Towards Developing a Digital Twin for a Manufacturing Pilot Line: An Industrial Case Study

Fatemeh Kakavandi, Cláudio Gomes, Roger de Reus, Jeppe Badstue, Jakob Langdal Jensen, Peter Gorm Larsen, and Alexandros Iosifidis

1 Introduction

Increasing manufacturing customization, reducing the manufacturing cost and CO_2 emissions, faster system verification and validation are goals that have been sought in industry. Smart manufacturing and Industry 4.0, assisted by different technologies, are introduced to make these objectives possible. Digital twin (DT), the virtual counterpart of a physical entity, is one of the current concepts of Industry 4.0 that has gained attention both in industry and academia [1]. A DT can simultaneously represent, monitor, optimize, and control the physical replica [2].

F. Kakavandi (✉) · C. Gomes · P. G. Larsen · A. Iosifidis
Department of Electrical and Computer Engineering, Aarhus University, Aarhus, Denmark
e-mail: Fatame.kakavandi@ece.au.dk

C. Gomes
e-mail: claudio.gomes@ece.au.dk

P. G. Larsen
e-mail: pgl@ece.au.dk

A. Iosifidis
e-mail: ai@ece.au.dk

R. de Reus
Device Manufacturing Development, Novo Nordisk A/S, Hillerød, Denmark
e-mail: rder@novonordisk.com

J. Badstue
CIM Industrial Systems A/S, Aarhus, Denmark
e-mail: jba@cim.as

J. L. Jensen
Alexandra Institute, Aarhus, Denmark
e-mail: jakob.langdal@alexandra.dk

E. Karaarslan et al. (eds.), *Digital Twin Driven Intelligent Systems and Emerging Metaverse*, https://doi.org/10.1007/978-981-99-0252-1_2

1.1 Definitions of Digital Twins

Different academics and industrial practitioners have defined DTs over the past years, but most of these definitions touch upon the same concepts [3, 4]. One of the first definitions of the DT comes from NASA and the US Air Force Research Laboratory, which referred to DT as "an integrated multi-physics, multi-scale, probabilistic simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, and so forth, to mirror the life of its flying twin" [5]. However, nowadays, DT is defined by a broader definition as a virtual replica that continuously updates and represents products, assets, personnel, or processes and adapts synchronously to depict changes in geometric characteristics, resource states, or working conditions [6].

1.2 Elements

Different vital elements of DTs are referred to as dimensions, and according to the literature, DTs can include three or five dimensions [3]. The basic three-dimension model of DT points out the physical entity, virtual counterpart, and the connection as the main DT elements. Later, a five-dimensional definition of DT was proposed by Tao et al. [7], which introduced the services that DT provides, and the data transferred between two entities as vital elements of DT.

1.3 Enabling Technology

Five-dimension DT emphasizes the need for a high-fidelity model with a comprehensive perception of the environment. For this purpose, different sensing technologies are needed to make the virtual model aware of the physical entity status. Furthermore, different big data analytics methods are needed due to the large volume of data. Moreover, for transmitting data between the twins, different communication protocols are needed [8].

1.4 Case Study

This case study focuses on developing a DT for an industrial manufacturing prototype with multiple steps. The adjustments and assembly of several components and subassemblies are performed at speeds suitable for high-volume production. Therefore, the data collection is implemented with a mature method, and all equipment and process signals from various sensors are organized as a database. Furthermore, the details related to the assembly machine, the product, and the process of interest are explained in Sect. 3.

Standard programmable logic controller (PLC) data collection mechanisms are typically limited by a sampling rate that is considerably lower than the scan cycle at which the PLCs operate, e.g., 100 ms intervals versus 5 μs scan cycles. For the data collection to keep up with the scan cycle speed, we employ a Kafka-based ingestion solution where data are buffered and transmitted efficiently from the PLC to a Kafka broker running on standard PC hardware [9]. Using Kafka allows us to consume the PLC data, unpack and interpret them so that the data can be re-published back into Kafka, and become ready for further analysis. By delegating the data processing to standard PC hardware, we can currently process approximately 500,000 datapoints/s. Moreover, data collection and ingestion architecture of this case study is described in Sect. 4.

Furthermore, to gain access to the data in a structured manner, we utilized the CATCH.AI proprietary system. The system gives the ability to store, visualize, and act on the data in an easy-to-use interface, so the focus can stay on developing the analysis tools. Furthermore, the CATCH.AI system is used to set up a trigger for the data analytics model that is executed when a particular manufacturing process step is concluded. In Sect. 5, the details of the CATCH.AI tool are clarified.

The data analytics model deploys a machine learning method to extract the summarized knowledge from the database. In other words, to evaluate the quality of the products, an anomaly detection model is developed for one step of the assembly process. To train the anomaly detection model, some products in normal status are produced. However, to validate the detector, some fault injection experiments also need to be conducted. The quality and sample numbers also play an essential role in the performance of the model; therefore, some synthetic data are produced with a data augmentation method. Furthermore, Sect. 6 describes the experiments and machine learning methods that have been used in the case study.

This chapter describes our case study and introduces different definitions and elements. We will discuss about the novelty of this work, the physical system, the volume of the data, and the data analysis and processing tools. Finally, we discuss the challenges in enabling a DT of such a large case study, along with lessons learned and future work.

The remainder of the chapter is organized as follows. In Sect. 2, related work in DT for manufacturing is discussed. Section 3 introduces the physical system and the process of interest. Furthermore, the details about the data collection and storage tools that have been used in this case study are provided in Sect. 4. Moreover, Sect. 5 describes the dashboard and visualization tool, CATCH.AI and its connection to the data management section and the data analytics tool. Afterward, Sect. 6 presents the anomaly detection model and different assembly experiments to produce normal and abnormal products. Finally, Sect. 7 summarizes the chapter and discusses the challenges of constructing a DT for a manufacturing pilot line.

2 Related Work

In recent years, people in academia and industry have invested in designing DTs for various physical entities, which has led to different simulation tools employed for such physical machines. Data storage and transmission methods for further analysis have been explored concerning data with diverse volumes or content in various applications. Furthermore, DTs with diverse services have been developed for physical devices with different objectives. For displaying the results of the DT, various visualization tools and dashboards have been designed. This section introduces some of the DT enablers, tools, and techniques briefly.

2.1 Physical Systems

Designing DT has gained attention in different industries; therefore, various physical entities have been studied for developing DT. In [10], the authors defined an architecture for developing DTs and examined their proposed method in a case study which includes a refinery automation system with four valves. Moreover, some physical systems like grinding wheel [11], 3D printer [12], welding production line [13], rotating machinery [14], machine tools [15], and robots [16] have been invested for designing DT. In [17], the authors proposed an incubator system that is complex enough to highlight the need for DTs while at the same time being simple enough to be built from widely available tools. Later, in [18], the authors described the implementation of a DT for the incubator system. The resulting architecture has been generalized in [19] based on the comparison with another DT for a race car test bench.

2.2 Data Management Technologies

Data storage and transmission are data-related functionalities for expressing the physical entity and connecting it to its digital replica. Therefore, various methods have been proposed and applied for these purposes.

2.2.1 Data Storage

There are various frameworks for big data storage, for example, MySQL and HBase. In the MySQL framework, data are stored as tables with records of data in rows and the data description in columns [20]. Furthermore, for storing the data related to the machine tool in [15], the PostgreSQL database is employed. This database is an open-source database running on a local PC through the Internet using a script in Python.

