Developing Real-Time Smart Industrial Analytics for Industry 4.0 Applications

Pankesh Patel and Muhammad Intizar Ali

Abstract Industry 4.0 refers to the 4th Industrial Revolution—the recent trend of automation and data exchange in manufacturing technologies. Traditionally, Manufacturing Executing System (MES) collects data, and it is only used for periodic reports giving insight about past events. It does not incorporate real-time data for up-to-date reports. Production targets are mostly predefined, before the actual production starts. The different production anomalies are known to happen in the real world, affecting the predefined production targets. Moreover, a key challenge faced by industry is to integrate multiple autonomous processes, machines, and businesses.

A broad objective of our work is to build an integrated view that can make data available in a unified model to support different stakeholders of a factory (e.g., factory planners, managers) in decision-making. In this chapter, we focus on designing an approach for building Industry 4.0 smart services and addressing real-time data analytics, which can integrate multiple sources of information and analyze them on the fly. Moreover, we share our experience of applying IoT and data analytics approach to a traditional manufacturing domain, thus enabling smart services for Industry 4.0. We also present our key findings and challenges faced while deploying our solution in real industrial settings. The selected use case studies demonstrate the use of our approach for building smart Industry 4.0 applications.

P. Patel (✉)
Data Science Institute, NUI Galway, Galway, Ireland
e-mail: pankesh.patel@insight-centre.org

M. I. Ali
Data Science Institute, NUI Galway, Galway, Ireland

Dublin City University, Dublin, Ireland
e-mail: ali.intizar@insight-centre.org; ali.intizar@dcu.ie

M. Maleshkova et al. (eds.), *Smart Service Management*,
https://doi.org/10.1007/978-3-030-58182-4_17

1 Introduction

Industry 4.0 or smart manufacturing brings the 4th Industrial Revolution by integrating all digitized services and facilitating automation in manufacturing. The aim of Industry 4.0 is to provide end-to-end connected smart solutions. Cyber Physical Systems are a key pillar of Industry 4.0 as they provide interconnected services between physical assets and their computational spaces (Lee et al. 2015). Industrial data analytics is another important pillar of the Industry 4.0, as it supports intelligent, automated decision-making. With the recent scientific achievements in machine learning and deep learning technologies, it is now possible to analyze a large amount of data and provide actionable insights.

The key goal of this chapter is to demonstrate on how to build smart services for Industry 4.0 domain that can use data analytics to make intelligent decisions (Lee et al. 2014). Among the key challenges to build the smart services for Industry 4.0, the fundamental challenge is *lack of real-time analysis*. The traditional approach in Industry 4.0 is to compile historical data and generate reports for decision-making (Berson and Smith 1997). A common pattern found is that data are stored in the databases. Then, the stored data are retrieved later to generate periodic reports, analyzing the insights about past events. This pattern is not able to incorporate real-time data (e.g., device real-time alarms). An analysis in real time can be a key to accurately predict at more granular time intervals. Therefore, it is necessary to set up an Industry 4.0 data collection infrastructure that can provide end-to-end transparency in real time (e.g., the status of the production in the manufacturing process), allowing for optimization not only across the factory sites but also in the entire supply chain. Moreover, the blend of historical data and contextual data generated by IoT devices can improve the outcomes of decision-making algorithms (Watson and Wixom 2007). A few middleware solutions have been proposed (Gao et al. 2017; Intizar et al. 2017) for real-time analytics. However, the applications of the proposed approaches are missing in the manufacturing domain (Zhong et al. 2017). Another challenge is *lack of interoperability*. The data collection largely is not interconnected. This results into silos of data, making the interoperability of data very difficult. The complex and heterogeneous nature of the equipment used in the manufacturing industry sometimes makes it difficult to get an overall perspective. As the technology advances, the new machines are often delivered with powerful technologies. For SMEs (small-to-medium-sized enterprises) with older machines installed at its factory, it can be challenging to catch up with the complex IT standards that come along. The equipment used in factories is often based on proprietary software that uses proprietary protocols, and it is often difficult to update to more modern protocols. This environment makes it challenging to create solutions that monitor equipment across entire factory floors and across different factories.

By achieving interoperability, it is possible to build Industry 4.0 smart services, in which multiple autonomous systems can be capable of exchanging information on the fly and make automated intelligent decision after analyzing the collected

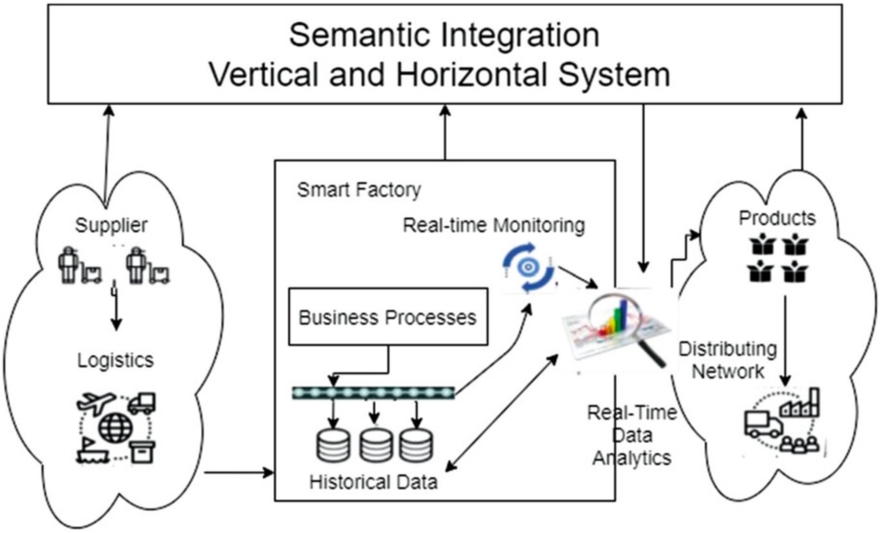


Fig. 1 A semantic-enabled platform for Industry 4.0 systems

information. Currently, the business intelligence is mostly limited to a department level or to a site level at the most. We envision an ecosystem of Industry 4.0 applications, where multiple autonomous systems can share information in real time and collectively make decisions for the common good (see Fig. 1). Imagine a scenario of supply chain management, where multiple stakeholders are involved. An integrated middleware should enable the integration of the systems supported by multiple stakeholders and optimize manufacturing tasks accordingly. In such scenarios, a delay in the supply chain must already alert the manager of the shop floor to optimize manufacturing processes accordingly or a weather calamity event must automatically trigger actions expecting abruption in manufacturing processes and consequently a reduction in daily production goals.

An integrated and holistic view of a factory can be established to improve the decision-making and to reduce the overall complexity. The interlinking includes the interlinking of diverse data sources such as anomalies in real time (e.g., machine breakage), the manufacturing execution system (e.g., production data), business processes, and so on. Although much of these data are already captured by IT system, it largely remains inaccessible in an integrated way without investigating manual effort. Thus, *the broad objective of our research is to build an integrated view that can make data available in a unified model to support different stakeholders of a factory (e.g., factory planners, managers) in decision-making.* Section 3 presents AI- and semantic-based conceptual framework (named SWeTI Patel et al. 2018) to achieve this broad objective.

