

Article

# A Universal Machine-Learning-Based Automated Testing System for Consumer Electronic Products

Atif Siddiqui <sup>1,2,\*</sup>, Muhammad Yousuf Irfan Zia <sup>1,2</sup>  and Pablo Otero <sup>1,2</sup> <sup>1</sup> Telecommunications Engineering School, University of Málaga, 29010 Málaga, Spain;

yirfanzia@uma.es (M.Y.I.Z.); pablo.oter@uma.es (P.O.)

<sup>2</sup> Institute of Oceanic Engineering Research, University of Málaga, 29010 Málaga, Spain

\* Correspondence: atsiddiqui@uma.es

**Abstract:** Consumer electronic manufacturing (CEM) companies face a constant challenge to maintain quality standards during frequent product launches. A manufacturing test verifies product functionality and identifies manufacturing defects. Failure to complete testing can even result in product recalls. In this research, a universal automated testing system has been proposed for CEM companies to streamline their test process in reduced test cost and time. A universal hardware interface is designed for connecting commercial off-the-shelf (COTS) test equipment and unit under test (UUT). A software application, based on machine learning, is developed in LabVIEW. The test site data for around 100 test sites have been collected. The application automatically selects COTS test equipment drivers and interfaces on UUT and test measurements for test sites through a universal hardware interface. Further, it collects real-time test measurement data, performs analysis, generates reports and key performance indicators (KPIs), and provides recommendations using machine learning. It also maintains a database for historical data to improve manufacturing processes. The proposed system can be deployed standalone as well as a replacement for the test department module of enterprise resource planning (ERP) systems providing direct access to test site hardware. Finally, the system is validated through an experimental setup in a CEM company.



**Citation:** Siddiqui, A.; Zia, M.Y.I.; Otero, P. A Universal Machine-Learning-Based Automated Testing System for Consumer Electronic Products. *Electronics* **2021**, *10*, 136. <https://doi.org/10.3390/electronics10020136>

Academic Editor: Juan M. Corchado

Received: 8 December 2020

Accepted: 7 January 2021

Published: 10 January 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

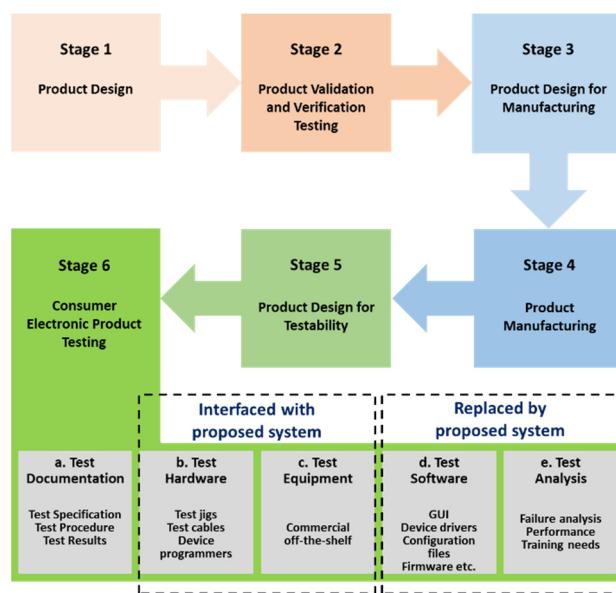
Electronic products and gadgets are an essential part of our daily life. Every day numerous new products are launched. Electronic manufacturing companies are spending a lot of time and effort to design and manufacture these products. There are various parameters that these companies focus on depending on the type and end-users of these products. For example, the products designed and manufactured for the medical industry and deployed underwater communication should be reliable and precise, which is the priority for the manufacturing companies. Some electronic products such as toys are usually low cost but require more features to attract children and for such products quick turnaround is a priority. Successful product launch mainly depends on complete and precise yet quick testing. Failure to do so results in failed products, defects identified after product launch, product recalls, and ultimately loss of a satisfied customer.

### 1.1. Consumer Electronic Product Lifecycle

Consumer electronic product (CEP) manufacturing is based on the product lifecycle where a product design team (PDT) initiates and completes the product design [1], which is qualified through validation and verification testing (VVT) [2]. The next step is to perform the design for manufacturing (DFM) [3] process, which makes the product ready

for manufacturing. The product is then manufactured. To validate that the product is manufactured according to the set standards and design, a manufacturing test is carried out. This manufacturing test is setup and validated through a design for testability (DFT) process [4]. The products that pass this manufacturing test are shipped to the customers.

Figure 1 illustrates the CEP lifecycle. The process starts with the PDT designing a product. The design is validated through a VVT process. This validation testing process can take weeks or months. Depending on the product, it can take a year before the first batch of prototypes is ready for the manufacturing test. The next step is to perform DFM so that the product can be manufactured. DFM process includes sourcing components, build printed circuit board (PCB) and manufacture printed circuit board assembly (PCBA), procure material and build product casing or housing, etc. Once the product is manufactured, DFT is carried out to capture requirements to test the product and identification of tools required for testing. The final stage is testing where the product is tested for functionality as well as manufacturing-related faults.



**Figure 1.** Consumer electronic manufacturing product lifecycle.

During the CEP testing stage, a test site is setup and various activities are carried out, which are shown in Figure 1. The important point is that for every new product, a test site is required to be designed and implemented and the activities mentioned in stage 6 are repeated. Setting up a test site normally takes a few weeks to months to complete. Designing test sites is out of the scope of this research. The system proposed here is interfaced with two building blocks of the test site, while replacing the other two as shown in Figure 1. The proposed system also adds some features, which are normally not available in individual test sites and their details are presented in Section 1.4.

## 1.2. Related Work

Consumer electronic manufacturing (CEM) companies require a system to support and control their processes. These companies deploy enterprise resource planning (ERP) systems to streamline their process and manage various aspects of manufacturing including data collection, storage, and analysis. These systems are essential to this industry. ERP systems, though, cover all aspects of the manufacturing industry, but they still do not provide a direct interface to test equipment and unit under test (UUT) sites. There is also no universal system, both software and hardware, available for connecting to CEP test sites, so for every new product, a test site is required to be setup, which takes time and is also expensive.

In this research, a background study has been done within the categories mentioned in the following sections. The implementation of the proposed system is based on the outcome of this study.

### 1.2.1. Test Data Collection and Analysis Techniques

Data collection for any system is not an easy task. It is important to have a balance between what is important and useful. Collecting all data means there can be storage capacity issues. It is important to have a good understanding of what data are to be collected and useful for analysis at a later stage. In [5], the authors presented a process to measure the quality of data. This approach helps to decide what data are to be collected and stored, resulting in efficient use of data storage space. Having an automated process also improves consistency. In [6], the authors presented a virtual factory concept to perform modeling and simulation. This approach clarifies the process before actual implementation and helps the companies to decide how and what data to collect. The conclusion and recommendations from this paper can form the base for further research. Authors in [7] discuss the defects in electronic products and how this approach can be used by companies to streamline their processes. The defects in electronic products are first considered based on the problems encountered during the design phase for new products. Once the first few batches of the products go through production then the process is updated, and actual faults or defects are considered. This is another approach that helps to determine what data are to be collected.

