Middleware for Real-Time Event Detection and Predictive Analytics in Smart Manufacturing

Muhammad Intizar Ali*, Pankesh Patel†, John G. Breslin* *Senior Member, IEEE*
* Confirm SFI Research Centre for Smart Manufacturing,
Data Science Institute, National University of Ireland Galway, Ireland
Email: {ali.intizar, john.breslin}@nuigalway.ie
†Fraunhofer USA – CESE, College Park, Maryland, USA
Email: dr.pankesh.patel@gmail.com

***Abstract*—Industry 4.0 is a recent trend of automation for manufacturing technologies and represents the fourth industrial revolution which transforms current industrial processes with the use of technologies such as automation, data analytics, cyber-physical systems, IoT, artificial intelligence, etc. The vision of Industry 4.0 is to build an end-to-end industrial transformation with the support of digitization. Data analytics plays a key role to get a better understanding of business processes and to design intelligent decision support systems. However, a key challenge faced by industry is to integrate multiple autonomous processes, machines and businesses to get an integrated view for data analytics activities. Another challenge is to develop methods and mechanisms for real-time data acquisition and analytics on-the-fly.**

In this paper, we propose a semantically interoperable framework for historical data analysis combined with real-time data acquisition, event detection, and real-time data analytics for very precise production forecasting within a manufacturing unit. Besides historical data analysis techniques, our middleware is capable of collecting data from diverse autonomous applications and operations in real time using various IoT devices, analyzing the collected data on the fly, and evaluating the impact of any detected unexpected events. Using semantic technologies we integrate multiple autonomous systems (e.g. production system, supply chain management and open data). The outcome of real-time data analytics is used in combination with machine learning models trained over historical data in order to precisely forecast production in a manufacturing unit in real time. We also present our key findings and challenges faced while deploying our solution in real industrial settings for a large manufacturing unit.

I. Introduction

Within the vision of Industry 4.0, a key goal is to use data analytics as the main driving force for businesses to make intelligent decisions [7]. Traditional data warehousing approaches used in industry rely on historical data to compile reports, whereby a view of the past is used for making better future decisions [2]. With the latest advancements in the Internet of Things (IoT), it is now possible to collect data in real time, not only related to key business process, but also around the context and environmental conditions. A combination of historical data, real-time business process data and IoT-based contextual data can certainly improve the outcomes of data algorithms [12]. A few middleware solutions for real-time data analytics have been already presented within a single domain such as smart cities [4] , or even across multiple domains such as a combination of smart cities and smart homes [6]. However, in smart manufacturing, mostly a variety of in-house historical data is used to build intelligent systems for manufacturing shop floors [13].

In this paper, we discuss a use case of production forecasting for a manufacturing unit, and elaborate our experience and approach to design a real-time data analytics-based production forecasting tool. We have used historical data related to production processes to train machine learning algorithms (regression models) for better predictions about future production goals. These production forecasting models help to set realistic and optimal targets for production. Contrary to traditional machine learning algorithms which consider only historical data patterns, we support a real-time monitoring and events detection approach which can detect abnormal events (e.g. machine breakages, head count shortages, unavailability of raw materials, etc.). We calculated the impact of all such abnormal events and designed an integrated framework which can adjust the hourly, daily and weekly production targets accordingly. With integrated real-time monitoring techniques, we can automatically detect the impact of any event and trigger a notification about remedial actions. Having a real-time integrated middleware provides optimal production forecasting at very granular time intervals.

Our main objective in this paper is to share our experience of applying modern IoT and data analytics approaches for a traditional manufacturing unit. We discuss key data-related challenges faced by the industry and then advocate how our middleware can improve upon state-of-the-art technologies to yield better results.

A Real-World Case Study - Real-Time Production Forecasting. We consider a real-world use case of a large biomedical devices manufacturing company. The company produces orthopedic devices and specializes in knee, hip and shoulder joint replacements. The company has a large distributed infrastructure with multiple manufacturing units installed at different locations within Ireland and globally.