2.2.2 Data Transmission

Different data transmission protocols have been introduced to efficiently and securely connect the two digital and virtual replicas. Lu et al. in [2] describe how industrial communication protocols have evolved from the fieldbus legacy communication method to the second generation, Ethernet-based protocols and the current category, wireless network technologies. All these improvements are implemented to satisfy the real-time and reliability requirements of industrial processes. On the other hand, transmission mechanisms can be divided into two categories wire-based or wireless methods. Some of the wire-based tools are twisted pair coaxial cable and optical fiber; the wireless methods are Zig-Bee, Bluetooth, Wi-Fi, ultra-wideband (UWB), and near-field communication (NFC) [10].

2.3 Digital Twin Services

Depending on the physical system, DTs have diverse objectives; for instance, in the aerospace domain, a DT should be able to predict the life cycle of an aircraft. However, the DT machining application enables real-time quality inspection of machining results [21]. Furthermore, one of the important yet less achievable goals of a DT is process optimization and control that is implemented for a machine tool to stabilize machining parameters for achieving optimum surface roughness in [22]. On the other hand, fault diagnosis is one of the DT services that has been implemented for different physical systems. For example, the authors in [14] developed a pilot digital twin prototype of a rotor system that effectively diagnoses rotor unbalance fault and predicts its progression.

3 Physical Pair

This section describes the physical system in this case study briefly. Moreover, different machine and the device components are shown and explained to better understand the process. The process of interest is briefly introduced since the anomaly detection model is designed for this process.

3.1 Pilot Line

The physical pair consists of a test bench for medical device assembly by Stevanato Group/SVM. As shown in Fig. 1, this small assembly line contains a base and top frame (1 and 2). The transport system (3) is a modular XTS linear motion platform by Beckhoff automation. The linear motors (5 and 6, Linmot PR02-52) provide both vertical and rotary motion and are equipped with force and torque transducers for continuous process monitoring with sampling frequencies up to 20 kHz. A sensor system (4) for single-point data is used to verify the assembly process. The most common way of doing so is by measuring the total height or other geometric dimensions of the assembly.

A closer look of the system is provided in Fig. 2. The mover, or pallet, is mounted on the transport system. It contains the sub-assemblies and component to be assembled and can move to an arbitrary position along its axis of movement.

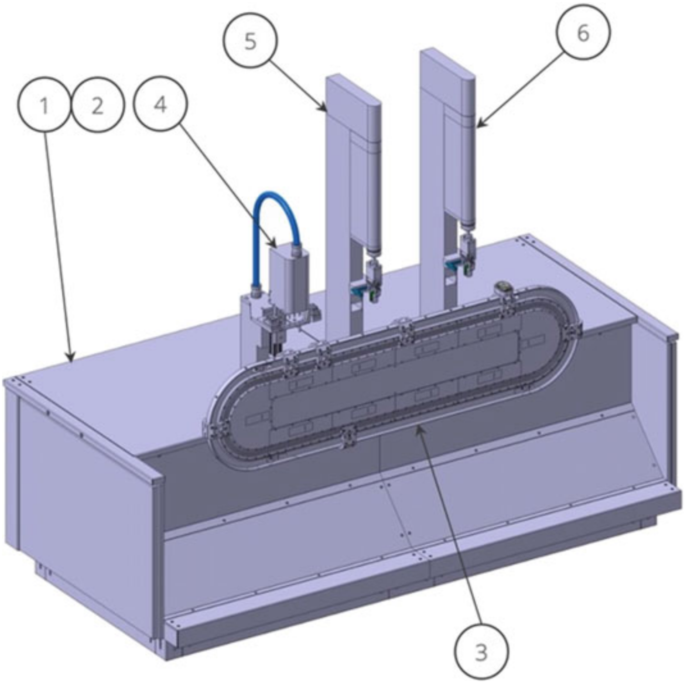

Fig. 1 Schematics of the physical pair while the actual machine is shown in Fig. 2. The assembly equipment consists of (1) base frame, (2) top frame (not shown in figure), (3) transport system, (4) sensor system, (5) and (6) linear motors. The linear motors perform the assembly steps, and the sensor system verifies the assembly

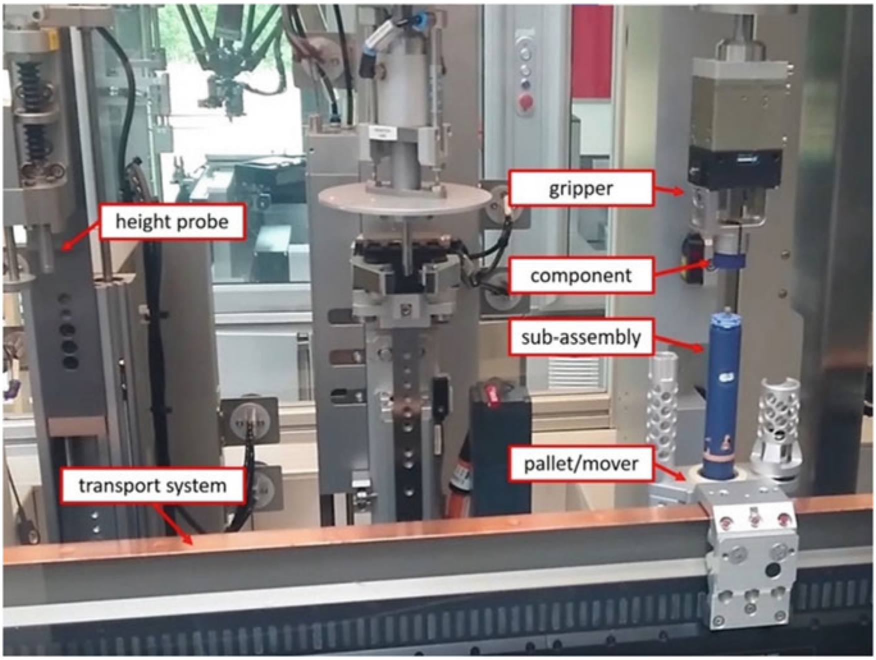

Fig. 2 A more detailed look at the equipment from Fig. 1. The transport system moves a pallet, also called mover, which contains components to be assembled. In this figure, a gripper, mounted on linear motor (6), holds a component which must be mounted to a subassembly for the fabrication of a device. Correct assembly can typically be verified by using a height probe

3.2 Process of Interest

The test case in this study is the assembly of a medical device. Its constituent components and assembly process steps are shown in Fig. 3. First, two modules, also called subassemblies, are mounted together (as shown in Fig. 3a). This requires linear movement in the direction of the arrow with prior alignment and rotational orientation along the long axis of the modules. Next, the modules are joined using a snap-fit consisting of two main snaps and two minor snaps to stabilize the assembly. A certain axial force is required for successful operation. Second, a similar process with a segmented ring snap is performed with a single component mounted onto the new subassembly (shown in Fig. 3b) to form the complete device shown in Fig. 3c.

A snap joint between two plastic components is illustrated in Fig. 4. Upon mounting the green component (moving right to left) onto the blue one, the entire snap structure is bent inward until the "hook" snaps into place and establishes the joint. The axial force mentioned above is needed to bend the snap structure and overcome friction during the assembly process.