The manufacturing data collection can be divided into two categories. First is the information related to performance, key indicators, etc., while the other one is specific to a product type. In [8], the authors presented an automated system to test backplanes. In this approach, the test data are collected automatically, which improves consistency, repeatability, and is timesaving. These data can also be used to efficiently design similar systems in the future without going through each and every step once again.

In [9], the authors focused on the data capacity issue. They used cloud computing and storage. This is more relevant to large scale manufacturing companies where data storage is an issue. However, small-scale manufacturing companies can also benefit from this approach. Cloud computing has gained popularity as more companies are looking at moving towards a cloud-based solution. In [10], the authors discussed sensor data management. A data processing framework for data sourcing, analysis, and visualization is presented by the authors in [11].

### 1.2.2. ERP Systems Used in the CEM Industry

In this research, the implementation and use of ERP systems within the test department of the CEM industry is considered. Various aspects of existing ERP systems including advantages, disadvantages, and limitations are reviewed within the scope of activities carried out during testing of electronic products. The proposed system is designed considering certain important parameters of existing ERP systems, which include cost, complexity, and limitation.

Manufacturing companies mostly use ERP systems in their factories. Some commonly used ERP systems used in the CEM industry are EPICOR [12], Syspro [13], Microsoft Dynamics [14], Sage Business Cloud X3 [15], SAP Business ByDesign [16], and Oracle ERP [17]. Details of these ERP systems are presented in Table 1 in alphabetical order. The names of the ERP systems are presented in column 1, and the deployment in the CEM industry (small, medium, and large) scale is shown in column 2. The features are provided in column 3 followed by the limitations in column 4. Finally, the unavailability of the hardware interface is explicitly mentioned in column 5. It can be seen from Table 1 that none of the ERP systems provide a direct interface to hardware or test equipment.

**Table 1.** Enterprise resource planning systems implemented in consumer electronic manufacturing companies.

ERP Systems	Deployment in the CEM Industry			Features	Limitations	Hardware Interface
	Small	Medium	Large			
EPICOR [12]	✓	✓	✓	Production and material management, report generation	Limited financial analysis, user complexity	N/A
Microsoft Dynamics [14]	✓	✓	✓	Product visualization, financial forecasting	Complex front end, difficult to interface with 3rd party tools	N/A
Oracle ERP [17]	✓	✓	✓	Flexibility, easy integration of modules	Difficult training, low system performance	N/A
Sage Business Cloud X3 [15]	✓	✓	✓	Inventory management, production interface, shop floor, and quality control features	Difficulty in accessing interfaces quickly	N/A
SAP Business ByDesign [16]	✓	✓	✗	Improved data analysis, data visualization, application grow with the business	Difficult customization, not user friendly	N/A
Syspro [13]	✓	✓	✓	Flexibility, cost tracking	Complex interface, difficult to integrate with 3rd party tools	N/A

Further comparison of ERP systems is provided in [18]. The authors conducted a survey and studied various benefits provided by ERP systems and their impact on companies. In [19], the authors interfaced their system with ERP and extracted relevant data. The paper in [20] discusses factors affecting manufacturing cost and have used analytic resources to achieve manufacturing test cost reduction.

### 1.2.3. Machine Learning Techniques

The use of machine learning is increasing in various industries. In this section, different approaches and applications of machine learning are reviewed. In [21], authors used Python to develop a machine-learning algorithm using the Scikit Learn module. The authors highlighted limitations in the performance of machine-learning techniques in [22]. The use of historical data for detecting deviation from normal performance is mentioned in [23] along with the limitations of existing machine-learning algorithms and tools. The machine-learning algorithms are reviewed in [24] with the focus on quality control and how production lines can be analyzed and benefited from machine-learning techniques. Finally, various machine-learning algorithms are compared in [25].

### 1.2.4. Review of Some Existing Test Sites

In this section, some existing test sites are reviewed and details including product type, hardware, and software details are provided. The important point is that for each of these test sites, the authors developed their test software or used a commercially available application. For taking measurements, the authors used different commercial off-the-shelf (COTS) test equipment.

In [26], the authors set up a test site for testing PCBs. They have used LabVIEW for controlling a pci eXtensions for instrumentation (NI PXI) system. The authors also provided details of test jig, other cables and highlighted the importance of automated testing. In [27], the authors used an open-source application developed in LabVIEW for acquiring data and instrument control. They used this software for controlling a LASER for imaging and depth profiling mass spectrometers and other instruments. A test site is setup for temperature measurements in [28], acquired through a wireless system and deployment scenarios are presented. In [29], authors used LabVIEW as the platform for testing an induction motor. The test software is used for both data acquisition and motor control. The

authors in [30,31] presented a site for a radio frequency (RF) product. They used a COTS vector network analyzer (VNA) for taking measurements. A test site is setup in [32] for a complex digital product where the authors used a joint test action group (JTAG) port for downloading firmware at a high speed. Similarly, a test site is setup in [33] for testing an aerospace product. This is a fully automated test site that includes fault finding capability. Finally, the authors in [34] used LabVIEW for instrument control. They interfaced a power supply and digital multimeter using a general-purpose interface bus (GPIB) interface.

Table 2 provides a summary of the test sites reviewed here. This table lists the basic information related to product type, hardware, and software for each test site is provided.

**Table 2.** Summary of some existing test sites.

Reference	Product Type	Hardware/Interface	Software
[33]	Aerospace	Automated test equipment (ATE)	Visual basic
[30,31]	Antenna	VNA	Simulation software
[32]	Complex digital	JTAG	JTAG configuration manager
[29]	Induction motor	Data acquisition (DAQ)	LabVIEW
[27]	Laser	DAQ	LabVIEW
[26]	PCB	NI PXI	LabVIEW
[28]	Temperature sensor	Antenna and VNA	COTS software
[34]	Wireless sensor network node	Power supply and digital multimeter	LabVIEW

### 1.3. Limitations of Existing Systems

When a new CEP is launched, the test development team has to go through all the steps and design a standalone test site, which can only be used for testing that specific CEP. A typical test site includes test equipment, interfaces, test jigs, test software, etc. In the absence of a universal test site framework, there is very limited reuse of available test sites.

For automated testing, a software application is developed for each CEP, which also requires validation. Time to market CEPs is important so any delay added to an already tight schedule means revenue lost and chances of faulty products being shipped.

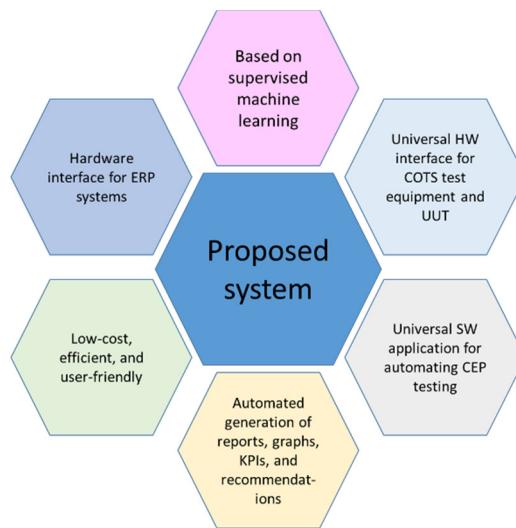
In a typical CEM company, thousands of UUTs are tested monthly, and there is no central system that collates data, reviews faults, recommends repair or rectification of fault, and presents these in different categories. As an example, a UUT can fail due to more solder on a component pin, and the solution is to reduce the solder paste. This is an issue for the conventional soldering department, so unless this department is informed, the fault may not be rectified. Manual fault finding and putting them in categories is a tedious task and can take several hours.