In this chapter, we focus on designing an approach for building Industry 4.0 smart services and addressing real-time data analytics, which can integrate multiple sources of information and analyze them on the fly. Moreover, we share our experience of applying IoT and data analytics approach to a traditional manufacturing domain, thus enabling smart services for Industry 4.0. Using our open-source and standards-based approach, autonomous systems could be seamlessly integrated using semantic technologies. The proposed approach can analyze large amount of historical manufacturing data by applying machine learning algorithms and collecting and analyzing sensor data on the fly. It facilitates an integrated view of information from historical as well as real-time data perspective and facilitates intelligent decision-making.

We also discuss a real-world production manufacturing use case, provided by a large manufacturer of bio-medical devices (more details in Sect. 2). We elaborate our approach to design a real-time data analytic solution based on production forecasting. The proposed approach uses historical data of production processes to train ML algorithms for future production goals, which help the manufactures to set optimal and realistic goals for production. Contrary to traditional machine learning approach that only considers historical data pattern, the proposed approach supports a real-time monitoring to detect abnormal events (such as machine breakages, head count shortages, and unavailability of raw materials). The impact of these abnormal events is calculated and used to adjust the hourly, daily, and weekly production targets accordingly. Our proposed approach integrates real-time monitoring techniques to trigger notifications for taking the remedial actions in real time.

Outline The remainder of this chapter is structured as follows: In Sect. 2, we take one real-world Industry 4.0 case study. Section 3 presents our AI- and semantic-based conceptual framework (named SWeTI Patel et al. 2018) for building smart services for Industry 4.0. Background and existing approaches to build smart services for Industry 4.0 are discussed in Sect. 4. We discussed our approach to address the objective of the case study in Sect. 5, before concluding in Sect. 6.

2 Motivating Use Case: Smart Industrial Analytics

This section presents a production forecasting use case in the Industry 4.0 domain. We consider a production forecasting of a large medical device manufacturer, which is one of our industrial partners at the CONFIRM SFI Research Centre for Smart Manufacturing (https://confirm.ie/). Our industry partner manufactures orthopedic devices such as knee, hip, and shoulder joint replacements. The organization has multiple manufacturing units at various geographical locations, across Ireland and worldwide.

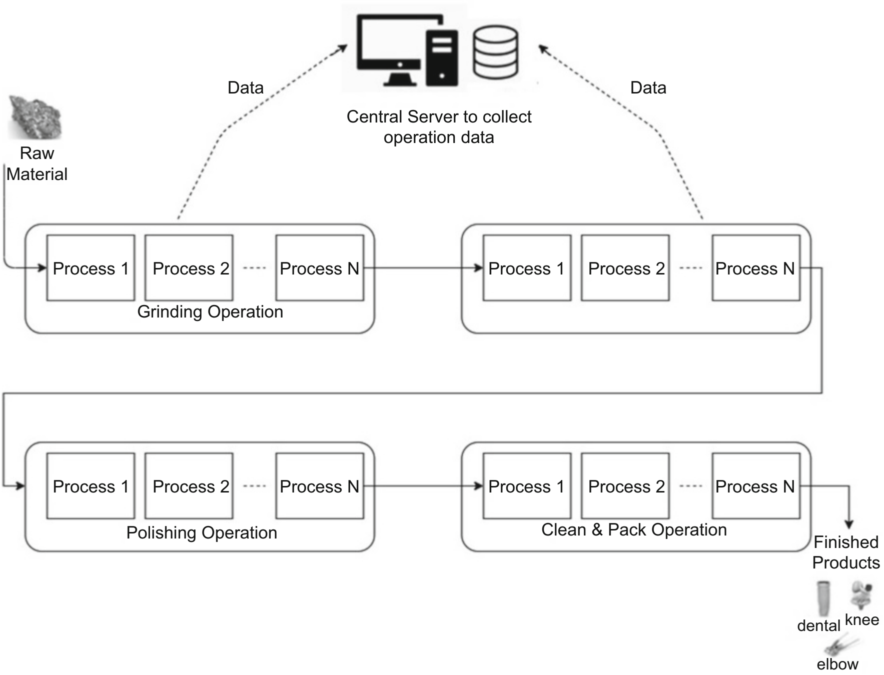


Fig. 2 Production process layout, from raw material to a finished product

Figure 2 presents a production process layout. At a manufacturing unit, a typical line of production at the shop floor is sequential. For simplicity reasons, we present broad steps of the manufacturing process, however, the actual lines of production are usually very complex. Figure 2 presents the production process layout steps from raw material to grinding, from grinding to polishing, and from polishing to cleaning and packing. A machine is responsible for executing one or more steps of an operation (e.g., grinding). Each machine has specific characteristics that restrict a set of products, which can be allocated to it. The manufacturing process is carried out in a batch processing manner. Each machine can only run one batch at a time.

Due to the sequential production process at the manufacturing unit, any kind of anomaly at any stage leads to domino effect on subsequence manufacturing processes. The organization uses an internal Manufacturing Execution System (MES) to keep track of its daily processes and stores relevant information about each processing step of each manufacturing process. The collected data are used to generate periodic reports, summarizing the actual production between requested time frames. The generated reports are used by factory planners to set future production targets.

The current system at the company is facing some challenges, which needs to be addressed to achieve overall goals of the manufacturing company. These goals are to reach production goal on time to meet the product demand, to reduce manufacturing cost, and to maximize utilization of resources. In the following, we present the challenges that need to be addressed to achieve these objectives:

Real-Time Visibility The existing system collects data, and it is only used for the generation of periodic reports giving insights about past events. It does not incorporate real-time data, events for up-to-date reports, and feedback from the supervisor(s) of shop floor. To address these limitations, the company needs a system that allows them to capture data in real time from each process and shows the production targets in real time. If the threshold condition is not met, then deviations are recorded and supervisors are notified. So, the supervisors can take appropriate actions to minimize the effect. Moreover, the deviation reasons can be recorded for future analysis for improvements and production planning. Sections 4.1 and 4.2 present the state-of-the-art tools to address this challenge.

Anomalies at Runtime The production targets are set, before the actual production starts. More specifically, the planners largely define production goals based on the plant's current capacity of producing the number of units, supply and demand consideration, and consideration of past events or situations. The different anomalies are known to happen in the real world. For instance, the anomalies such as machine breakage, raw material low supply due to some external events (e.g., logistics delay, supplier or distributor issues), manpower shortage, quality issues such as scrap and rework. These are not considered during the goal settings, thus affecting the overall production targets. To address this limitation, a company would need a set of tools (state of the art presented in Sect. 4.2) to monitor, detect, and report events. To detect an event, different thresholds based on historical data analysis and domain knowledge from staff members of the organizations are implemented.