A typical production line on a shop floor in these manufacturing units is sequential (e.g. an assembly line). In such sequential processes, any kind of anomaly at any stage can lead to a chain reaction or domino effect on subsequent processes. The company has an internal Manufacturing Execution

System (MES), which keeps track of their daily processes and stores relevant information at each processing step for the produced products. The collected data is used to generate periodic reports giving a summary of actual production during a time frame in the past. These reports are used to set future production targets. In the era of technology, businesses are more and more interested in adaptive approaches which can automatically adjust goals and targets based on current progress, e.g. the system should be able to automatically reduce daily targets if there is any unexpected event like machine failure [11]. In order to ensure the maximum utilization of available resources daily or even hourly, production goals can be adjusted based on the current situation. With IoT and sensor data, it is now possible to collect data in real time. However, current data collection systems have a few quite significant issues, as mentioned below:

- Data collected during separate processes is not interconnected and interoperable, resulting in silos of information being created for each process. However, to ensure accurate prediction, we need to integrate data from all relevant processes.
- The collected data is only used for the generation of periodic reports giving insights about past events, and systems are currently unable to incorporate real-time data and events for up-to-date reports and feedback.
- Domain knowledge is not properly captured, and usually an experience-based learning approach is used by operational staff.A lack of domain knowledge makes it harder to properly analyze such limited data.
- Sensors installed within various machines produce a lot of real-time data. However, this data is often not stored for later analysis, which results in a lack of seed data for training data models for real-time analysis.

II. Key Challenges: Industrial Data Analytics

In this section, we outline a few key challenges faced by industrial data analytics systems, and discuss how our approach can help to address these challenges.

Partial or Incomplete Data. Recent technology advances in cloud storage and data analysis have created many opportunities to utilize the whole gamut of data generated from a factory floor. Technology advances can not only be used to monitor the health of machine, but also to predict when a machine is likely to fail or malfunction. However, in reality – as simple as it sounds in principle – it is often difficult to come by all the necessary data to make predictions, thus making the collected data unsuitable for modeling. The main reason behind not having enough data is due to manual paper-driven processes where data is lost after a certain amount of time. In order to avoid issues around partial and incomplete data, the best option is to facilitate factory workers by providing very easy-to-use interfaces for data collection. Factory workers can create digital repositories of data which can be used by many applications within a smart factory.

Lack of Comprehensive Data. Collecting partial shop floor data leads to imprecise predictions and incomplete learning. Therefore, it is necessary that the data should be collected from all the relevant machines or processes of a factory. For example, if we want to predict a production count accurately, it is necessary that the data should be captured from all production processes, starting from a raw material to final cleaning and packing processes. A model that learns from such rich data would be able to identify dependencies and may find patterns to predict a production count accurately. In machine learning techniques, it is usually not easy to estimate the right data variables to train learning models, and mostly a hit-and-miss method is used with different combinations of variables to see which combination leads towards accurate results. With smart factory concepts and the Industry 4.0 vision, a good approach is to collect end-to-end data about all processes and relevant variables. Having the maximum amount of data will help modern machine learning and deep algorithms to get a good understanding of patterns, trends and processes using a complete and comprehensive set of data.

Lack of Domain Knowledge. When building a solution, a clear understanding of business requirements and the domain is necessary. For example, in order to build a model to predict production counts accurately, significant domain expertise is necessary in order to identify relevant points in the captured data, and the tolerances for false positives and negatives. Failure to predict accurately can be costly. Hence, data models for predictive analytics must be carefully tuned with a high level of assurance regarding the correctness of results with high recall and precision. While a better understanding of any business domain expertise is key, unfortunately little effort is made to capture this domain knowledge. The best way for capturing domain knowledge is to constantly engage domain experts and facilitate solution developers by providing them with tools to accurately capture and represent domain knowledge. The Semantic Web and related technologies have proven themselves to be useful techniques for capturing domain knowledge and for building intelligent applications that utilize the resulting knowledge base.

III. Background

Existing industrial analytics approaches are largely divided into two broad categories [5]: offline analytics and online analytics.

Offline Analytics. An industrial analytics process typically starts with an offline exploration. The data from industrial machines within a selected time period are collected. Most commonly, the collected data is transferred to a storage infrastructure for further analysis. Then, the stored data is analyzed by applying various data analysis methods. From various methods, data scientists select the best option that can provide good, actionable insights.