Designing any universal system requires knowledgebase and historical data, which is a difficult task. To collect this data, a process is required, which can focus on what information is available, and how this information can be collected and categorized.

### 1.4. Novelty of this Research

The proposed system is low-cost, efficient, and user-friendly and provides a solution to all tasks performed in a typical test department of a CEM company. Figure 2 presents six novel features of the proposed system. These features include a universal hardware interface for COTS test equipment and UUT interfaces, a universal software application for automating CEP testing, test data collection and instrument control, automated reports, graphs, KPIs, and recommendation generation for continuous improvement based on supervised machine learning. Unlike the existing modules used in test departments of CEM companies, the proposed system can be connected to ERP systems to provide direct access to test hardware, including test equipment and the electronic products being tested.

Alternatively, it can also be deployed standalone system as a replacement for the ERP system specifically in the test department of a CEM company.



**Figure 2.** Novel features of the proposed system.

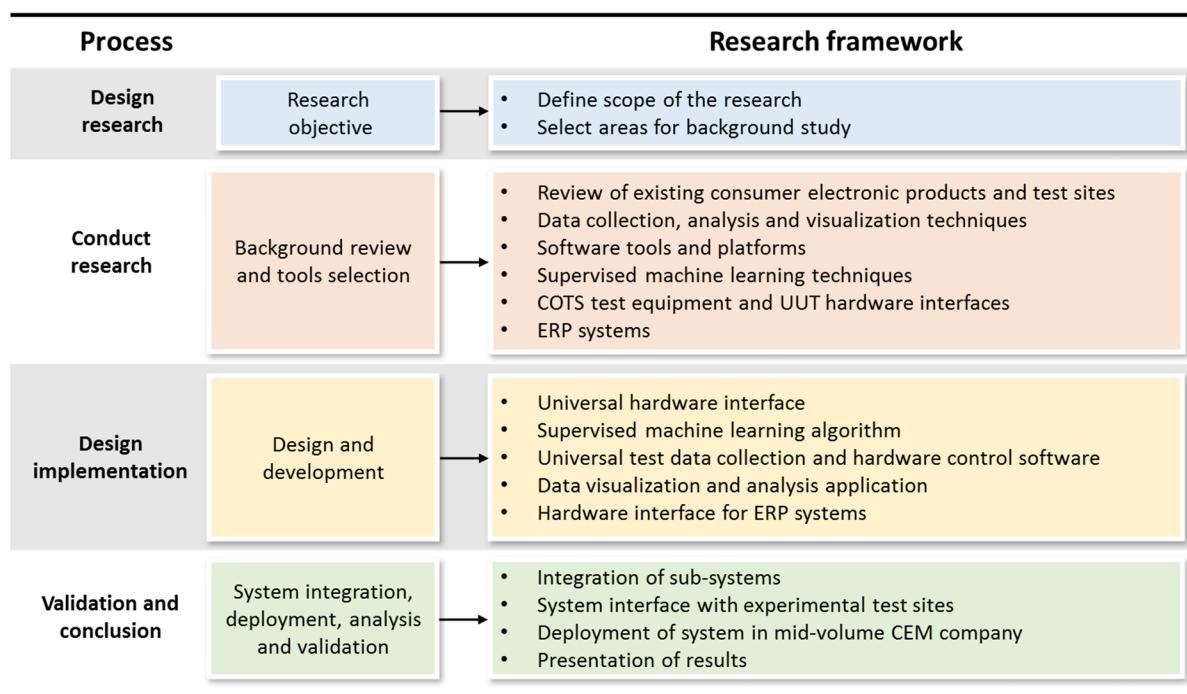
In addition to the proposed system, the article provides a detailed insight into all activities related to CEP testing within the CEM industry. This opens venues for researchers to explore this under-researched area. Some problems are highlighted, and their solutions are provided. A machine-learning algorithm is applied to CEP testing for the first time at this scale. The fault detection and categorization process, and a learning dataset is created and presented here.

### 1.5. Organization of the Article

The rest of the paper is structured as follows. Section 2 presents the research methodology through four processes. Each of these is discussed in this section. The proposed automated testing system is discussed in Section 3. The system block diagram, data interfaces, and architecture of the machine-learning-based application are also presented in this section. Section 4 discusses the experimental setup for validation of the universal test hardware interface and universal test data collection sub-systems. Section 5 presents the experimental results for validation of the proposed test data analysis sub-system. Section 6 provides an overall discussion, and finally, the conclusions are drawn in Section 7.

## 2. Research Methodology

The proposed system provides a universal and one-window solution for testing products within the CEM industry. This manuscript focuses on a universal system to control existing test sites, test data collection and analysis, and recommendations; however, designing of test sites is outside the scope of this research. Figure 3 shows the various processes carried out within the research methodology. These processes are listed within design research, conduct research, design implementation, and validation and conclusion.



**Figure 3.** Research methodology.

### 2.1. Design Research

In this section, various techniques, applications, and processes used by CEM companies are studied. For this purpose, a review is carried out within the categories mentioned in Section 1.2 (related work). A thorough and extensive review is carried out before designing the proposed universal testing system.

### 2.2. Conduct Research

The first step is to review the effectiveness of the proposed system. For the proposed system to be effective, lots of historical data are required. Therefore, manufacturing test data for around 100 test sites and 125 CEPs is collected for months where the products, i.e., UUTs range from simple digital, complex digital, camera, analog, high precision, RF, communication, and low-frequency products, etc. The review includes studying various aspects of these test sites, identification and collection of useful data, and trends analysis. The main categories of this review include test software, test equipment, and test measurements. This wide range of test data has helped in building a knowledge base related to these product types, which includes the test times, faults and repair, test operator (TO) skills needed, and other parameters. This information is also used in defining performance-related thresholds that are published using key performance indicators (KPIs). Some existing test sites are also reviewed, and details are presented in Section 1.2.4. Secondly, various data collection, analysis, and visualization techniques are reviewed in this research. The specific details of different techniques are discussed in Section 1.2.1.