Interoperability Data collected at each process (e.g., grinding, polishing, cleaning, and packaging) are not interconnected and interoperable, resulting into silos of information for each process. This largely occurs because the company is using different systems, supplied by different vendors, each has its own data collection software, different communication protocols as well as different data format and files. To ensure accurate prediction, the company would need to integrate data from all relevant processes. Semantic Web approaches discussed in Sect. 4.3 can play a role to achieve this objective.

Self-Configuration Due to the advancement in technologies, the manufacturers may be interested in self-adaptive approaches, which can automatically adjust goals and targets based on current processes. An ideal scenario is to develop a system that can automatically reduce daily production targets according to unexpected events such as machine failure (Wang et al. 2016). This approach would ensure a maximum utilization of available resources.

To address the objectives, such as mentioned above, the next section presents our AI- and semantic-based conceptual framework (named SWeTI Patel et al. 2018) for building smart services for Industry 4.0.

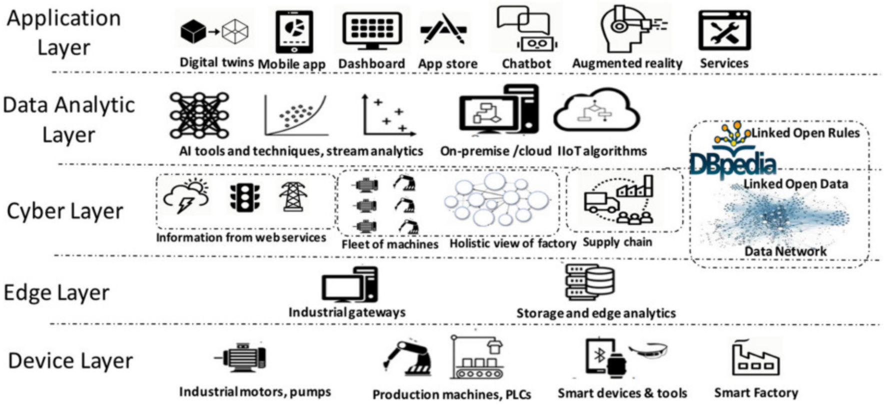


Fig. 3 A layered view of SWeTI platform

3 SWeTI: A Semantic Web of Things Platform for Building Industry 4.0 Smart Services

This section presents a layered architecture of SWeTI platform briefly (Patel et al. 2018). Figure 3 presents an architecture. It begins with the data processing pipeline at the machine level and moves toward intelligent autonomous applications.

Device Layer The shop floor at a factory hosts various industrial devices (e.g., pumps, motors, PLCs, industrial robots) and smart devices (e.g., mobile phones, smartwatches) that enhance human–machine interactions. From a connectivity viewpoint, they could be devices with legacy communication protocols or IoT standard protocols (e.g., OPC-UA, BLE, MQTT).

Edge Layer It transforms raw data generated at the device layer into information. Typically, powerful gateway devices are deployed at this layer. The gateway devices implement various edge analytics techniques such as data aggregation, data filteration, and data cleansing to further refine acquired data (some of the edge analytics tools are presented in Sect. 4).

Cyber Layer It acts as a distributed information hub, preparing a ground for specific data analytics. Diverse information could be collected from different players of a supply chain (e.g., logistics, distributors, suppliers), industrial machines on factory floors from edge devices. The information is pushed to form a linked network of information (Linked Data[1]). Linked data are a natural fit for the connected data as they provide abstraction on top of a distributed set of information.

[1] https://bit.ly/29YZz5b.

Data Analytic Layer The massive amount of data collected at the cyber layer creates an opportunity to apply industrial analytics, leveraging AI techniques. The aim is to identify an invisible relationship among data and enhance Industry 4.0 applications for better decision-making. The industrial analytics algorithms can be on premise (state of the art presented in Sect. 4.2) or cloud based (state of the art presented in Sect. 4.1).

Application Layer This layer builds meaningful and customized application on top of data and services exposed by the data analytic layer. In recent years, a wide variety of Industry 4.0 applications are demonstrated. For instance, developers can create digital twin by combining data from the data analytic layer and functionality exposed by an industrial machine. GE digital has demonstrated an advanced digital twin.[2] The customer can ask questions related to the machine's performance and potential issues through a natural language interface and receive the answers. Moreover, a manufacturer can interact with the digital twin through Microsoft HoloLense,[3] an augmented reality (AR) device, and the manufacturer can have a 3D view of an industrial asset to analyze its internal parts.

4 Related Work

This section presents existing approaches to implement the use case, presented in Sect. 2. The existing approaches are largely divided into three categories: (1) cloud-based approaches (Sect. 4.1), (2) open-source tools to develop an infrastructure that enables real-time analytics (Sect. 4.2), and (3) Semantic Web technologies to achieve the interoperability among industrial devices (Sect. 4.3).

4.1 Cloud Manufacturing

To realize the use case mentioned in Sect. 2, different cloud vendors (such as Microsoft Azure, AWS, Google, IBM) provide cloud-based services. A common approach, adopted by cloud, is to ingest data from an IoT device to cloud infrastructure. Then, all processing takes place on top of the ingested data, and appropriate decisions are made. The cloud-based approaches provide a set of services to implement industrial analytics solutions. The following present some of the cloud vendors and describe the services offered by them to implement smart industrial analytics:

[2] https://youtu.be/2dCz3oL2rTw.

[3] https://www.microsoft.com/en-us/hololens.

- **Microsoft Azure**.[4] It provides storage services (e.g., data lake) to store structured and unstructured data. Moreover, its streaming service allows the users to ingest data into the cloud from industrial devices. This service is supported by an analytic service to analyze the streaming data and to derive insight out of the streaming data. The analytic service component interfaces data visualization services to implement analytic dashboard, machine learning service to make predictions, and data lake services to store big data in various data formats.
- **Siemens Offers MindSphere**.[5] It is an Industrial Internet of Things/Industry 4.0 solution, hosted on AWS. Using this service, the users can connect various industrial devices. MindSphere provides marketplace of preconfigured industrial analytics solutions, using which the users can quickly prototype a solution.
- **GE Offers Predix**.[6] It is a cloud-based Industry 4.0 solution with a preconfigured industrial analytics solution (in form of preconfigured apps and machine learning solutions such as predictive maintenance). Moreover, Predix offers an operating system for Industry 4.0 devices that let manufacturers deploy intelligent algorithms at the edge.

Shortcomings of Cloud Manufacturing The cloud-based approaches keep industrial analytics solutions largely at a center (Patel et al. 2017, 2018). Thus, it is easy for maintenance. Moreover, it provides tools and technologies that reduce the application development efforts. However, it is not suitable for some of these Industrial Internet of Things applications. In the following, we present the shortcoming of cloud approaches:

- The cloud approaches rely on the constant Internet connectivity among Industry 4.0 devices and Cloud services. The Internet connectivity may not remain consistent, due to several reasons such as the manufacturing unit setup at remote places or at the area where enough infrastructure for the Internet is not available. Imagine a scenario, where an oil and gas unit is located at the seashore. Even if we accept the fact that the technologies advancements can address the Internet connectivity issues, there will always be concerns related to security and sharing data to the third-party cloud vendors.
- A "development environment" of a cloud vendor can be very specific to a platform, because each cloud platform brings its own platform-specific environment. This could be a problem when a developer wants to migrate a solution from one cloud provider to the other cloud provider. For instance, the developer may want to migrate his/her solution from Microsoft Azure IoT hub solution to AWS IoT solution. He/she may have to make changes in the cloud-based configuration and perhaps changes in the application's front-end code as well.