A typical approach for an offline analytics process starts with developing domain understanding, business requirements and use cases. The second step is the data exploration step, which is about understanding existing data sets first (using various techniques such as data visualization) and refining noisy data. The third step is about performing a dimensional reduction of columns based on the objectives of the use case and business requirements, learned from the first step. The fourth step is focused on either selecting existing models or building a new model. Various models may be tried and applied to learn the outcomes. After several iterations, the right model may be selected. Finally, the last step is about validation. This involves presenting results to users and taking feedback from them for further refinement, if necessary.

Online Analytics. As more and more data is generated from industrial sensors and devices, transforming these streams into actionable insights is a must-have need of various time-critical Industry 4.0 applications such as predictive maintenance, real-time production planning, and so on. Online analytics (or stream analytics) starts with a set of input data stream(s), sourced from industrial machines. On top of these data streams, a query, specifying what (e.g., data, event, patterns) to look for, is used to filter and aggregate data. After preparing this data for analytics, the data can be ingested into analytical models to derive insights from it. The transformed data can then be consumed by applications for decision makers such as in dashboards for real-time visualization.

IV. Middleware for Real-Time Event Detection and Predictive Analytics

In this section, we present our middleware that is designed to enable real-time data analytics for smart manufacturing. Figure 1 represents an architectural overview of our middleware. We broadly divide our middleware into two main layers: (i) a historical data analysis layer; and (ii) a real-time data analysis layer. In the following subsections, we will discuss some components of the middleware layers.

A. Historical Analysis Layer

This layer is mainly concerned with the processing of historical data stored in the database. Some major components in this layer are:

ETL: This component is responsible for establishing the connection between our middleware and traditional databases including relational databases, data warehouses and/or No-SQL databases. We developed a set of standard connectors which can directly query the relevant databases. We follow traditional ETL (extract-load-transform) processes to acquire data from existing data sources.

Data Pre-Processor: This module is responsible for a standard set of data cleansing operations to perform anomaly detection, missing value replacements, fault corrections or fixing out-of-range values.

Machine Learning Module: This module acts as a core component for historical data analytics. We built a generic component which can perform data intake in the form of vectors and can apply supported machine learning algorithms, training the most suitable model based on application requirements.

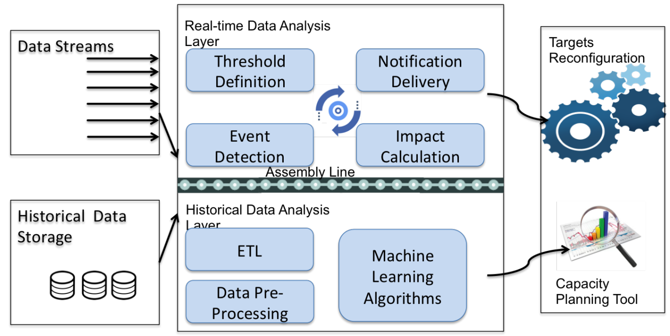


Fig. 1. Middleware for Real-Time Data Analytics for Production Forecasting

B. Real-Time Data Analysis Layer

In this layer, we developed a set of modules which can identify real-time events and evaluate their impact. Components in this layer are:

Event Detection: In this module, we define mechanisms for real-time event detection when using streaming data. A set of pre-defined thresholds are used for each type of product and its associated production data for granular time intervals (e.g. hourly in our case). The monitoring mechanism for real-time event detection uses live production data, evaluates the production data values against the pre-defined thresholds, and reports an event if the production values deviate beyond the thresholds. We also introduce a buffering mechanism which ensures that events are generated only when the live production data deviates beyond the threshold by a certain margin, e.g. +/- 5 percent of daily average production.

Impact Calculation: This module is responsible for calculating the impact of any unexpected event on the performance of overall factory operations. In our case, we calculated the impact of each event by comparing the production data of a day that had an unexpected event with the average daily production. In cases where no historical data related to events is stored, this module helps to gather new insights related to the impact of each unexpected event.

Notification Generation & Delivery: We developed a notification generation mechanism which can deliver a notification to the relevant person whenever an unexpected event is detected. We provided various methods of notification delivery including pop-up notifications, alarms and a system-generated email.