The third step is to review test software for the test sites being evaluated so that a universal test application can be developed that takes into account the features reviewed in existing test sites software. To develop an application as an alternate for ERP systems and universal test site software, different available packages, or software platforms have been evaluated, two of which (LabVIEW and Python) are discussed here. LabVIEW [35] is selected as the software application to implement the whole process primarily due to several useful built-in features. One of the features is a web-based control of the application, which is presented in [36]. In [37], the authors interfaced LabVIEW with Arduino, which is another important feature where LabVIEW can be used to control hardware. The image processing functions are used in [38] in LabVIEW. There are test sites

for camera-related products and having these built-in functions helps in integrating with the proposed system. In [39], the graphical user interface (GUI) and data acquisition (DAQ) features of LabVIEW are used. The authors in [40] have used TCP/IP built-in functions of LabVIEW. In [41], authors used LabVIEW for interfacing with field programmable gate array (FPGA). In both the previous references, the authors used LabVIEW for hardware interfacing. The advantage of using LabVIEW is that this application can be used to implement every stage in the process including setting up test sites. If an existing test site software is already available in other applications, then LabVIEW also provides an interface to those applications. An advantage of using LabVIEW is that the existing test software or algorithms created in other applications can be easily and quickly integrated within LabVIEW. A closed-loop controller application is implemented in LabVIEW in [42]. The quality of LabVIEW GUI is far better than some other software applications with the added advantage of quick development. In [43], authors used LabVIEW's vision control module for image acquisition and interfaced their application with Lego Mindstorms EV3 controller. They have also highlighted the quick development feature of LabVIEW. There are other applications available such as Python [44]. The authors in [45] used Python to create a suite to control laboratory experiments, which include hardware interface and GUI. Data visualization is also an important aspect of the research, and authors in [46] presented an interactive visualization system developed in Python.

The fourth step is the review of machine-learning techniques, which is already discussed in Section 1.2.3 and includes the areas where these techniques are applied.

The fifth step is to review a hardware interface required to connect the proposed system to the test site under consideration. Here, the details of the CEP test sites are reviewed, which include COTS test equipment, their interfaces, test measurements, and UUT hardware interfaces. The test sites' data show the complexity and variety of COTS test equipment, their interfaces, and test measurements, which the proposed system can handle. Some COTS test equipment is identified that include signal generators, arbitrary waveform generators, spectrum analyzers, power meters, network analyzers, digital multimeters, power supplies, etc. This COTS test equipment is controlled using different interfaces, which include GPIB, LAN, USB, RS232, etc., which are added to the proposed universal hardware sub-system. Details of test measurements are also collected. Some of these are audio, RF, amplitude modulation (AM) and phase shift keying (PSK) signal generation using signal generators, frequency and spurious and adjacent channel power measurements using spectrum analyzers, cable impedance and S-parameter measurements using network analyzers, amplitude and frequency measurements using oscilloscopes, basic measurements using digital multimeters, etc. Many test sites reviewed also require connecting to the UUT directly without going through COTS test equipment. Some of the UUT interfaces reviewed include a serial peripheral interface (SPI), inter-integrated circuit (I2C), RS422, RS485, JTAG, etc. These interfaces are used for various test activities including firmware programming, UUT configuration, boundary scan, enabling/disabling of parts of the circuit, etc.

Finally, the last step is the review of the ERP systems. The ERP systems deployed by CEM companies are reviewed in Section 1.2.2.

### 2.3. Design Implementation

Based on the findings, a universal hardware interface and software application is proposed, which stores the most common COTS test equipment drivers and several software subroutines for test measurements. Apart from that, device drivers and software subroutines for taking measurements from some common interfaces available on UUTs are also stored in the database. During the review several manufacturing test parameters are also identified and collected and based on the research findings, it is demonstrated that successful data analysis can be achieved by using the parameters identified after thorough research and presented in this manuscript.

LabVIEW is selected as the software application based on the review. LabVIEW files are called virtual instruments (VIs) and are saved with “.vi” extension. LabVIEW provides a modular approach where sub-routines can be created and stored as subVIs. The outcome of this review is a top-level GUI application through which subVIs for test measurements and COTS test equipment drivers are selected from the database. The selected test measurement subVIs can be executed in a defined sequence.

The process mentioned above requires decision making without human interaction. To cater to this problem, a supervised machine-learning technique is applied, so that the top-level VI can automatically select the required COTS test equipment drivers and test measurements subVIs. The details of COTS test equipment, UUT interfaces, and test measurements are entered by the TO for the test site under consideration and the test software automatically selects the subVIs from the database. The proposed machine-learning-based algorithm uses the COTS test equipment name, model number, interface type, measurement type, etc., to select the required subVI from the database.

The hardware interfaces discussed and reviewed in this section are selected, and a universal hardware interface is proposed, which includes two sub-systems for COTS test equipment and UUT interface, respectively. Details of these interfaces are discussed in Section 3. The proposed system also provides an interface for connecting to manual legacy test sites that do not have any integration capability. Through this interface, TO can manually enter test results, which are then stored in the database. The next step is to review information related to COTS test equipment. A list of COTS test equipment is created, and device drivers, written in LabVIEW, are stored in the database as subVIs. The system is developed and integrated in a way that it is easily expandable, i.e., more COTS test equipment drivers, test measurements, etc., can be added to improve coverage. The universal hardware interface is designed considering both the data rate and the number of devices on the UUT that can be connected.

At this stage, the proposed universal software and hardware sub-systems are integrated. The proposed system is connected to the test site through a universal hardware interface, and test data are acquired. The test data are then processed, and outputs are generated as customizable reports, graphs, and KPIs. Common customer requirements like logging of test times, staff performance analysis, logging of customer order details, test results, work in progress (WIP) quantity, etc., are also considered. The software uses machine learning to generate recommendations from the outputs. This is a unique feature of the proposed system, which can support CEM companies to implement continuous improvement.

The study of existing ERP systems deployed by companies in the CEM industry is carried out to understand various interfaces used, and based on the study, an interface is proposed for connecting to the ERP systems. Through the proposed ERP interface, all raw and analyzed data collected by the proposed system can be forwarded to the ERP system. This interface effectively provides ERP systems with direct access to the test site hardware. The proposed system can also be used as a standalone system in the test department.

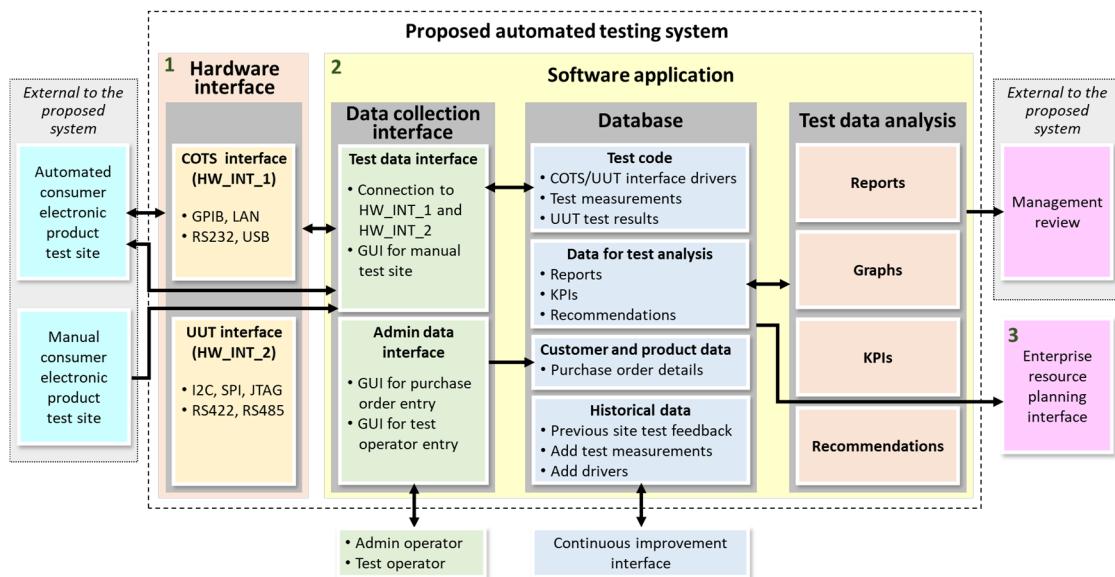
#### 2.4. Validation and Conclusion

The final step is the validation stage. To validate the proposed data collection sub-system, experimental test site setups are connected to the universal hardware and software interface, which are discussed in detail in the next section. The data analysis, report generation, and recommendation sub-system are validated by deploying the proposed system in the test department of a CEM company. The proposed system has been successfully deployed in a mid-volume CEM company in their test department, and results are presented.