[4] https://azure.microsoft.com/en-us/industries/discrete-manufacturing/.

[5] https://siemens.mindsphere.io/.

[6] https://www.predix.io/catalog/services/.

- The innovation path may depend on “cloud vendor specific” offering. For instance, a manufacturer may not be able to customize certain cloud-specific features in a certain way if a cloud vendor is not offering that feature.

  A common practice of cloud manufacturing is that developer uses the on-premise tools and technologies (mentioned in Sect. 4.2) for initial prototyping of the solution. Then, the solutions are deployed in the cloud for better scalability, when there is an increase of customer base.

4.2 On-premise and Open-Source Approaches

A common pattern found in this approach is that sensor data are collected using Industry 4.0 standards such as OPC-UA, Modbus, MQTT, BLE. The collected data are sent to the more powerful devices such as gateway, which are responsible to aggregate data or to send control signals back to the devices. Moreover, the processed data are pushed to powerful servers, where the data are analyzed. Various machine learning algorithms are used to make predictions. A set of open-source technologies from the Eclipse foundations have been released to build such on-premise systems for Industry 4.0. In the following, we present some of the open-source tools to build on-premise solutions. Table 1 summarizes all these tools and technologies.

Table 1 Summary: open-source tools to build on-premise Industry 4.0 applications

Tools	Description	Layer
Ditto	It is a platform for building a digital twin. A digital twin is a virtual representation of its physical industrial asset (e.g., electric motor). The Ditto provides HTTP APIs to access industrial asset. Using Ditto platform, the developers do not need to know how or where the Industrial assets are connected, thus it simplifies the development of digital twins	Cloud, edge
Kura	It is a platform for building Industry 4.0 gateway devices. It implements the remote management of gateway devices, deployed in the factory. It provides various APIs that allow the developers to build and deploy customized logic at the gateway devices	Gateway
HONO	It provides service interfaces for connecting Industry 4.0 devices to a back end. It provides interfaces to interact with the devices in a uniform way regardless of the device communication protocol	Gateway, edge
Kapua	It is an open-source cloud platform to manage and integrate devices and their data.	Cloud, analytics
Unide	Unide stands for “Understand Industry Devices.” It is a Production Performance Management Protocol (PPMP), which is a lightweight server–client implementation using REST APIs and JSON	Edge

Ditto[7] It is an IoT technology to build "digital twin," which is a virtual representation of its real-world counterpart. For instance, a digital twin of an electric motor in a smart factory can collect data from a physical motor. The user can interact with the digital twin to know the current status of the motor. Eclipse Ditto provides high-level APIs, connecting devices to the back end and implementing business application on top of the high-level APIs.

Kura[8] It is a software platform (runs on the edge devices) for building IoT gateways. Eclipse Kura provides several services. These services include (1) I/O services to connect and access sensors and resource constrained devices such as microcontrollers, (2) Data services to store and forward the telemetry data collected by the sensors, (3) Cloud services to push data to cloud servers such as AWS and Azure, and (4) Kura wire services to customize logic on gateway devices. All these services are exposed by Web service interface.

HONO[9] HONO is an open-source remote service interface for connecting IoT devices to back-end services. It is a quite active project in the community with a lot of documentations and examples. The goal of HONO is to provide a platform to interact with devices regardless of communication protocols. The community has developed solutions for HTTP, MQTT, AMQP, and Kura. Moreover, it allows developers to plug custom device protocols, thus it does not limit the Industry 4.0 developers to only supported protocols. On top of Eclipse HONO, it provides a uniform interface to interact with underlying IoT devices regardless of the communication protocols they implement. HONO supports scalable and secure ingestion of sensor data, and its command control API allows to send and receive command message.

Unide[10] Unide stands for "Understand Industry Devices." It is a Production Performance Management Protocol (PPMP),[11] , which is a lightweight server–client implementation using REST APIs and JSON. Unide provides tools for the validation of PPMP messages and for visualization and persisting of PPMP data. It provides the public REST API with the purpose of receiving measurement and message data from machine. To validate PPMP messages, Unide offers a validator that compares the payload you send to the given JSON-schema. By sending HTTP-POST requests to the validator endpoint, you receive a message confirming whether the PPMP message is correctly written according to the specification.

Kapua[12] The goal of the Kapua project is to provide an open-source cloud-based IoT integration platform. The Kapua is a platform to integrate data from various IoT

[7] https://www.eclipse.org/ditto/.

[8] https://www.eclipse.org/kura/.

[9] https://www.eclipse.org/hono/.

[10] https://www.eclipse.org/unide/.

[11] https://www.eclipse.org/unide/specification/.

[12] https://www.eclipse.org/kapua/.

devices. The Kapua provides a comprehensive management of IoT devices. The management services include the connectivity to IoT devices supporting different ingestion mechanisms, device configuration remotely, application development on top of the Kapua APIs, controlling the device remotely using appropriate access mechanisms, and sending device updates. The Kapua tools can be combined with the Kura project to develop an end-to-end Industry 4.0 solution. This would accelerate community-driven open-source implementation and avoid proprietary vendor lock-in.

We continue leveraging our IoT tools to implement the discussed application: IoTSuite (Chauhan et al. 2016), a tool suite to develop IoT application rapidly; SWoTSuite, a tool suite to implement Semantic Web of Things applications; and a middleware (Alie et al. 2017) for real-time analytics to implement essential Industry 4.0 components. In the following, we present them briefly:

IoTSuite[13] The objective of this programming framework is to make the application development easy by hiding IoT development-related complexity. It provides high-level and platform independent programming abstractions and specification. The developer specifies high-level specification, which is parsed by IoTSuite to generate the platform-specific code. The high-level specification includes the specification about sensing, actuating, and computational components as well as the device properties. The developers do not need to concern about the platform and runtime-specific aspect of development. More specifically, the following key characteristics of this tool suite make it suitable for building real-time industrial analytics.

- The current version of IoTSuite generates code in C, Python, Java, Android, and Node.js. The code generator is flexible to generate IoT framework in a new programming language. The developers just need to write a small plug-in to generate IoT framework code in new programming language. IoTSuite has been tested on devices such as Raspberry PI, ABB's RIO 600, Arduino, and Android smartphones.
- The current version of IoTSuite plugs MQTT, WebSocket runtime. However, the integration of a new runtime is easy. The IoTSuite simply exposes well-defined interfaces (Soukaras et al. 2015) to integrate a new runtime. The developers simply implement runtime-specific interfaces to plug a target runtime system.

SWoTSuite[14] It is a framework intended to build cross-domain IoT applications by leveraging semantic technologies to achieve interoperability among heterogeneous IoT systems. The SWoTSuite reasoning mechanism over semantically annotated IoT data generates user suggestions. The framework applies Linked Open Data (LOD), Linked Open Vocabularies (LOV), and Linked Open Service

[13] https://github.com/pankeshpatel/IoTSuite/wiki.

