B. R. Kowalski, "Partial least-squares regression: A tutorial," *Anal. Chim. Acta*, vol. 185, pp. 1–17, 1986.
- [35] S. Mund, *Microsoft Azure Machine Learning*. Birmingham, U.K.: Packt, 2015.
- [36] P. Zornio, "New application business models—The real IIoT difference," *CIO Rev.*, 2018.
- [37] A. Lakshman and P. Malik, "Cassandra: A decentralized structured storage system," *ACM SIGOPS Oper. Syst. Rev.*, vol. 44, no. 2, pp. 35–40, Apr. 2010.
- [38] J. G. Andrews et al., "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.
- [39] M. R. Palattella, P. Thubert, X. Vilajosana, T. Watteyne, Q. Wang, and T. Engel, "6TISCH wireless industrial networks: Determinism meets IPv6," in *Internet of Things*. Cham, Switzerland: Springer, 2014, pp. 111–141.
- [40] M. R. Palattella et al., "Internet of Things in the 5G era: Enablers, architecture, and business models," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 510–527, Mar. 2016.
- [41] E. Dahlman, S. Parkvall, and J. Sköld, *4G: LTE/LTE-Advanced for Mobile Broadband*. New York, NY, USA: Academic, 2013.
- [42] Y.-P. E. Wang et al., "A primer on 3GPP narrowband Internet of Things," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 117–123, Mar. 2017.
- [43] R. Ratasuk, B. Vejlgaard, N. Mangalvedhe, and A. Ghosh, "Nb-IoT system for M2M communication," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2016, pp. 1–5.
- [44] P. Struckmann, *The GSM Evolution: Mobile Packet Data Services*. Hoboken, NJ, USA: Wiley, 2003.
- [45] H. Kurunathan, R. Severino, A. Koubaa, and E. Tovar, "IEEE 802.15.4e in a nutshell: Survey and performance evaluation," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1989–2010, 3rd Quart., 2018.
- [46] G. Wang, Z. J. Shi, M. Nixon, and S. Han, "SMChain: A scalable blockchain protocol for secure metering systems in distributed industrial plants," in *Proc. Int. Conf. Internet Things Design Implement.*, 2019, pp. 249–254.
- [47] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaram, "Blockchain technology: Beyond bitcoin," *Appl. Innov.*, vol. 2, nos. 6–10, p. 71, 2016.
- [48] M. Pilkington, "Blockchain technology: Principles and applications," in *Research Handbook on Digital Transformations*, vol. 225, 2016.

ABOUT THE AUTHORS

Gang Wang received the B.S. degree in software engineering from Qingdao University, Qingdao, China, in 2010, and the two M.S. degrees in computer science from the Institute of Computing Technology, Chinese Academy of Sciences (ICT, CAS), Beijing, China, in 2013, and Temple University, Philadelphia, PA, USA, in 2014, respectively. He is currently working toward the Ph.D.



degree at the Department of Computer Science and Engineering, University of Connecticut, Storrs, CT, USA.

He is currently at Emerson Automation Solutions, Round Rock, TX, USA. His current research interests include blockchain, computer security, and industrial Internet of Things.

Mark Nixon received the bachelor's degree from the University of Waterloo, Waterloo, ON, Canada, in 1982.

He was a System Engineer in Cambridge, ON, Canada, in projects of oil and gas, chemicals, and pulp and paper. He moved from Cambridge, ON, Canada, to Austin, TX, USA, in 1988, where he held various positions in both research and development. From 1995 to 2005, he was involved in the inception of DeltaV and was a Lead Architect in Austin, TX, USA. In 2006, he took a very active role in the design and standardization of WirelessHART. He is involved in several standards groups including ISA, IEC, FieldComm, IEEE, and OASIS. He currently leads the Applied Research Group, Austin, TX, USA, where he is involved in control, analytics, wireless, mobile, and advanced graphics. He has coauthored five books on wireless, control, and operator interfaces. He holds more than 150 patents.



Mr. Nixon is an ISA Fellow and was inducted into the Automation Hall of Fame in 2012. He is one of the editors of ISA101.

Mike Boudreaux received the B.S. degree in chemical engineering from the University of Houston, Houston, TX, USA, and the M.B.A. degree from the Kellogg School of Management, Northwestern University, Evanston, IL, USA.

He held various roles in engineering, product management, marketing, and sales at Emerson, Round Rock, TX, USA, Alcoa, Rockdale, TX, USA, and AkzoNobel, Deer Park, TX, USA, and Chicago, IL, USA. He is currently the Director of Connected Services with Emerson Automation Solutions, Round Rock, TX, USA.