According to the process sequence in Fig. 3, the mover is positioned, and the one subassembly is picked up by the gripper, which then moves upward. The mover is

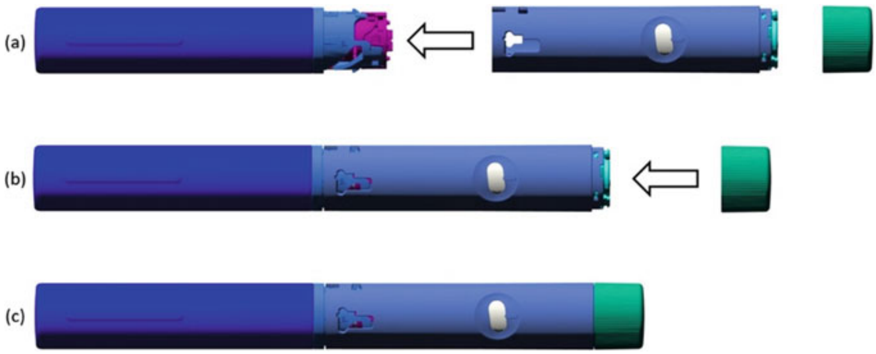

Fig. 3 Assembly of a medical device consisting of two modules (subassemblies) and one component (**a**). The two modules are assembled in (**b**), and the device is completed when the final component is mounted (**c**)

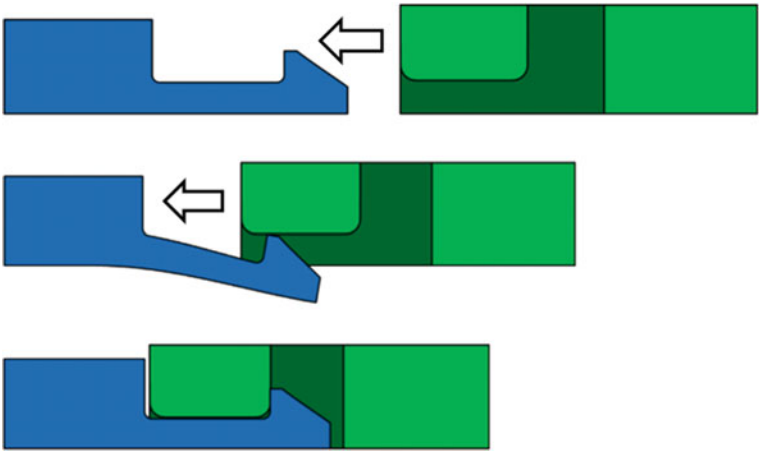

Fig. 4 Snap joint between two plastic components (image by Christoph Roser at AllAboutLean.com under the free CC-BY-SA 4.0 license)

then repositioned, and the two subassemblies are mounted by a controlled movement of the linear motor. This sequence is then repeated for the second step in which the single component is mounted to complete device assembly.

3.3 *Process Assessment*

The entire sequence is shown in Fig. 3. Both for the assembly process itself, as well as the axial acceleration and deceleration of the motor with the gripper, forces are required. The process is typically verified by a height measurement, which ensures that all components are assembled in the correct position. However, geometrical tolerances may cause the verification to be insufficient, and additional action is needed to ensure the snap quality is perfect.

One of the solutions that can address this problem is to correlate the product quality and the process behavior. In other words, by looking into the force and displacement curves that represent the process behavior, we can evaluate the quality of the products. For this purpose, a data transfer layer is needed to collect the data and store them on a hard drive for data access and visualization.

4 Data Management

In this section, the data collection and storage methods are briefly described. First, the data transfer layer Apache Kafka, a distributed event store and stream-processing platform applied for data collection, is explained. The data transfer layer runs on a local computer close to the physical system. Then, data are temporarily stored in Kafka, and later, it can be transferred to a cloud-based solution.

4.1 Data Collection

Extracting data from PLCs in real time and with a resolution sufficient to capture all state changes requires a very efficient data collection mechanism. In order to collect and harmonize the data coming from the different PLCs of the manufacturing line, an Apache Kafka-based data collection platform has been implemented.

Kafka is chosen as a transport layer in the solution due to its unique capabilities for high throughput and strict ordering guarantees. Furthermore, Kafka is well supported and integrated into standard data engineering and analytics tools and therefore provides a bridge between the industrial engineering traditions and modern data science development methodologies.

The platform consists of a number of input adapters (Kafka Connect sources) that consume a continuous stream of binary data from the PLCs that are optimized for transmission efficiency and publish them into a separate Kafka topic for each PLC. The binary data are then consumed and converted into a structured format and published back into new topics ready for further consumption by data analytics tools as shown in Fig. 5.

In our setup, the input adapters support two commonly used protocols, MQTT and plain TCP. The data format transmitted through these protocols is typically not standardized, and therefore, the interpretation/conversion into a structured format is tightly coupled to this format.

Many solutions for consuming process data generated by PLCs rely on sampling-based technologies like OPC-UA where the current state of the PLC is sampled at

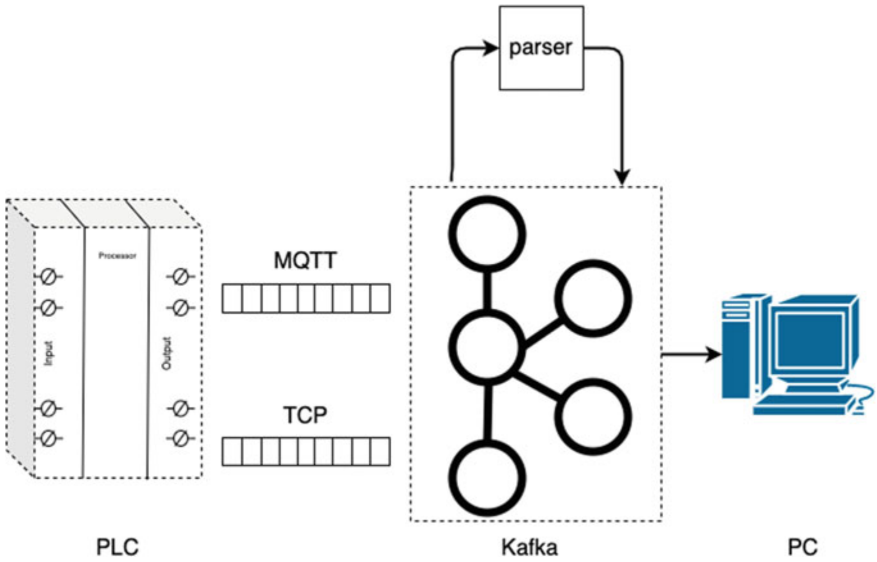

Fig. 5 Data collection structure with Kafka

regular intervals, typically every 100 ms. This is often sufficient to capture high-level process steps, but not always enough to fully create a model of the physical system. Therefore, the PLC code has been augmented with a block that during each scan cycle transmits all changed data values to the Kafka input adapters.

4.2 Data Storage

Kafka supports the concept of automatic retention management. That is, each topic containing data in Kafka can be configured with a retention parameter telling Kafka for how long to store data in the topic. This is used to allow Kafka to take on the role of a data buffer. Data are typically stored in Kafka for hours or a few days in which time-relevant data are retransmitted to more permanent data storage, e.g., an SQL database or a cloud-based system.

5 Dashboard CATCH.AI

In this section, CATCH.AI, the dashboard and visualization tool, is briefly introduced. CATCH.AI is a flexible software solution made for working with significant amounts of data. The data model structure, dashboard, and some CATCH.AI capabilities are explained in the following. It is outside the scope of this chapter to give a full overview of CATCH.AI features. We refer the reader to [23] for more details. In the following, we discuss the features of CATCH.AI used in connection to the case study.

5.1 Catch.AI Overview

Catch.AI is a tool that allows the creation and configuration of dashboards, visualizations, data collection, and feedback loops that can reconfigure the physical system. Figure 6 shows an example dashboard, where the time series related to a process and different diagrams related to it are displayed. However, CATCH.AI cannot use data analysis tools internally; it can activate external sources for data analysis purposes.