[14] https://github.com/pankeshpatel/SWoTSuite.

(LOS) to achieve interoperability and derive meaningful knowledge from annotated data (Gyrard et al. 2016).

ACEIS[15] It contains a set of tools, designed for IoT data analytics. It leverages Semantic Web technologies to build various components including one each for integration on the fly, event detection, and streaming data discovery (Gao et al. 2017).

4.3 Semantic Web Technologies for Industry 4.0

This section presents Semantic Web tools and technologies to achieve interoperability among Industry 4.0 devices. In the following, we present Semantic Web components that can be used to implement the use case, mentioned in Sect. 2.

Data Ingestion Data ingestion is a process of getting data into an analytic platform. It ingests sensor data for further processing and device description for discovery. The data collection could be in various semantic formats such as JSON, EXI, XML.

Data Representation A common data representation format such as RDF could be used for data exchange among industrial devices. The work (Grangel-González et al. 2016) notes several benefits of employing RDF as data representation format for Industry 4.0.: first, various data serialization formats are generated easily and transmitted. Second, the data representation can be generated on the fly from the data stored in relational database or other data representation formats. This is very important aspect, because this flexibility enables data sharing among legacy systems and new systems. Third, SPARQL (a W3C standard for an RDF query) can be used on top of RDF data. This would make data available through a standard interface. However, Industry 4.0 devices such as PLCs may not have enough processing power to process RDF data.

The work (Su et al. 2015) emphasizes adding semantic technologies on devices and evaluates several different formats for representing sensor measurements and device properties in terms of energy efficiency for data communication and processing. The evaluation conducted by (Su et al. 2012) finds that JSON for Linked Data (JSON-LD[16]) and Entity Notation (EN) are compact as well as lightweight data representations. Many non-RDF lightweight emerging standards are available for representing industrial devices and sensor measurements in the domain of Industry 4.0. In the following, we present some of the Industry 4.0 standards for representing data from industrial devices.

[15] https://github.com/CityPulse/Stream-Discovery-and-Integration-Middleware.

[16] http://json-ld.org/.

- **OPC Unified Architecture (OPC-UA).**[17] It is a machine-to-machine Industry 4.0 protocol. It integrates an information model for information integration. Using OPC-UA information model, the complex data can be modeled.
- **Production Performance Management Protocol (PPMP).**[18] It can be challenging for SMEs to catch up with the complex IT standards such as OPC-UA that come along. To address these challenges and requirements in Industry 4.0, PPMP is designed. It specifies a format that allows capturing data for performance analysis of production facilities. It is structured into three payload formats: measurement payload, message payload, and process payload. The measurement payload contains measurements from machines (e.g., temperature, vibrations of a machine). The message payload contains alerts sent by a machine. A process message consists of information (e.g., tightening process with all their characterizing data), which is needed to describe and analyze it. The Eclipse Unide aims to provide sample implementations and further development of PPMP in and with the Eclipse Open Source community.

Data Transformation This component is responsible for transforming various formats to standardized format. It enables reasoning of sensor data in a uniform way. For instance, the work by Su et al. (2014) transforms Sensor Markup Language (SenML)[19] to RDF. SenML is an industry-driven lightweight solution for representing sensor measurements, accepted by many industrial vendors. Eclipse Unide presents an open-source implementation that transforms PRC7000 format to PPMP format using Apache Camel, Which is a versatile open-source integration framework.

Data Storage and Processing This component is responsible for storing and processing data. Broadly, there are two approaches: first, the use of cloud for processing. Second, edge computing that stores process data locally. The RDF storage on resource-constrained devices may not possible due to the textual representation of RDF. To address this problem, formats such as Binary XML and EXI are promising compact representation, proposed by W3C. The work (Le-Phuoc et al. 2010) proposes "RDF on the Go" that offers a full-fledged RDF storage for Android devices. Similarly, MicroJena and MobileRDF[20] present an approach to store and query RDF data locally.

Reasoning at Edge To derive new knowledge, it is necessary to push reasoning on the edge. However, existing reasoning tools such as RacerPro, Jena, Fact++, Pellet cannot be used for edge devices, due to their high computation cost. The work (d'Aquin et al. 2010) demonstrates reasoning engines require several hundred KBs of memory to process one RDF triple. Thus, technically it is possible to port

[17] https://opcfoundation.org/about/opc-technologies/opc-ua/.

[18] https://www.eclipse.org/unide/.

[19] https://tools.ietf.org/html/draft-jennings-senml-10.

[20] http://www.hedenus.de/rdf/.

a reasoner on device with some code-level modification, a reasoning engine can consume a huge amount of resources (Tai et al. 2015).

5 Our Approach and Implementation

This section presents our approach to achieve the objectives of the case study, described in Sect. 2. In the following, we present various data analytics steps, performed on industrial data.

5.1 Data Ingestion

This is an entry point of getting data into the platform. This module has two major roles to play: first, scale to meet the demand of diverse data sources including relational/non-relational database as well as real-time data. Second, move data as fast as possible to the next module for further processing. This module collects data in various formats (e.g., JSON, RDF, XML), discussed in Sect. 4.3. We use Apache Kafka[21] for data ingestion service. Kafka provides a set of standard connectors[22] to query the relevant databases directly, following traditional ETL (Extract–Transform–Load) pattern as well as connectors to ingest real-time data that exhibit a number of interaction patterns such as request–response (Berners-Lee et al. 2001), publish–subscribe (Eugster et al. 2003), and streaming (Aggarwal et al. 2006). Depending on the nature of the underlying information source and the data policy, this module performs either a full ETL on the whole dataset or partial data are acquired using an on-demand ETL policy.

5.2 Data Pre-processing

As a first step, we identified the relevant variables that are important for production forecasting and selected a set of dependent and independent variables. The extracted data spanned over the last three years. Table 2 describes the selected variables, which are collected at each manufacturing step. For the purposes of our analysis, we considered three independent variables, namely (1) Scrap: the number of units scrapped during production, (2) Rework: the number of units sent back for reworking, and (3) Lead time: the overall time it takes for a container to be processed between the first and last operations. We use query-based approach to extract data,

[21] https://kafka.apache.org/.

[22] https://www.confluent.io/connectors/.