Targets Reconfiguration: This is a very crucial component for modern autonomous and reconfigurable assembly lines within a smart factory. Our middleware provides real-time insights of a factory's operation, which are used by this module to automatically reconfigure machines, adjust daily targets, and/or increase production capabilities based on the outcome of the real-time analytics process.

Attributes	Description
Scrap	Number of units scrapped.
Rework	Number of units in a given container that are sent back for reprocessing through some operation steps.
Lead Time	Total time for a full process from start to finish, including any queue times.
Operation Process Time	Actual time a container is being processed. (This includes containers that are on hold.)
Operation Queue Time	Actual time a container is queued for before entering the next operation step.
Machine Uptime	Time that a machine is in a productive state.
NCR Occurrences	Any event where a container is non-conforming.
Containers On Hold	When a container is place "On Hold", pending further investigation.
Sample Tests Failed	Samples pulled for test purposes that have failed an inspection step.

TABLE I
MOST SUITABLE VARIABLES FOR PRODUCTION FORECASTING

V. INDUSTRIAL CASE STUDY: REAL-TIME PRODUCTION FORECASTING

In this section, we present our work on a real-world case study with the use case already defined earlier. We discuss various data processing and analytics steps conducted on industrial data and implemented using our middleware.

A. Data Pre-Processing

The company has installed a database server for storing relevant data on all processes and manufactured products. As an initial data pre-processing step, we identified the relevant variables related to production forecasting and selected a set of dependent and independent variables. Table I lists the selected variables and their descriptions. We executed various queries to get data related to the selected variables. We used a query-based approach in order to ensure the flexibility of the system developed on top of the extracted data in such a way that any future versions of the database can be easily linked to our prediction tool.

After the execution of queries, our next steps were data preparation and data cleansing. During these steps, we prepared a matrix to contain the results of live queries executed over the database, and stored the results in this data matrix following a structure which can be easily used by machine learning algorithms. The data matrix contains data related to production operations and also data related to potential influential variables. We manually analyzed the extracted data to ensure that the prepared data has been properly cleaned and is free of any discrepancies, missing values or incorrect information. For the purposes of our analysis, we considered three independent variables, namely (i) Scrap: the number of units scrapped during production, (ii) Rework: the number of units sent back for reworking, and (iii) Lead time: the overall time it takes for a container to be processed between the first and last operations. The extracted data spanned a period over the last three years. We followed the 80%-20% approach for the training and testing phases. Once the model is trained, we use a validation set to check the accuracy of our trained model.

B. Regression-Based Approaches for Prediction

We applied different machine learning algorithms over the collected data to identify the best performing algorithms, depending on the nature of the data collected. We used regression-based models such as Multiple Linear Regression, Support Vector Regression, Decision Tree Regression and Random Forest Regression. We analyzed data related to each of the dependent variables in order to accurately predict the values for the independent variable, e.g. the number of units produced in this case ("Output"). All models were trained using training dataset (80%), while a validation dataset (20%) was used for testing.

Results of our experimental evaluation using the aforementioned four algorithms are presented in Figures 2(a), 2(b), 2(c), and 2(d) respectively. The results show a comparison between the actual (blue lines) and predicted (orange lines) values for the number of units manufactured during a period of six months.

In order to select the most appropriate algorithm for our use case, we used a Root Mean Square Error (RMSE) mechanism to calculate the accuracy of an algorithm. RMSE shows how close a trained model (or a regression line) is to a set of actual points. This is achieved by taking the distances from the points to the regression line (these distances are the "errors") and squaring them before taking the root for the final value. The smaller the RMSE, the closer the line is to being a best fit.