CATCH.AI is made as a flexible solution that can cater to a wide range of different inputs; in this case study, we will focus on the input from the Kafka system. The raw data are transferred from the Kafka database to the CATCH.AI data model. It provides an easy-to-use Web interface that makes the data accessible event based. Furthermore, the report section in CATCH.AI makes it possible to access historic data for data mining purposes. In addition, CATCH.AI provides dashboard functionality with both historical and live data.

5.2 CATCH.AI Connection to Kafka

CATCH.AI has a REST API with OPENAPI3 description available. This interface is used to connect Kafka and CATCH.AI. A custom mapper application is made for the project to take care of the Kafka consume interface and map it into the domain model of CATCH.AI. This mapper also ensures that we have more clean data to work with, since NULL characters can occur in the raw data streams. Therefore, the data are cleaned up before being made available in CATCH.AI for easier usage by the end-user.

5.3 CATCH.AI Data Structure

As mentioned above, CATCH.AI organizes data in an event-based manner to make it easier to access data related to different steps of product assembly. Therefore, labeling the data with three keywords: **device**, **event**, and **property,** enables an easy access method to reach the data with specific characteristics.

The different equipment in the physical system or the DT correspond to the keyword **device** in our case study; for example, the linear motor, shown in Fig. 1 as number (6), in the assembly line is addressed as one device in CATCH.AI.

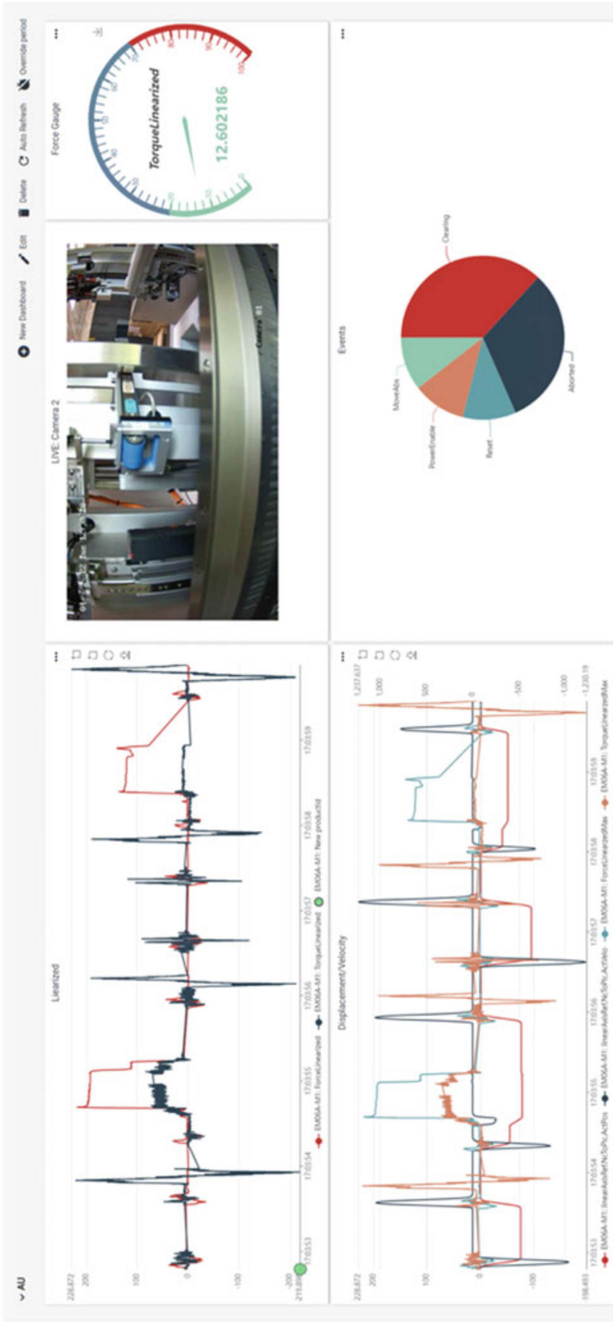

Fig. 6 CATCH.AI dashboard. The plots in the left show EM06-M1 (device) properties (force, torque, and displacement) related to one product assembly, while the camera shows the live video of the assembly process. The diagram in the right bottom corner shows different events in the product assembly

Furthermore, the **property** of a device is defined as measurement values or states related to the device; for example, the force sensor in the linear motor is defined as a property.

Moreover, different state changes in properties of the machine form **event**. For example, the stop and start points in producing a product can be defined as events (see Fig. 6). However, different events can be described based on different properties.

In this case study, the process of interest (as described in Sect. 3.2) is defined as an event base on the displacement as the property and the linear motor as the device. Therefore, it is easier and significantly faster to access the data related to the process of interest.

5.4 CATCH.AI Data Access

The event-based data structure lets the end-user access the specific data in the timeline of the assembly process. On the other hand, if the end-user wants to access the data related to one unit in production, they can select two events, "Start Process" and "End Process."

Therefore, the advantage of describing the data structure based on events is that it makes it possible to search a significant volume of data from weeks of production, including billions of data points in a short period. For example, Fig. 7 shows data related to a unit produced in the assembly process. Different properties of Device 1 regarding starting and ending events are visible there.

5.5 CATCH.AI for External Services

Instead of polling for data in CATCH.AI on an interval, the system also contains the "RuleEngine," allowing us to work directly with the stream of data. For example, the RuleEngine can be set up to trigger an external process when the product is finished. In this case, later additions to the system, such as machine learning algorithms, reporting, and other visualization tools, would not necessarily have to parse everything in real time, but only act when required, that is, when a specific pattern is found within the data stream.

In this case study, the data related to each product assembly process can be transferred to the anomaly detection model via CATCH.AI by triggering the corresponding Python code. Figure 8 shows the rule management system in CATCH.AI where the anomaly detection model is triggered when the product assembly is finished.