### 3. Proposed Automated Testing System

The proposed system is divided into three subsystems namely (1) hardware interface, (2) software application, and (3) ERP interface are discussed here. Figure 4 shows the

top-level block diagram of the proposed system. The figure illustrates the flow of data through a universal hardware interface and machine-learning-based software application.



**Figure 4.** Proposed automated testing system block diagram.

### 3.1. Hardware Interface

The first sub-system is a universal hardware interface for connecting to any CEP test site to control COTS test equipment and direct control of UUT as shown in Figure 4. Due to the absence of any universal hardware interface currently in the industry, this proposed system can save development time, cost, and effort.

The hardware interface is further divided into two subsystems, the first is hardware interface 1 (HW\_INT\_1) for connecting the proposed system to COTS test equipment using GPIB, LAN, RS232, and USB interfaces. The second is hardware interface 2 (HW\_INT\_2), which connects to the UUT through I2C, SPI, JTAG, RS422, and RS485 interfaces. Almost all the new COTS test equipment can be connected via one of the interfaces proposed here.

### 3.2. Software Application

The second sub-system is a supervised machine-learning-based software application developed in LabVIEW as shown in Figure 4. This application uses supervised machine learning to automatically select drivers for COTS and UUT and select the required measurement test code for any test site through a universal hardware interface. Further, this application collects real-time measurements and other test data, performs data analysis, generates reports, and KPIs. CEM companies and their clients spend months developing software for automated testing of almost every new product; using this universal software application, they can save development and validation time, which leads to a quicker product launch. Finally, it provides recommendations automatically using machine-learning techniques. This sub-system also includes a database that stores test sites' data, other raw manufacturing test data, analyzed data, COTS, and UUT interface drivers, and test code for taking measurements.

This sub-system is further divided into three sub-systems: test data collection interface, database, and test data analysis interface. The test data collection interface connects, controls, and takes measurements from COTS test equipment and UUT. Test measurements code and device drivers are stored in the database. Device drivers and test measurements code are selected from the database automatically using machine learning. The test data collection sub-system provides an interface for entering test data manually for manual legacy test sites. This is an important feature as well, which provides compatibility with the legacy sites, which cannot be automated. Further, a third interface is provided to enter

other manufacturing data, which include purchase order (PO), product, customer, and TO details.

Using the proposed system, the UUT test start time is logged automatically when the TO starts testing, and the test end time is logged when the testing is completed. This feature increases accuracy, and the TO is not required to use another system for time logging. Another advantage is that real-time analysis can be carried out while UUTs are tested. CEM companies can record details of each UUT including test time, faults if any, etc., automatically.

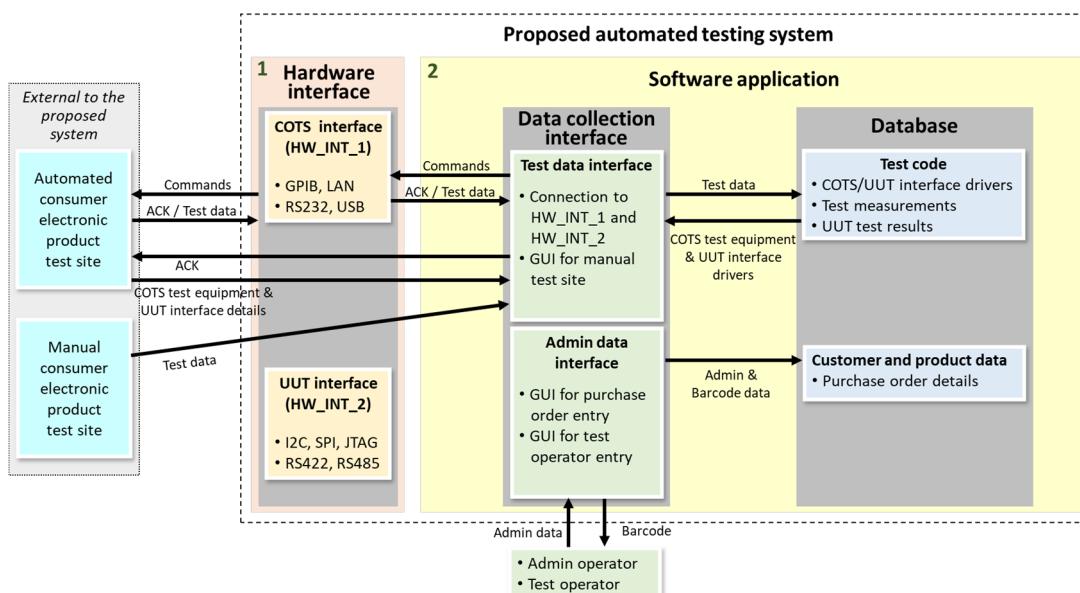
The test data analysis interface retrieves test data from the database, performs analysis, and generates output in the form of customizable reports, graphs, and KPIs. The management team can review the performance using these graphs and KPIs. Another unique feature is that this system provides recommendations automatically using machine learning, based on the results generated. For machine learning to be effective, a lot of data are collected and analyzed, and the management team is interviewed to understand how they make decisions.

The database stores all the test data, device drivers, and test measurement code. A continuous improvement interface is provided so that more data can be added to improve the overall performance of the system, increase coverage, and to improve machine-learning-based decision making. The database stores the data of around 100 test sites reviewed as part of this research. These data are the key to making machine-learning-based decision making efficient and accurate.

### 3.3. ERP Interface

The third sub-system is an interface for ERP systems to connect to COTS and UUT hardware interface and acquire all CEM test department related data directly as shown in Figure 4. This is a unique feature that supports cost and time savings for CEM companies. The proposed interface for ERP systems is established using structured query language (SQL) commands to share both raw test data and analyzed test data. This proposed system is a replacement for the test department module of ERP systems in the CEM industry.

Figure 5 shows the proposed data collection building blocks of the proposed system through various interfaces and interconnections showing the direction of data flow and type of data.



**Figure 5.** Proposed data collection sub-system data flow direction.

Similarly, Figure 6 presents the proposed data analysis building blocks of the proposed system, showing the data flow and connectivity.