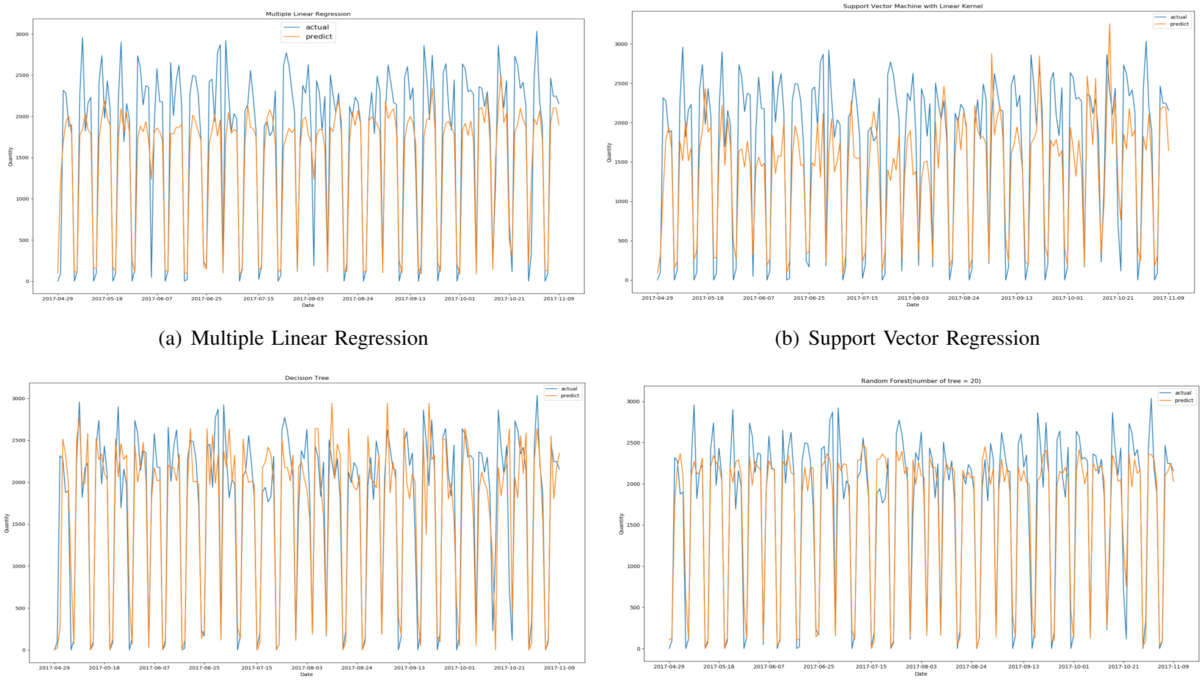
As we mentioned earlier, Scrap, Rework, Lead Time and Output are the variables we considered for our use case. In this case, Output is the dependent variable, while Scrap, Rework and Lead Time are independent variables. The model is trained using the independent variable data for a particular time period and validated by comparing the predicted output to the actual output for the same time period. Table II shows the results of the RMSE score for each of the four regression algorithms used. Based on the results, we selected the Random Forest algorithm.

Regression Types	RMSE
Multiple Linear	467.89
Support Vector	587.84
Decision Tree	434.54
Random Forest (n = 20)*	312.37

TABLE II
RMSE SCORES FOR DIFFERENT REGRESSION MODELS

C. Real-Time Event Monitoring and Event Notification

In our use case, there had been no prior data collected that was related to detected events and their causation. We set different thresholds and targets based on historical data analyses and domain knowledge collected from existing experienced staff. We developed a set of tools to monitor, detect and report events. We also assessed the impact of unexpected



(a) Multiple Linear Regression

(b) Support Vector Regression

(c) Decision Tree Regression

(d) Random Forest Regression

Fig. 2. Results of Predictions using Different Machine Learning Algorithms

events by comparing the average values with the live data after the unexpected event. We give a brief description of all steps below:

Target Definition & Threshold Setting. One of the goals of real-time analysis is to be able to alert users when a particular processing step deviates from a predefined target. Firstly, we need to define realistic targets and use these targets as a threshold for deviation detection. In a manufacturing unit, the production is usually defined in terms of the number of parts per minute (PPM), and this was the same for our industrial use case. We defined targets in terms of PPM for each type of product and process, however we leveraged the outcomes of historical analyses and use the predicted/estimated values to automatically define targets. In order to provide flexibility and accommodate any unexpected situations, we also provided an interface which allows shift supervisors to set goals for each shift and also to log any reasons if a target is increased or decreased from the automatically suggested target.