Table 2 Selected variables for production forecasting

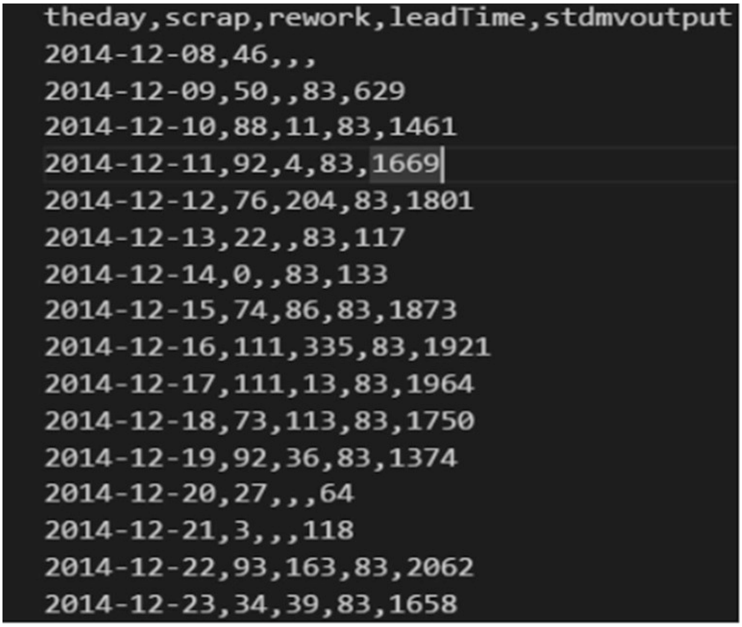
Variables	Description
Scrap	It is a number of units scrapped during the manufacturing process
Rework	It is a number of units (in a given container) that are sent back for reprocessing through a manufacturing operation step
Lead time	It is a total time of a manufacturing process, from process start to finish, including any queue times
Operation process time	It is an actual processing time of a container, including container on-hold time
Operation queue time	It is an actual queued time of a container, before entering the next operation step
Machine uptime	It indicates a time during the manufacturing process a machine is in a productive state
NCR occurrences	It is an event, when a container is non-conforming
Containers on hold	It is a state of the manufacturing process when a container is placed "on-hold," requiring further investigation
Sample tests failed	It indicates a number of samples, pulled for test purposes for the quality control, that have failed an inspection step

in such a way that any future versions of the database can be easily linked to our tool.

We analyzed the extracted data manually and ensured that the prepared data are properly cleaned and free of any missing values or any discrepancies. Figure 4 presents a snapshot of collected data. This process was the most time-consuming part. We leverage a variety of tools for data cleansing including anomaly detection, handling incomplete and noisy data, identifying any missing values, contradictory and out of range values, and an automated features extraction tool to identify relevant features before applying to the next step, discussed in Sect. 5.3.

5.3 Machine Learning Algorithms for Prediction

We applied regression-based ML algorithms over collected data to identify the best performing algorithms. We used models such as Multiple Linear Regression, Support Vector Regression, Decision Tree Regression, and Random Forest Regression. The models were trained using 80% training dataset, and 20% validation dataset was used for testing.



```
theday,scrap,rework,leadTime,stdmvoutput
2014-12-08,46,,,
2014-12-09,50,,83,629
2014-12-10,88,11,83,1461
2014-12-11,92,4,83,1669
2014-12-12,76,204,83,1801
2014-12-13,22,,83,117
2014-12-14,0,,83,133
2014-12-15,74,86,83,1873
2014-12-16,111,335,83,1921
2014-12-17,111,13,83,1964
2014-12-18,73,113,83,1750
2014-12-19,92,36,83,1374
2014-12-20,27,,,64
2014-12-21,3,,,118
2014-12-22,93,163,83,2062
2014-12-23,34,39,83,1658
```

Fig. 4 Data collection from MES before data pre-processing step

Figure 5a–d presents the results of our evaluation of the four, respectively. The results demonstrate a comparison between the actual value (as a blue line) and predicted value (as an orange line) for the number of units manufactured during 6 months. We employed Root Mean Square Error (RMSE) mechanism to evaluate the accuracy of an algorithm. This mechanism helped us to select a most appropriate algorithm for our use case. RMSE shows how close a trained model is to a set of actual points. This is calculated by taking the distances from the points to the regression lines and squaring them before taking the root for the final value. The smaller RMSE indicates a best fit. Table 3 presents the results of RMSE value of each regression ML algorithm. As we can see that, Random Forest model shows the smallest RMSE value.

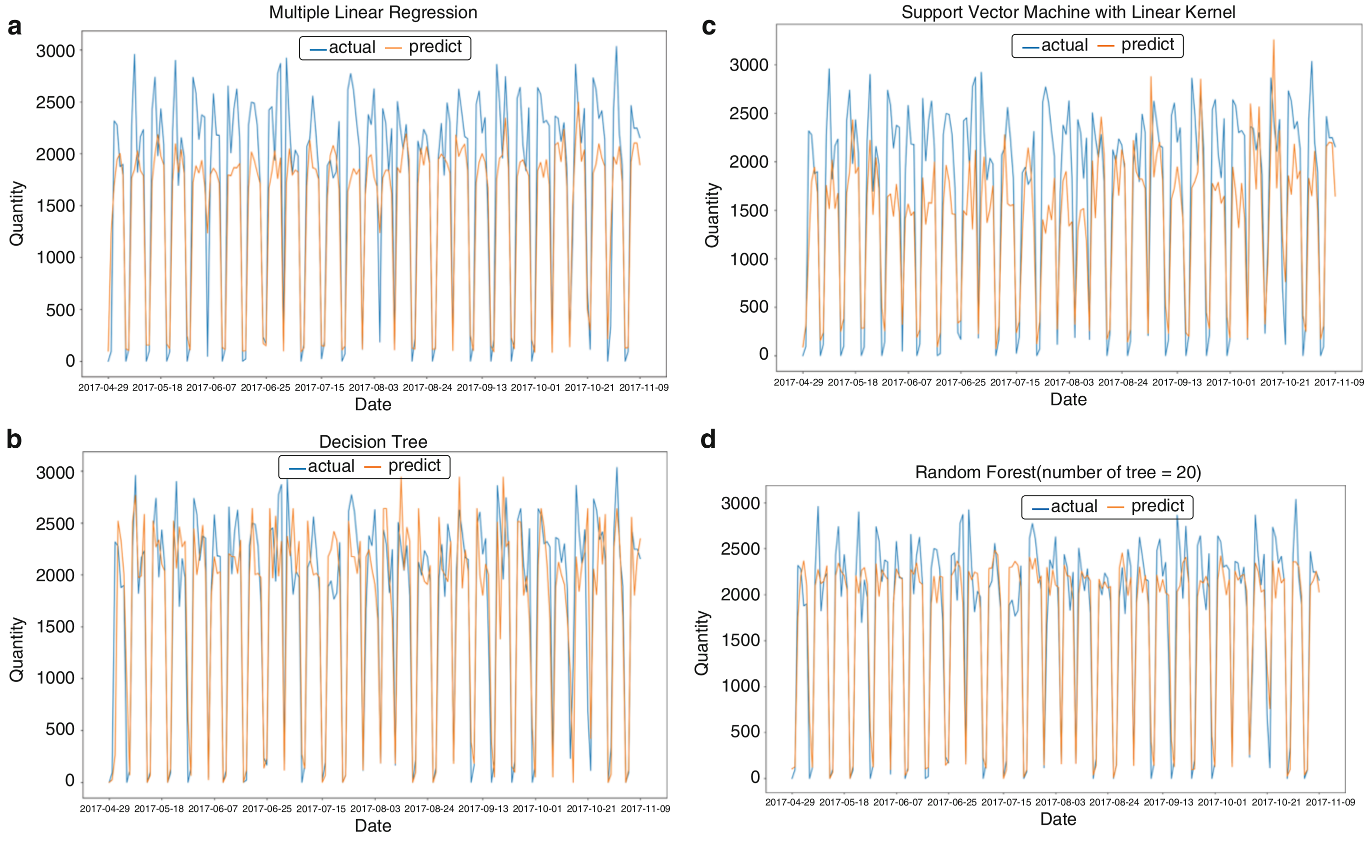


Fig. 5 Results of predictions using different machine learning algorithms. (**a**) Multiple linear regression. (**b**) Support vector regression. (**c**) Decision tree regression. (**d**) Random forest regression