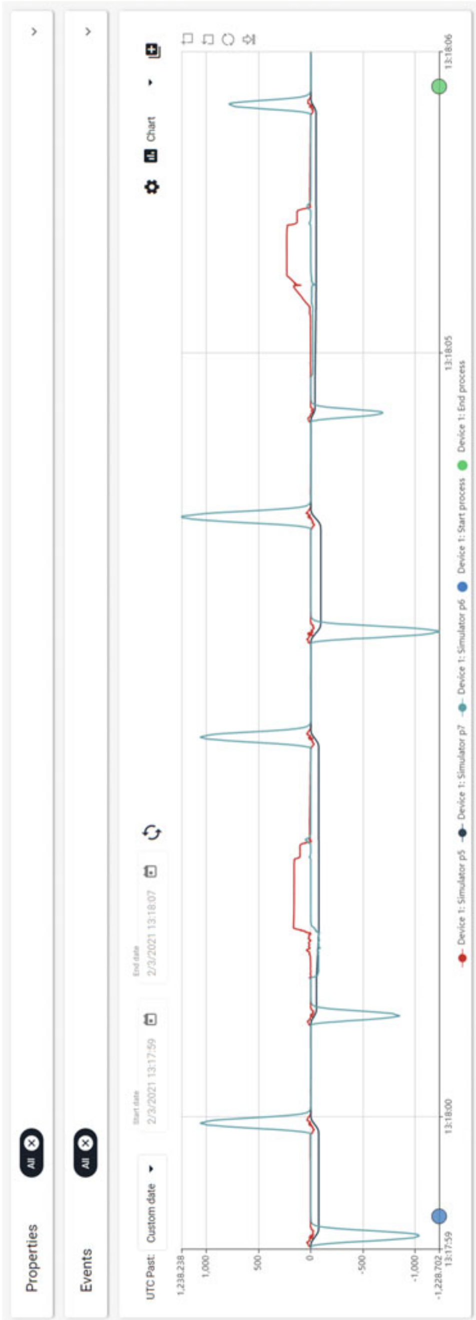

Fig. 7 Access history data in CATCH.AI showing Device1 all properties and events

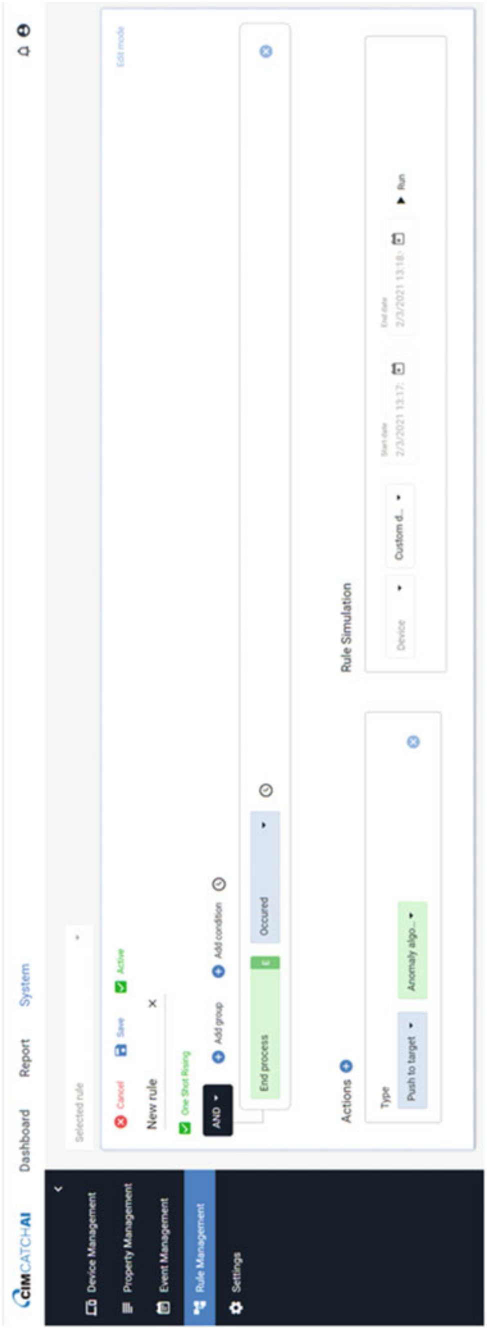

Fig. 8 Rule management system in CATCH.AI

6 Anomaly Detection Model

In this section, the anomaly detection model is described. This includes the machine learning model, the dataset, and the experiment settings used to train and test the anomaly detection model.

6.1 Product Quality Assessment

Evaluating the quality of products in an assembly process is critical. In this case study, we evaluate the products by analyzing the behavior of the assembly process, which is equal to looking into various signals from different parts of the process and using this to predict the product quality. Anomaly detection also is a process monitoring tool for early warnings before the product quality is compromised.

This case study is a proof of concept. It is essential to evaluate the process quality based on the recorded signals. Figure 9 shows the force and displacement recorded over the time from one of the equipment in the pilot line. The highlighted area in Fig. 9 shows the force and displacement curves in the process. The force profile is the measurement we investigate since it records the force applied to assemble the components and the reaction.

The aim of analyzing the force profile is to find the anomaly in the products via signals recorded while producing the product. For example, Fig. 10 shows that the abnormality can be visible with the force signal by plotting the force curve related to a normal and abnormal product in the same image. The blue signal was recorded

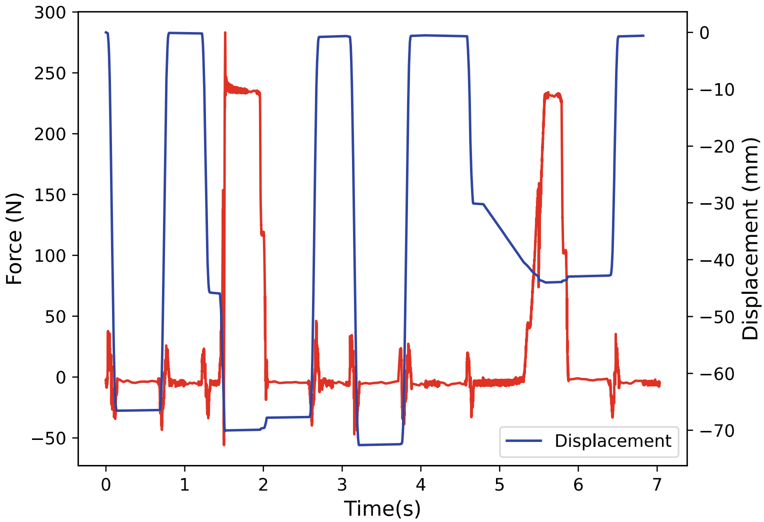

Fig. 9 Force and displacement of the process

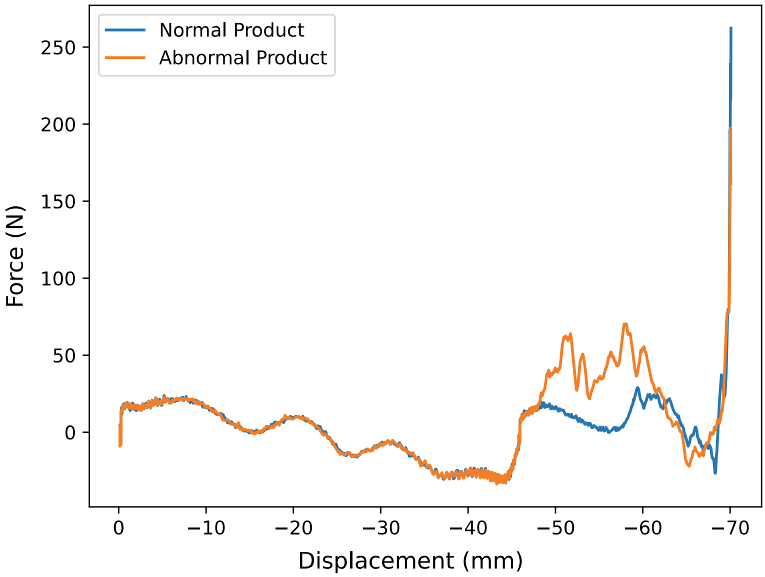

Fig. 10 Force–displacement for normal and abnormal products

when a normal product was assembled; while the orange one displays an abnormal product that deviates from the normal status.

An anomaly detection model should be developed to detect the anomalous cases automatically based on the deviation from the normal situation. Since anomalies can rarely happen, mining the signals related to normal products is valuable. We want to identify the outer bounds of the normal date such that data outside this limit can be classified as abnormal concerning what is considered normal data.

6.2 Detection Model

The detection model consists of a one-class support vector machine (OCSVM) [24] to find the boundary enclosing the normal data and random guided warping to generate augmented data. The augmented data can help generalize the detection model and improve performance. Below, the theory related to these methods is described briefly.