Event Detection & Event Logging. In our case study, the definition of an event is a situation that happens whenever a predefined threshold level is breached by a deviation. We used the following notations and definitions for event detection:

- *P*: is a process which is defined as a set of work-flow steps. Each *P* is assigned with a target *T*.
- $R = \{ r_1, r_2, ..., r_n \}$ is a set of reasons which 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 negative effect on *T*, which can be calculated based on the following formula:

f(R,T) = $\frac{\sum_{i=1}^{n} w_i f(r_i,T)}{\sum_{i=1}^{n} W_i} > 0$ (or < 0), where w_1, ..., w_n are the weights of the contributions of reasons r_1, ..., r_n respectively.

Given a target *T*, a set of reasons *R*, and two thresholds α, β ($\alpha < \beta$). An "Event" is detected whenever the value of *R* on *T* surpasses the predefined value of threshold. More precisely,

f(R,T) $< \alpha$ *and f(R,T)* $> \beta$

Alerting & Notification. We developed two types of notification methods for notification delivery. An alert system was integrated within the progress dashboard application. Supervisors were able to monitor the real-time progress of the production unit by following a visual interface installed at the shop floor. For managers, we provided an email-based notification delivery mechanism, and a system-generated email is forwarded to selected managers notifying them of any unexpected events or breaches of thresholds defined to monitor the productivity.

D. Capacity Planning Tool for Production Forecasting

We developed a capacity planning tool, which can be used by managers to set long-term targets and goals related to

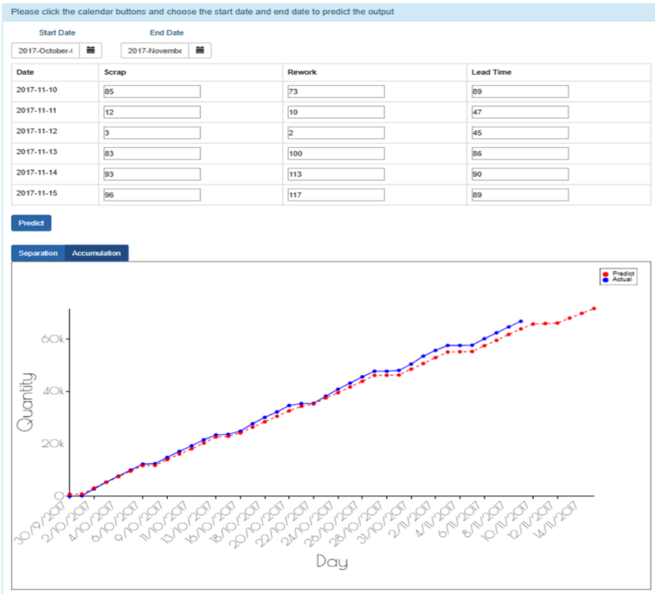


Fig. 3. An Interface for Accumulative Capacity Planning using Production Forecasting

their production. As shown in Figure 3, the tool provides results of production forecasting, where the blue line is the actual production while the red line is the prediction. Using this tool, the managers can adjust the values of different dependent variables to analyze their data following different *what-if* based assumptions. Historical data analyses were able to provide an estimated value for each of the days as auto-filled values, which can be changed by the user to see the impact of the change.

VI. Related Work

In this section, we describe existing technologies to build applications that enable real-time event detection and perform predictive analytics functionality. We divide these technologies into two broad categories: **cloud-based** and **on-premise** solutions.

Cloud-Based Manufacturing. Recently, we have seen the use of cloud services for building Industry 4.0 applications. Cloud-based manufacturing is a centralized single-shop place that allows manufacturers to apply industrial analytics on top of stored data. For example, Microsoft Azure[1] allows users to store structured and unstructured data at any scale though its "data lake" component. Azure's stream analytics component[2] is an event-processing engine that allows developers to ingest and transform high volumes of data streamed from IoT devices. Using these services, users can run different analytics – from simple analytics such as data visualization to complex analytics such as real-time analytics, machine learning, and big data processing. Siemens has launched MindSphere[3] (hosted on AWS), a cloud-based Industry 4.0 operating system, which lets manufacturers connect their industrial machines to the cloud and offers a marketplace (like an App Store) to use deployment-ready industrial applications. GE has developed Predix [4], an industrial Internet platform, which offers a marketplace to deploy various apps and services, including predictive maintenance, anomaly detection, algorithms for intelligent edge, and more.