Table 3 RMSE scores for different regression models

Regression types	RMSE
Multiple linear	467.89
Support vector	587.84
Decision tree	434.54
Random forest ($n = 20$)*	312.37

*The most accurate algorithm with lowest RMSE score

5.4 Real-Time Event Monitoring and Notification

The limitation of the existing system at the factory is that there was lack of tools to visual data in real time, to set production targets for the supervisors, and to record events that could affect the overall production and their causation. To address the limitation, we have developed a set of tools. To detect events, we set different thresholds based on historical data analysis and domain knowledge from staff members of the organization.

The major benefit of real-time analytics is to support a detection of events. An event can be defined in various ways, such as continuously looking for the occurrence of any predefined pattern or continuously monitoring the streaming data values and trigger an event whenever a pre-specified threshold for the values is breached. This module implements a real-time event detection mechanism for streaming data. A set of predefined thresholds are used for production data for granular time interval. We use the live production data to monitor real-time events and then continuously analyzed and matched values of the production data by comparing the values against the predefined thresholds. We reported an event whenever the values observed from the live data streams go beyond the defined threshold values. We also introduce a buffering mechanism, which ensures that events are generated only when the live production data deviate beyond the threshold by a certain margin, e.g., ±5% of daily average production.

We developed a set of tools to monitor, detect, and report events. In the following, we present each module in detail:

Real-Time Analytic Module The first goal of a real-time analytics module is to capture data in real time and visualize them on a dashboard. The second goal is to set realistic production target and use these thresholds for deviation detection. The third goal is to alert supervisors at a factory when a processing step deviates from a predefined target. If the threshold condition is not met, then deviated containers are recorded and notifications are sent to supervisors.

Target Definition and Threshold Setting The objective of this module is to alert users when a processing step deviates from a predefined target. To achieve this objective, we defined a realistic target and use it as threshold for deviation detection. We used parts per minute (PPM) to define as a target for each type of product. However, we leveraged the outcomes of historical analysis and used the predicated values automatically to define targets. We also provided a web interface to allow shift supervisor to set goals of each shift manually and to log the reasons if a target is increased or decreased from suggested target.

Event Detection and Event Logging In order to provide a mechanism for event detection, we build a tool to log all detected events. The UI in the tool, lets supervisors to input why deviations happened are provided by the supervisors. These reasons can provide later on an additional information for the historical analysis model. This additional column perhaps helps the prediction model more precise. We used the following notations for event detection:

- P: It is a process that is defined as a set of work-flow steps. Each P is assigned with a target T.
- $R = \{r_1, r_2, \ldots, r_n\}$ is a set of reasons that are either defined by users or detected automatically by the system. Each reason r_i can have a positive or negative effect on target T. Let $f(r_i, T)$ be the effect value that r_i produces on T, where $f(r_i, T) > 0$ $(f(r_i, T) < 0)$ represents a positive (negative) effect.

Given a target T and a set of reasons R. Assume that each reason in R holds a different level of effect on the overall target, i.e., some reasons can adversely affect the overall target more or less compared to another. Hence, different weights are added to each reason. Any R can have either a positive or a negative effect on T, which can be calculated based on the following formula:

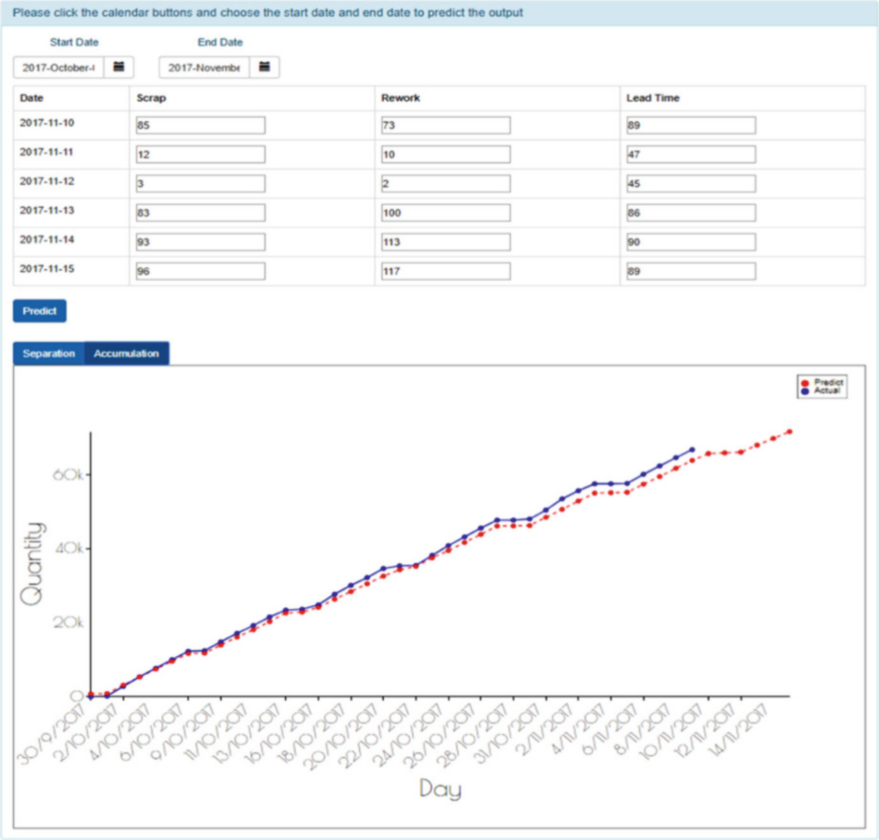
$$\text{f(R, T)} = \frac{\sum_{i=1}^{n} w_i f(r_i, T)}{\sum_{i=1}^{n} W_i} > 0 \text{ (or } < 0),$$

where $\mathrm{w}_1, \ldots, \mathrm{w}_n$ are the weights of the contributions of reasons $\mathrm{r}_1, \ldots, \mathrm{r}_n$, respectively.

Alerting and Notification This module is responsible for the generation of notification whenever an event is detected. Upon the detection of an event, this module triggers an action for the detected event. The action can be either a notification, an alarm, an alert or even an email to the relevant person who can take appropriate action. We implemented two types of notification delivery methods: first, an alert system was integrated within the dashboard. The factory supervisor was able to monitor the real-time progress of the production by following a visual interface installed at the shop floor. Second, a system-generated email is sent to the selected managers whenever there are any unexpected events.