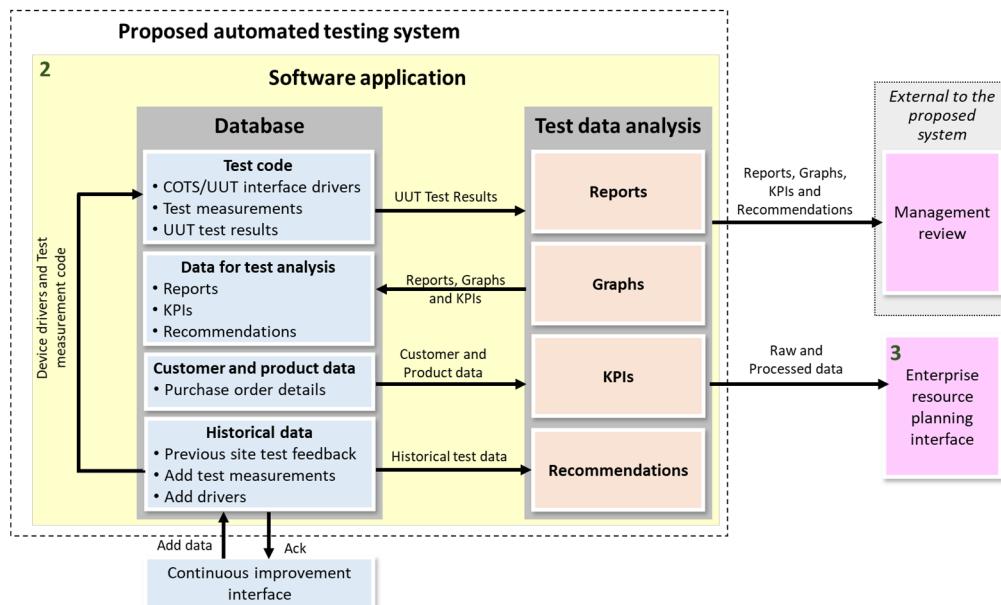


Figure 6. Proposed data analysis sub-system data flow direction.

In Figure 7, the structure of the machine-learning algorithm is presented. Four layers, i.e., input, two hidden, and the output layers are shown in the block diagram. The main inputs are the start and end dates during which recommendations are required. Once the dates are selected, all the CEPs tested in this duration are selected. Recommendations are generated in two main categories, which are product and the individual UUT within that product category. There are sub-categories of recommendations as well as shown within the hidden layers of the figure. The categories are selected based on certain keywords, some of these are listed in the figure. The machine-learning algorithm takes raw text, which is entered automatically by test software for FAIL UUTs and the text entered by TO. Categories are identified by checking for keywords, and once the category is identified, then recommendations are generated using some more words to form sentences. These sentences are also compared with historical data, i.e., previously generated recommendations. In the end, these recommendations are stored in the database. The system also prompts the user to add new keywords picked up from the input text data.

Some examples of machine-learning-based recommendations for specific products are “Test operator 5 is takes more time”, “Test operator 2 is expert in testing product 4”, “Test operator 8 is suitable”, “Product 4 estimated time is not correct”, “Product 3 tests are insufficient”, “Test jig 5 connector J9 is broken”, “Test jig 1 needs maintenance”, and “Test equipment 7 is going out of calibration”. Similarly, some examples for individual UUT are “Component C1 wrong value”, “Component D5 missing”, “IC3 orientation wrong”, “UUT 4 PCB quality is poor”, “UUT 5 solder paste is less”, and “UUT 8 reflow required”.