Although cloud-based approaches reduce application development efforts and maintenance costs by keeping the application login at a centralized cloud service [8], [9], they may not be suitable for Industry 4.0 applications, primarily because of their high latency and high bandwidth requirement. They also assume that sufficient connectivity exists between IoT devices and cloud services, which may not hold true in reality due to various reasons such as noisy factory environments and factory setups in rural areas where the sufficient infrastructure for high-speed Internet may not be in place. Even if we assume that advanced technologies could address the bandwidth, latency and connectivity issues (for example, edge analytics solution such as AWS Greengrass[5]), there will always be regulations and security concerns around sharing data.

On-Premise Solutions. For this category, a common approach is to send sensor data over the network through proprietary protocol standards (e.g., Modbus) or emerging standards (e.g., MQTT, OPC-UA, BLE). The data is collected at a gateway device to perform common operations (e.g., aggregation, alerts and control). Furthermore, the collected data is sent to more powerful servers to be analyzed and to train the machine learning algorithms for better decision making. A set of tools[6] from the Eclipse Foundation are available to build such a system. We continue to leverage our existing tools and middleware to build Industry 4.0 applications, for example, IoTSuite [3], the Semantic Web-based tool SWoTSuite [10], and other real-time analytics middleware [1].

VII. Conclusion

In this paper, we presented middleware for real-time data analytics in combination with traditional historical data analysis. The middleware has been successfully deployed in a large manufacturing unit, and we can consider it as being a first step for the company to build towards their larger vision of full automation and Industry 4.0. In the future, we plan to extend this middleware deployment at all processes within the factory and design more business intelligence tools relying on real-time data analytics.

VIII. 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 (Insight

[1] http://bit.ly/azuremanu
[2] http://bit.ly/azurestre
[3] https://siemens.mindsphere.io/
[4] https://www.predix.io/catalog
[5] https://aws.amazon.com/greengrass/
[6] http://bit.ly/eclipseiot

SFI Research Centre for Data Analytics), co-funded by the European Regional Development Fund.

References

[1] M. I. Ali, N. Ono, M. Kaysar, Z. U. Shamszaman, T.-L. Pham, F. Gao, K. Griffin, and A. Mileo. 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, 2017.

[2] A. Berson and S. J. Smith. *Data warehousing, data mining, and OLAP*. McGraw-Hill, Inc., 1997.

[3] S. Chauhan, P. Patel, A. Sureka, F. C. Delicato, and S. Chaudhary. 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, pages 37:1–37:2, Piscataway, NJ, USA, 2016. IEEE Press.

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

[5] C. P. Gatica, M. Koester, T. Gaukstern, E. Berlin, and M. Meyer. An industrial analytics approach to predictive maintenance for machinery applications. In *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4, Sept 2016.

[6] M. Intizar, P. Patel, S. K. Datta, and A. Gyrard. Multi-layer cross domain reasoning over distributed autonomous iot applications. *Open Journal of Internet Of Things (OJIOT)*, 3, 2017.

[7] J. Lee, H.-A. Kao, and S. Yang. Service innovation and smart analytics for industry 4.0 and big data environment. *Procedia Cirp*, 16:3–8, 2014.

[8] P. Patel, M. I. Ali, and A. Sheth. On using the intelligent edge for iot analytics. *IEEE Intelligent Systems*, 32(5):64–69, September 2017.

[9] P. Patel, M. I. Ali, and A. Sheth. From raw data to smart manufacturing: Ai and semantic web of things for industry 4.0. *IEEE Intelligent Systems*, 33(4):79–86, 2018.

[10] P. Patel, A. Gyrard, S. K. Datta, and M. I. Ali. Swotsuite: A toolkit for prototyping end-to-end semantic web of things applications. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, pages 263–267, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee.

[11] S. Wang, J. Wan, D. Zhang, D. Li, and C. Zhang. 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, 2016.

[12] H. J. Watson and B. H. Wixom. The current state of business intelligence. *Computer*, 40(9), 2007.

[13] R. Y. Zhong, C. Xu, C. Chen, and G. Q. Huang. Big data analytics for physical internet-based intelligent manufacturing shop floors. *International journal of production research*, 55(9):2610–2621, 2017.