6.2.1 One-Class Support Vector Machine

To detect the abnormality in the force curve, the OCSVM model is applied. This method is trained by using only normal data, which is mapped to another feature space through a nonlinear kernel function. In that feature space, the objective is to define

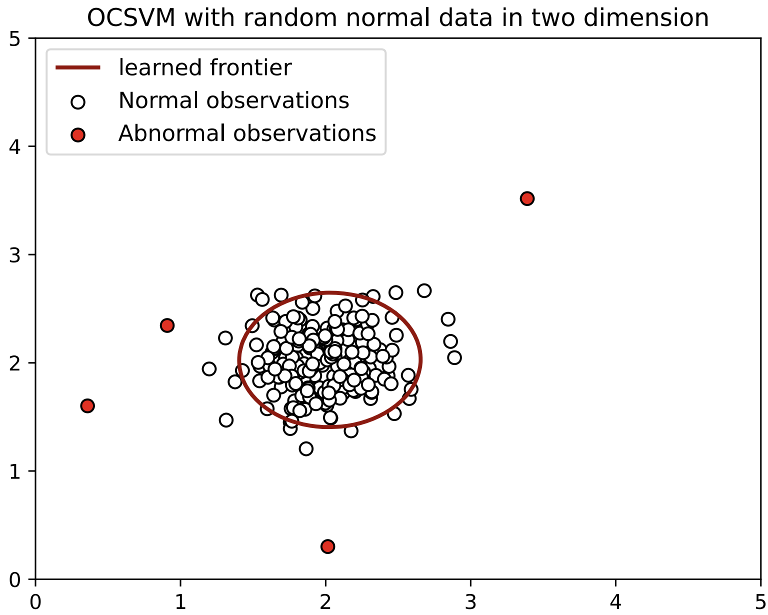

Fig. 11 Finding the boundary around the normal data with OCSVM for a random dataset in two dimensions

the hyperplane that best discriminates this data from the origin. This hyperplane in the feature space then corresponds to a complex shape in the original space enclosing the normal data.

For example, as shown in Fig. 11, the white and red dots show normal and abnormal random data points, respectively, in two-dimensional space. With the help of the radial basis function (RBF) as the nonlinear kernel, the boundary around the normal data can be calculated in a way that does not include the abnormal data (for more information about the theory, see [24]).

With adjusting the boundary, the accuracy of the model can be increased, or it can worsen the performance of the classifier, as shown in Fig. 11. Therefore, introducing a penalty term weighted by ν in the optimization formula as the regularization factor can express the trade-off between model complexity and training error; ν controls the number of training samples excluded by the decision boundary.

In Fig. 12, the normal training samples (white dots) are used to train the OCSVM. The abnormal data (red dots) and normal test samples (green dots) are used to evaluate the performance of the model. Figure 12 shows that increasing ν tightens the boundary and does not allow anomalous samples. On the other hand, a smaller ν will introduce some uncertainty in training dataset. Therefore, tuning the hyperparameter ν is an essential factor in training OCSVM to gain the suitable performance at test time.

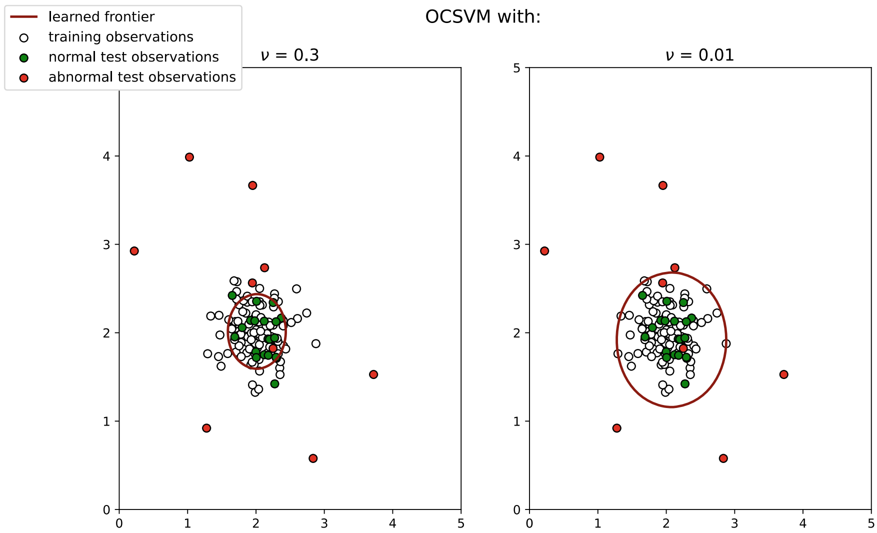

Fig. 12 OCSVM performance with different ν values

6.2.2 Data Augmentation Method

Random guided warping is applied to generate new normal samples to address data shortage [25]. The backbone of this method is a similarity search technique called dynamic time warping (DTW); this augmentation algorithm can generate new data similar to the real dataset.

Dynamic Time Warping

DTW is a classic method for finding the optimized distance between two time series, and it is robust to temporal distortion. Consider two time series $\mathbf{r} = r_1, \ldots, r_i, \ldots, r_I$ and $\mathbf{s} = s_1, \ldots, s_j, \ldots, s_J$ with sequence lengths $\boldsymbol{I}$ and $\boldsymbol{J}$, respectively, shown in Fig. 13. We consider $\boldsymbol{r}$ and $\boldsymbol{s}$ to be univariate time series. To find the global distance, DTW finds the minimal path on the element-wise cost matrix C (the Euclidean distance) using dynamic programming (for more information about the theory, see [26]). Furthermore, the two sequences are stacked in Fig. 13 with a bias to see the warping path.

This minimal path is referred to as the *warping path,* and the warping path is a mapping from the time steps of one series to the other. For example, the gray lines in Fig. 13 show the warping path. However, instead of using a one-to-one mapping from two time-axes in series, similar patterns are connected to find the optimum distance between $\mathbf{r}$ and $\mathbf{s}$.

Random Guided Warping

Random guided warping uses DTW and a reference pattern to generate synthetic patterns. In this case, instead of randomly warping the data sample and hoping it is

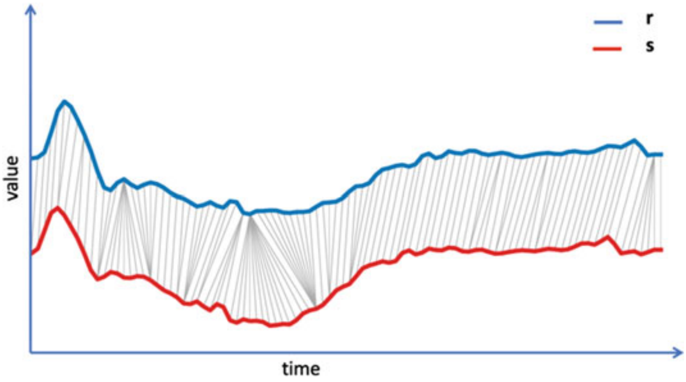

Fig. 13 Alignment with DTW between $\boldsymbol{r}$ and $\boldsymbol{s}$ [8]

realistic, a teacher is used to instruct warping in time domain. For example, assume $\boldsymbol{S} = \{s_1, s_2, \ldots, s_N\}$ is the training set, an augmented dataset will be generated called $\boldsymbol{S}'$ such that the accuracy of anomaly detection model trained on $\boldsymbol{S} \cup \boldsymbol{S}'$ is better than $\boldsymbol{S}$ alone. One of the advantages of using a reference for warping is that both the local patterns exist in the original dataset.

For generating an augmented sample $\mathbf{s}'$, a *random* sample $\mathbf{r}$ is chosen from the training set $\boldsymbol{S}$. Furthermore, the warping path between two data samples $\mathbf{s}$ and $\mathbf{r}$ can be calculated by DTW. Moreover, we can exploit the warping path to align the elements of the two time series. By aligning the elements in this way, sections of $\mathbf{s}$ are warped in the time domain to fit $\mathbf{r}$ as shown in Fig. 14.