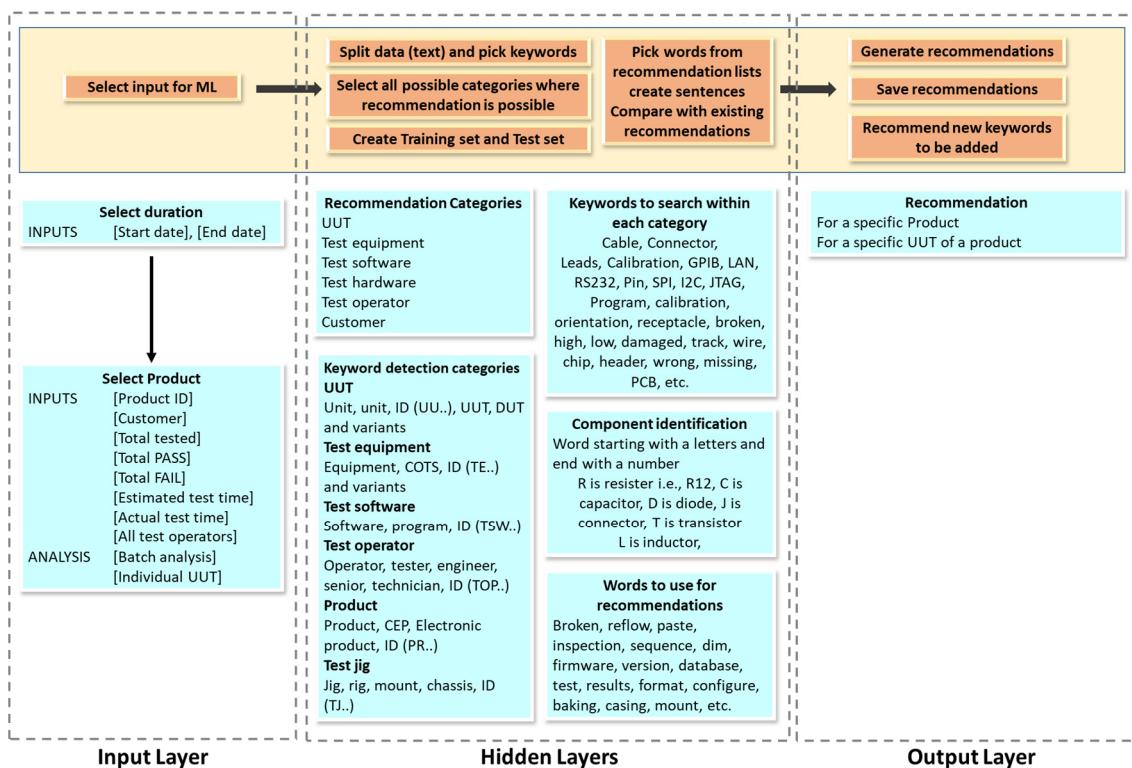


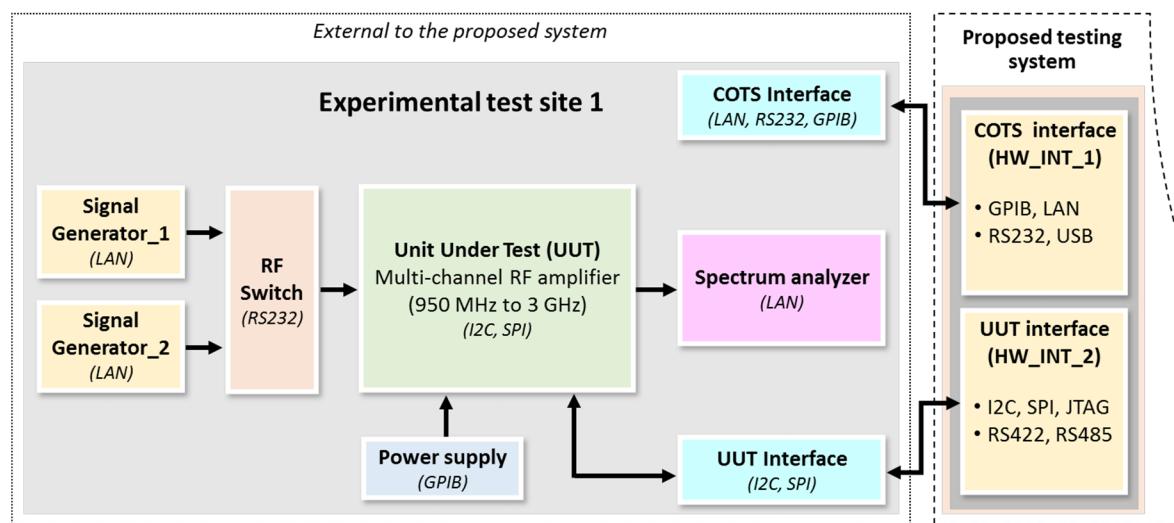
Figure 7. Machine-learning structure for automated recommendations.

#### 4. Experimental Setup

This section presents the experimental setup to validate the data collection sub-system. The validation of the test data collection sub-system, which includes a universal hardware interface, and a universal machine-learning-based software application is carried out using experimental test setups discussed in this section.

##### 4.1. Experimental Test Site Setup-1 for an RF Amplifier

For validation of the universal hardware interface and machine-learning-based universal test data collection sub-systems, an experimental test site is selected, which is shown in Figure 8. The UUT is a multichannel RF amplifier with a frequency range of 950 MHz to 3 GHz. This frequency range is divided into four sub-channels, each having a different gain setting. The sub-channels are selected using the I2C interface on the device. The UUT has an SPI interface that is used for downloading firmware. At the input, two signal generators are connected. The signal generators generate different frequencies within the UUT frequency range. These signal generators are connected to the UUT via an RF switch. Four possibilities are considered, i.e., both signal generators are connected at the same time, either one is connected at a time, or both are disconnected. The UUT is switched ON using a bench power supply. At the output, a spectrum analyzer is connected to check frequency response. The two signal generators and a spectrum analyzer are fitted with LAN interface, the power supply has a GPIB interface, and the RF switch is controlled via RS232 port. The two interfaces on the UUT are SPI and I2C.



**Figure 8.** Experimental test site 1 setup.

The first step is the programming stage where a firmware file is downloaded via the SPI interface. The I2C interface on the UUT is used for selecting different frequency channels. For this experimental setup, two RF measurements are selected, namely, gain measurement and channel isolation.

The TO enters the details of the test site, which includes the make, model, and control port details of five COTS test equipment and two interfaces, i.e., SPI and I2C on the UUT. The TO also enters details of firmware file, which include file location, file size, checksum, etc., and I2C interface commands to enable or disable each channel on the UUT. TO then enters the details of two test measurements, i.e., gain and isolation measurements. The details include frequency range, amplitude scale, amplitude marker positions, etc., for gain measurement and combinations of different channels between isolation is required to be measured. Finally, the test sequence is defined, which is to download firmware, gain measurement, and isolation measurement.

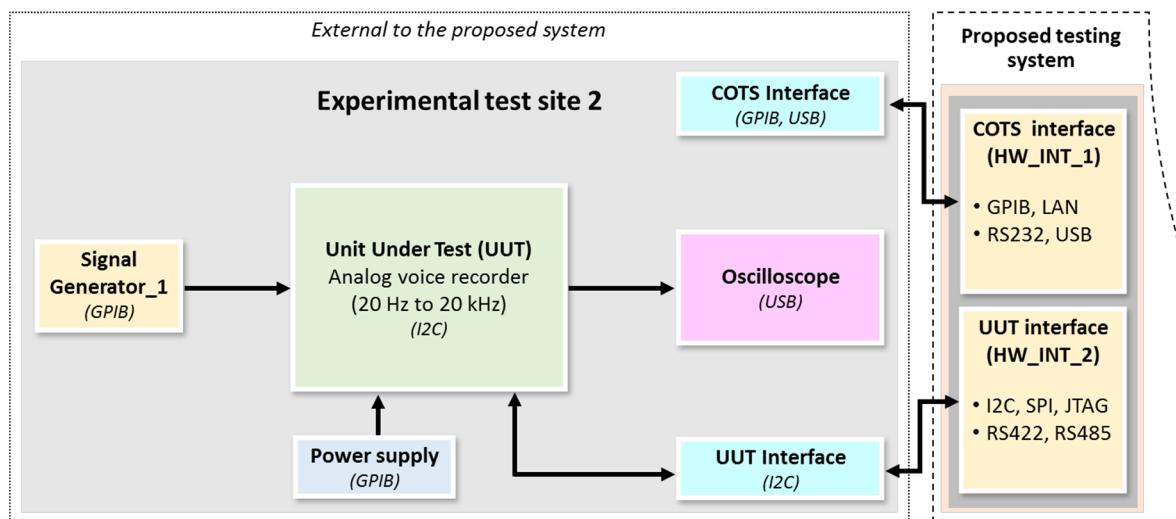
The machine-learning application uses this information to select the required test code, i.e., subVIs from the database. The COTS test equipment device drivers, which are LabVIEW subVIs, are selected using the make, model number, and control port details. For downloading firmware and configuring UUT, SPI and I2C device drivers are selected from the database. The test measurement code is selected from the database using the test measurement name and the COTS test equipment used. Test measurement details such as frequency range, marker position, test limits, etc., are saved in a text file external to the LabVIEW subVI, i.e., test measurement code. For firmware programming, machine learning uses firmware file location, file size, checksum, etc., to download the code. This information is also saved in the external text file. Once all these tasks are completed then the complete information is saved in the database with a unique ID for the test site under consideration so that next time the TO does not need to repeat these steps. TO then initiates testing by entering TO identity, estimated UUT test times, and purchase order details, which include customer name, product part number, etc. The universal test software then performs a health check test on the COTS test equipment to make sure all hardware is connected. The health check test requests equipment to send identity and is a confirmation that COTS test equipment used on the test site can be controlled via the universal hardware interface. The next is to switch ON the power supply and check how much current the UUT is drawing. The test software then executes the tests in sequence and test results are acquired and saved in the database.

Using the proposed system, the test software development and test hardware interface is not required to be designed and built again. These tasks can normally take a few weeks or

months to design and implement for a test site depending on the type. A new device driver or test measurement code can be added through the continuous improvement interface.

#### 4.2. Experimental Test Site Setup-2 for an Analog Voice Recorder

A test site for an analog voice recorder is integrated with the proposed system and automated using the universal interface. The test site includes a power supply, a signal generator, and an oscilloscope. The test site is shown in Figure 9, where the power supply and signal generator are controlled using the GPIB interface, while the oscilloscope is connected via the USB interface. Certain UUT parameters such as gain and filter settings are configured via the I2C interface. This was a manual test site, where TO was taking measurements by setting test equipment and configuring UUT. After connecting to the proposed system, the test site is now fully automated, and overall, 2 min per UUT time is saved. Similar to the previous section, the TO enters the details of COTS test equipment and UUT hardware interfaces. The software then creates the test sequence and executes the test.