5.5 *Capacity Planning Tool for Production Forecasting*

We developed a capacity planning tool facilitating the factory managers to set long-term production targets. Figure 6 shows the production forecasting results, where the blue line is an actual production and red line indicates the predicted values. Moreover, this tool lets the managers to adjust the values of different dependent variables to perform what-if analysis. Historical data analysis provides an estimated value for each of the days as auto-filled value that can be changed by the manager to see an impact of the change.



Please click the calendar buttons and choose the start date and end date to predict the output

Start Date: 2017-October-
End Date: 2017-Novembe

Date	Scrap	Rework	Lead Time
2017-11-10	85	73	89
2017-11-11	12	10	47
2017-11-12	3	2	45
2017-11-13	83	100	86
2017-11-14	93	113	90
2017-11-15	96	117	89

Fig. 6 An interface for accumulative capacity planning using production forecasting

6 Conclusion and Future Work

In this chapter, we presented an approach for event detection for smart manufacturing in real time. We presented a use case for the application of data analytics in the context of smart manufacturing. We reviewed the existing practices and solutions supported by the industry and discussed the key challenges faced while designing Industry 4.0 applications. We presented detailed components of our proposed approach, which can collect, integrate, and analyze historical as well as real-time data. We showcase the practical use of our approach by showing how an industry's use case was implemented using our proposed solution.

The proposed approach has been successfully deployed at a manufacturing unit as a prototype. We consider it as the first step for the organization toward building a larger vision of Industry 4.0. We plan to extend this deployment for all processes within the factory and design more business intelligence tools. Particularly, we will focus on the integration of multiple autonomous systems and show the integration and analytics of data collected from disperse autonomous systems for the supply chain management and manufacturing processes optimization.

Acknowledgments This publication has emanated from research supported in part by a research grant from Science Foundation Ireland under Grant Numbers 16/RC/3918 (CONFIRM SFI Research Centre for Smart Manufacturing) and 12/RC/2289_2 (Insight SFI Research Centre for Data Analytics), co-funded by the European Regional Development Fund. We thank Chan Le Van for his valuable contribution to acquire and process data provided by the industry partner.

References

Aggarwal, C. C. (2006). *Data streams: Models and algorithms (advances in database systems)*. New York, NY: Springer.

Ali, M. I., Ono, N., Kaysar, M., Shamszaman, Z. U., Pham, T. L., Gao, F., et al. (2017). Real-time data analytics and event detection for IoT-enabled communication systems. *Web Semantics: Science, Services and Agents on the World Wide Web, 42*, 19–37.

Berson, A., & Smith, S. J. (1997). *Data warehousing, data mining, and OLAP*. New York, NY: McGraw-Hill.

Berners-Lee, T., Hendler, J., Lassila, O., et al. (2001). The semantic web. *Scientific American, 284*(5), 28–37.

Chauhan, S., Patel, P., Sureka, A., Delicato, F. C., & Chaudhary, S. (2016). IoTSuite: A framework to design, implement, and deploy IoT applications: Demonstration abstract. In *Proceedings of the 15th International Conference on Information Processing in Sensor Networks, IPSN '16* (pp. 37:1–37:2). Piscataway, NJ: IEEE Press.

d'Aquin, M., Nikolov, A., & Motta, E. (2010). *How much semantic data on small devices?* (pp. 565–575). Berlin: Springer.

Eugster, P. T., Felber, P. A., Guerraoui, R., & Kermarrec, A. M. (2003). The many faces of publish/subscribe. *ACM Computing Surveys, 35*(2), 114–131.

Gao, F., Ali, M. I., Curry, E., & Mileo, A. (2017). Automated discovery and integration of semantic urban data streams: The ACEIS middleware. *Future Generation Computer Systems, 76*, 561–581.

Grangel-González, I., Halilaj, L., Coskun, G., Auer, S., Collarana, D., & Hoffmeister, M. (2016). Towards a semantic administrative shell for Industry 4.0 components. arXiv preprint arXiv:1601.01556.

Gyrard, A., Patel, P., Sheth, A. P., & Serrano, M. (2016). Building the web of knowledge with smart IoT applications. *IEEE Intelligent Systems, 31*(5), 83–88.

Intizar, M., Patel, P., Datta, S. K., & Gyrard, A. (2017). Multi-layer cross domain reasoning over distributed autonomous IoT applications. *Open Journal of Internet of Things, 3*.

Lee, J., Bagheri, B., & Kao, H. A. (2015). A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters, 3*, 18–23.

Lee, J., Kao, H. A., & Yang, S. (2014). Service innovation and smart analytics for industry 4.0 and big data environment. *Procedia Cirp, 16*, 3–8.

Le-Phuoc, D., Parreira, J. X., Reynolds, V., & Hauswirth, M. (2010). RDF on the go: An RDF storage and query processor for mobile devices. In *Proceedings of the 2010 International Conference on Posters &#38; Demonstrations Track—Volume 658*, ISWC-PD'10 (pp. 149–152). Aachen: CEUR-WS.org

Patel, P., Ali, M. I., & Sheth, A. (2018). From raw data to smart manufacturing: AI and Semantic Web of Things for Industry 4.0. *IEEE Intelligent Systems, 33*(4), 79–86.

Patel, P., Ali, M. I., & Sheth, A. (2017). On using the intelligent edge for IoT analytics. *IEEE Intelligent Systems, 32*(5), 64–69.

Soukaras, D., Patel, P., Song, H., & Chaudhary, S. (2015). IoTSuite: A toolsuite for prototyping internet of things applications. In *The 4th International Workshop on Computing and Networking for Internet of Things (ComNet-IoT), Co-Located with 16th International Conference on Distributed Computing and Networking (ICDCN)* (p. 6).

Su, X., Riekki, J., Nurminen, J. K., Nieminen, J., & Koskimies, M. (2015). Adding semantics to internet of things. *Concurrency and Computation: Practice and Experience, 27*(8), 1844–1860 (2015)

Su, X., Riekki, J., & Haverinen, J. (2012). Entity notation: Enabling knowledge representations for resource-constrained sensors. *Personal and Ubiquitous Computing, 16*(7), 819–834.

Su, X., Zhang, H., Riekki, J., Keränen, A., Nurminen, J. K., & Du, L. (2014). Connecting IoT sensors to knowledge-based systems by transforming SenML to RDF. *Procedia Computer Science, 32*, 215–222.

Tai, W., Keeney, J., & O'Sullivan, D. (2015). Resource-constrained reasoning using a reasoner composition approach. *Semantic Web, 6*(1), 35–59.

Wang, S., Wan, J., Zhang, D., Li, D., & Zhang, C. (2016). Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination. *Computer Networks, 101*, 158–168.

Watson, H. J., & Wixom, B. H. (2007). The current state of business intelligence. *Computer, 40*(9), 96–99

Zhong, R. Y., Xu, C., Chen, C., & Huang, G. Q. (2017). Big data analytics for physical internet-based intelligent manufacturing shop floors. *International Journal of Production Research, 55*(9), 2610–2621.