The result is a sequence $\mathbf{s}'$ that has the feature values of $\boldsymbol{s}$, but the time steps of $\mathbf{r}$ under the warping path constraints provided by DTW. Finally, the process is repeated by selecting any two random patterns in $\boldsymbol{S}$. It is possible to synthesize N^2 number of time series where N is the number of patterns in each class $\boldsymbol{S}$. Finally, the augmented training set $\boldsymbol{S'}$ and the real training set $\boldsymbol{S}$ train the anomaly detection model, as shown in Fig. 14.

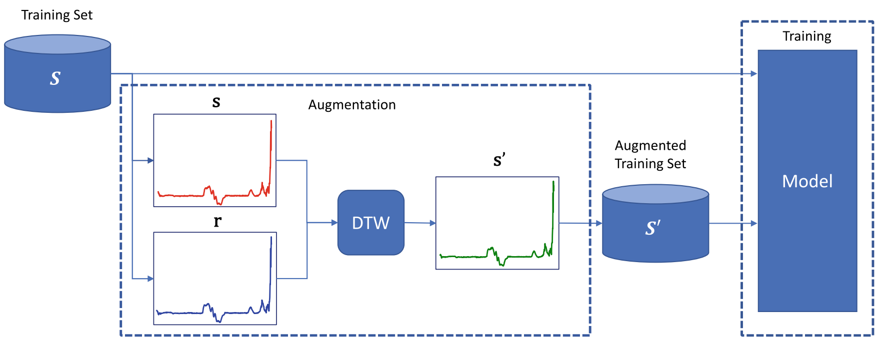

Fig. 14 Random guided warping for generating $\boldsymbol{S'}$ (inspired by [25])

Table 1 Different types of faults that have been applied

Type	Value	Product ID
Type 1	175°	8
	170°	9
	185°	10
	190°	11
Type 2	4	19
	2	20
	2	21
	3	22
	All	23

6.3 Experiments

Some experiments have been operated to collect data related to abnormal products, which is described in the following. Furthermore, a normal dataset is introduced in this subsection for training the model and is collected by running the physical system in the regular setting. For tuning the hyperparameter ν, the normal and abnormal data are split into train, validation, and test sets, which is described in detail in this subsection.

6.3.1 Fault Injection

Some data samples with abnormal labels are needed to test the anomaly detection model. However, the process is robust, and the probability of an anomaly happening is low. Therefore, many products need to be assembled before an anomaly occurs. However, producing many products with the pilot line is costly. Therefore, we had to introduce some fault into the process to test any future anomaly detection method. For this purpose, two types of faults have been applied.

- **First Type**: With this type of fault, the setting related to the gripper position is changed (see Fig. 2). The default setting is 180°, but we gradually changed the default setting with ±5° and ±10° according to Table 1.
- **Second Type**: With this type of fault, we remove some of the deformation structures of a plastic component in the product. Different number of components have been removed, as Table 1 shows.

6.3.2 Dataset

For training and testing the anomaly detection model, data samples with labels *normal*, *abnormal* are needed. Therefore, 102 devices were produced in normal situations, and force measurements related to these products were collected. These

Table 2 Dataset description with normal and abnormal labels

Data type	Number of samples	Time series length
Normal	102	11,239
Abnormal	9	11,299

force samples are referred to as normal data through this report. Furthermore, nine abnormal devices were assembled to test the anomaly detection model (see Table 1). Therefore, the data related to these abnormal devices are addressed as abnormal data.

Table 2 shows the normal and abnormal dataset and the time series length. The size of normal and abnormal dataset is one of the challenges in this case study; since producing the products is expensive, we relied on a small database. However, to overcome data shortage, we use a data augmentation method (see Sect. 2).

6.3.3 Hyperparameter Selection

For finding the boundary around the normal dataset, the OCSVM model is applied according to Sect. 6.2.1. In this model, the RBF is the kernel for mapping the raw data to a new feature space; therefore, tuning the hyperparameters ν in OCSVM is a vital task.

For tuning ν, we used a five-fold cross-validation method that can consider the anomaly data in the validation time. Values in $[0.001, 0.01]$ have been investigated with a step length of 0.001 to find the best value for ν.

For conducting the cross-validation process, the normal data are split into train and test sets with the ratio of (80%, 20%). In each iteration of cross-validation, we split the training set into five subsets which four subsets are used for training the model while the remaining subset is used for validation.

Furthermore, Fig. 15 explains how the five-fold cross-validation process is applied. The training set formed by only normal data is split into five subsets. For each candidate value of ν, we apply five experiments as follows. On each experiment, we use four (out of the five) subsets to train the model. The remaining subset is merged with the abnormal data (four samples) to form the validation set. We train the OCSVM with the training set (formed only by normal data), and we evaluate its performance on the validation set (formed by both normal and abnormal data). After running the five experiments, we calculate the average performance. Then, the average performances corresponding to different values of ν are sorted, and we select the value of ν corresponding to the best performance.

As shown in Table 3 for different values of ν, the performance of the model differs. In this project, the F1-score ($2\frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$) is the metric used for choosing the best model; the OCSVM model with a higher F1-score is selected. For example, among (0.002, 0.001) with the highest F1-score, we set up ν to 0.002.

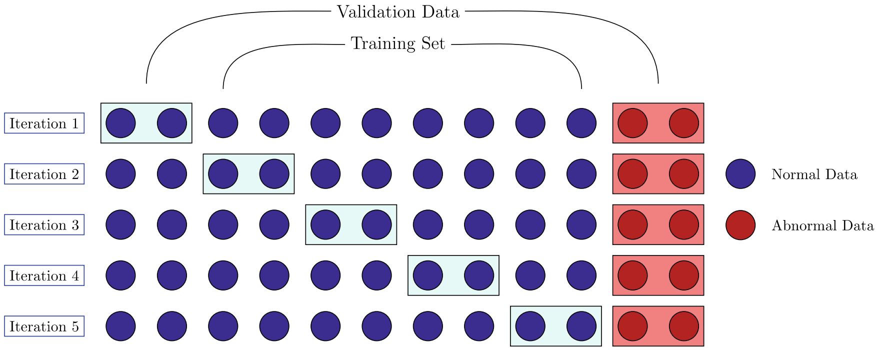

Fig. 15 Five-fold cross-validation. normal validation

Table 3 Results of the cross-validation for finding the hyperparameter ν

ν	F1-score
0.01	0.8975
0.009	0.8975
0.008	0.8975
0.007	0.8975
0.006	0.8975
0.005	0.8975
0.004	0.9006
0.003	0.9006
0.002	0.9041
0.001	0.9041

6.3.4 Results

The result for the OCSVM with the augmentation method is shown in Table 4. The five-fold cross-validation algorithm employs the normal and abnormal data as the train and validation set for tuning the hyperparameter for the OCSVM. As described in Sect. 6.3.3, in each iteration in the cross-validation, four subsets of data are used for training. The fifth subset of data is chosen to validate the model with specific ν. The last 20% of the normal data are reserved for testing the final anomaly detection model.

Table 4 Results of the anomaly detection model

Result/model	OCSVM	OCSVM with Random guided warping method
F1-score	0.67	0.89
Recall	1.0	1.0
Precision	0.5	0.8

Table 4 shows the results of the anomaly detection model with test set that includes 20% of normal data and five anomalous samples. We used the data augmentation method to improve the performance of the model. The results in Table 4 indicate that augmented data help improve the model performance by generalizing the model.

The OCSVM, with the assistance of augmented training data, can define a boundary around the normal samples; therefore, the abnormal products can be successfully labeled by testing the force measurements recorded while assembling them with the anomaly detection model.

7 Conclusion

This case study aims to establish a DT for a physical system, trace back the anomalies to the leading source, and predict the quality of the products with more confidence, higher speed, and less invasive methods. Moreover, the DT can help the operators by visualizing the signals related to the assembly process. In this case, they better understand the process and the machine.

This chapter discussed developing a DT for a medical device assembly pilot line. First, we described the physical machine and the product in detail and clarified the process of interest where we focused on developing the machine learning tool. In the process of interest, the subassemblies are mounted together with vertical displacement and applied force. The critical point in the process is the snap process quality, where two components should engage precisely.

Second, we introduced the Kafka data ingestion tool to collect and store data locally for further analysis. In this case, the binary data are collected through PLCs via Kafka with efficient speed and then consumed and converted into a structured format and published back into new topics ready for further consumption by data analysis tools.

Then, we presented the CATCH.AI as a tool for creating and configuring the dashboard, visualization, and feedback loop that can reconfigure the physical system. In addition, CATCH.AI can organize the data collected from the physical system, make it easily accessible, and trigger the external data analysis tools for data mining purposes.

Eventually, we introduced the anomaly detection model and the experiments we conducted to assemble the normal and abnormal products. First, we applied a one-class support vector machine model to determine the boundary around the normal data. To make the model generalize better, we used an augmentation algorithm to widen the decision boundary. The anomaly detection model is reliable, and we can apply a similar model to detect abnormal samples in the other steps of the assembly process.

One of the challenges in this case study has been collecting a large amount of data from different assembly process steps with high throughput and low latency, storing the data in a structured way and mining a large amount of data. Therefore, to overcome these challenges, we propose different solutions; the Kafka data layer can collect the data with high throughput and store the data locally. However, a cloud-based storage solution will be considered a permanent solution. Moreover, by focusing on one assembly step at a time, we split the big data mining problem into smaller subproblems; therefore, we analyzed a smaller volume of data to extract the knowledge.

Acknowledgements We would like to thank the following people from Novo Nordisk A/S; Jeppe Wind, Tinna Dofradóttir, Thomas Algot Søllested, and Sebastian Dengler for providing the physical system, conducting data collection, and insights to understand the assembly process.

Additionally, we extend our gratitude to Novo Nordisk A/S, CIM Industrial A/S and Alexandra Institute A/S for their participation in this case study.

References

1. Melesse, T. Y., Pasquale, V. D., & Riemma, S. (2020). Digital twin models in industrial operations: A systematic literature review. *Procedia Manufacturing*, *42*, 267–272.
2. Lu, Y., Liu, C., Kevin, I., Wang, K., Huang, H., & Xu, X. (2020). Digital twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robotics and Computer-Integrated Manufacturing*, *61*.
3. Hu, W., Zhang, T., Deng, X., Liu, Z., & Tan, J. (2021). Digital twin: A State-of-the-art review of its enabling technologies, applications and challenges. *Journal of Intelligent Manufacturing and Special Equipment, 2*(1), 1–34.
4. Schallmo, D., Williams, C.A., & Boardman, L. (2017). Digital transformation of business models—Best practice, enablers, and roadmap. *International Journal of Innovation Management, 21*(1).
5. Glaessgen, E., & Stargel, D. (2012). The digital twin paradigm for future NASA and US air force vehicles. In *AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference*, Honolulu, Hawaii.
6. Kritzinger, W., Karner, M., Traar, G., Henj, J., & Sihn, W. (2018). Digital twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine, 51*(11), 1016–1022.
7. Tao, F., & Zhang, M. (2017). Digital twin shop-floor: A new shop-floor paradigm towards smart manufacturing. *IEEE Access, 5*, 20418–20427.
8. Minerva, R., Lee, G. M., & Crespi, N. (2020). Digital twin in the IoT context: A survey on technical features, scenarios, and architectural models. *Proceedings of the IEEE, 108*(10), 1785–1824.
9. Sax, M. J. (2018). *"Apache Kafka," in encyclopedia of big data technologies*. Springer International Publishing.
10. Schroeder, G. N., Steinmetz, C., Rodrigues, R. N., Henriques, R. V. B., Rettberg, A., & Pereira, C. E. (2021). A Methodology for digital twin modeling and deployment for industry 4.0. *Proceedings of the IEEE, 109*(4), 556–567.
11. Kannan, K., & Arunachalam, N. (2019) A digital twin for grinding wheel: An information sharing platform for sustainable grinding process. *Journal of Manufacturing Science and Engineering*, *141*(2).
12. Hu, L., Nguyen, N.-T., Tao, W., Leu, M. C., Liu, X. F., Shahriar, M. R., & Su, S. M. N. A. (2018). Modeling of cloud-based digital twins for smart manufacturing with MT connect. *Procedia Manufacturing, 26*, 1193–1203.
13. Zheng, Y., Yang, S., & Cheng, H. (2019). An application framework of digital twin and its case study. *Journal of Ambient Intelligence and Humanized Computing, 10*, 1141–1153.

14. Wang, J., Ye, L., Gao, R. X., Li, C., & Zhang, L. (2019). Digital twin for rotating machinery fault diagnosis in smart manufacturing. *International Journal of Production Research, 57*(12), 3920–3934.
15. Cai, Y., Starly, B., Cohen, P., & Lee, Y.-S. (2017). Sensor data and information fusion to construct digital-twins virtual machine tools for cyber-physical manufacturing. *Procedia Manufacturing, 10*, 1031–1042.
16. Kaigom, E. G., & Roßmann, J. (2016). Toward physics-based virtual reality testbeds for intelligent robot manipulators—An eRobotics approach. In *International conference on intelligent robots and systems (IROS)*, Daejeon, Korea.
17. Feng, H., Gomes, C., Thule, C., Lausdahl, K., Sandberg, M., & Larsen, P. G. (2021). *The incubator case study for digital twin engineering* [Online]. Available: https://arxiv.org/abs/2102.10390. [Accessed April 2022].
18. Feng, H., Gomes, C., Thule, C., Lausdahl, K., Iosifidis, A., & Larsen, P. G. (2021). Introduction to digital twin engineering. In *Annual modeling and simulation conference (ANNSIM).*
19. Paredis, R., Gomes, C., & Vangheluwe, H. (2021). Towards a family of digital model/shadow/twin workflows and architectures. In *International conference on innovative intelligent industrial production and logistics, online streaming.*
20. Ongo, G., & Kusuma, G. P. (2018). Hybrid database system of MySQL and MongoDB in web application development. In *International conference on information management and technology (ICIMTech)*, Jakarta, Indonesia.
21. Hardwick, M. (2017). *Digital twin machining.* STEP Tools, Inc., [Online]. Available: http://www.steptools.com/blog/20171011_twin_machining/. [Accessed April 2022].
22. Zhao, Z., Wang, S., Wang, Z., Wang, S., Ma, C., & Yang, B. (2020). Surface roughness stabilization method based on digital twin-driven machining parameters self-adaption adjustment: A case study in five-axis machining. *Journal of Intelligent Manufacturing, 33*, 943–952.
23. CIM.AS. (2022). CATCH.AI [Online]. Available: https://cim.as/catch-ai/. [Accessed March 2022].
24. Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation, 13*(7), 1443–1471.
25. Iwana, B. K., & Uchida, S. (2021). Time series data augmentation for neural networks by time warping with a discriminative teacher. In *International conference on pattern recognition (ICPR)*, Milan, Italy.
26. Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 26*(1), 43–49.