**Figure 9.** Experimental test site 2 setup.

## 5. Experimental Results

The quality of test data collected is extremely important for any analysis. The analysis is performed using a test data analysis sub-system and various graphical results and KPIs are generated. More reports or results can be added as per requirements. Some of the graphical results are presented in this section.

The test data analysis sub-system of the proposed system is implemented and then validated in a mid-volume CEM environment for electronic product testing. The proposed system is deployed in the factory, and analysis is performed for the collected test data.

### 5.1. Test Data Analysis Sub-System Software Interface

Test data analysis interface is implemented for supervisors or managers to perform analysis of the test data recorded/logged. An interface is created as a proof of concept and is installed, and results are presented. The analysis can be done based on two categories. The first category includes individual analysis for TOs, product, specific customer, POs, etc. In the second category, reports are generated for any duration by selecting the start and end dates for any of the parameters such as products, customers, POs, TOs, etc.

### 5.2. Manufacturing Test Data Presented by Product (UUT)

In this section, the data collected for different product types are presented graphically. The graphs present data collected for a month. Two product types, “Complex Digital” and

“Camera”, are selected here for analysis. These products belong to different customers and are tested by TOs with different skills. The monthly testing is carried out and presented in this sub-section. The daily hours booked are shown on the *y*-axis, while three different parameters are shown on the *x*-axis. These *x*-axis parameters are test date, TO name and details, and customer details. The data are grouped based on the customers. The curved dotted line shown in the graphs is the data trend, i.e., hours booked by the TOs during the month. The straight line is the median value of the hours booked and the bar charts represent the actual hours booked daily.

Figure 10 shows results for product type “Complex Digital”. The products tested and presented here belong to four different customers, i.e., customer 15, customer 37, customer 40, and customer 5. There is a further division for each customer which is based on TO who worked on these products. Here 8 different TOs are utilized for testing these products.

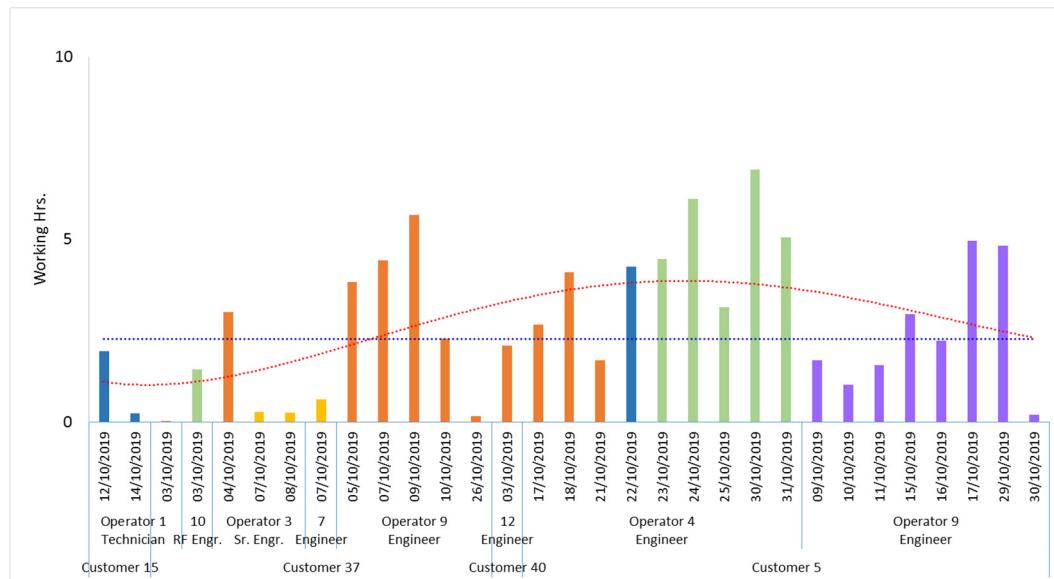


Figure 10. Product type Complex Digital—October 2019.

Figure 11 shows the results for the product type “Camera”. The cameras tested here belong to 2 different customers, and 6 different TOs, including a technician, 2 senior engineers, and 3 engineers, tested this product.

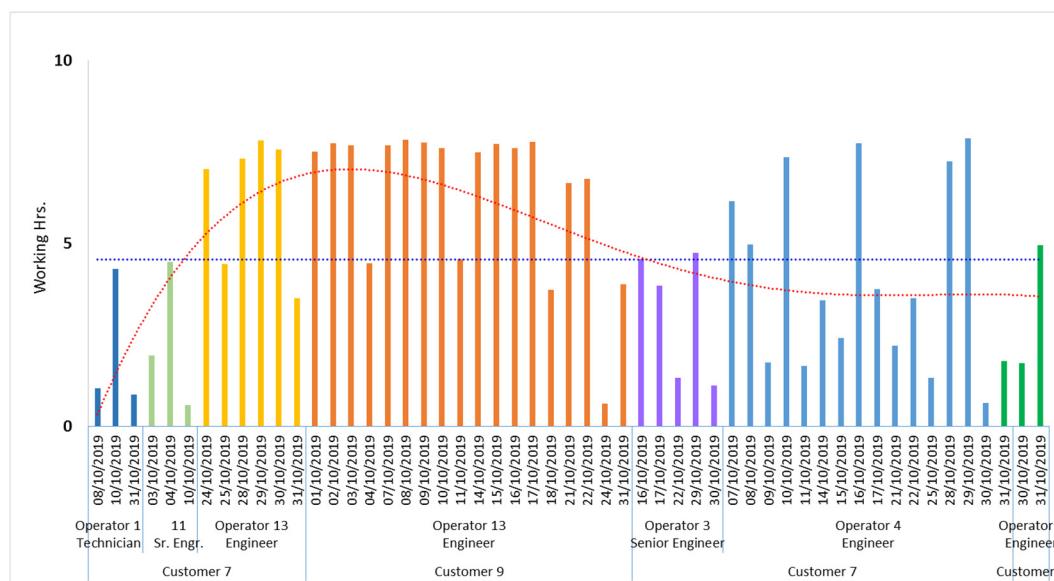


Figure 11. Product type Camera—October 2019.

### 5.3. Manufacturing Data Presented by the Customer

In this section, the data collected, based on customer product, are presented graphically. The graphs present data collected for 2 customers for 3 months. Different TOs were utilized for testing customer products. The product types are based on customer orders. Figures in this section show the testing carried out for 2 different customers for 3 months. The daily time booking is shown on the *y*-axis, while details of TOs and dates are mentioned on the *x*-axis.

Figure 12 shows the test results for customer 15. The graph shows that the TOs spend more time testing this “Complex Digital” product for customer 15 around the middle of the 3 months duration. The increase in test time can vary and is dependent on the order intake and deadlines agreed with the customers. The data are grouped based on the 4 different TOs utilized.

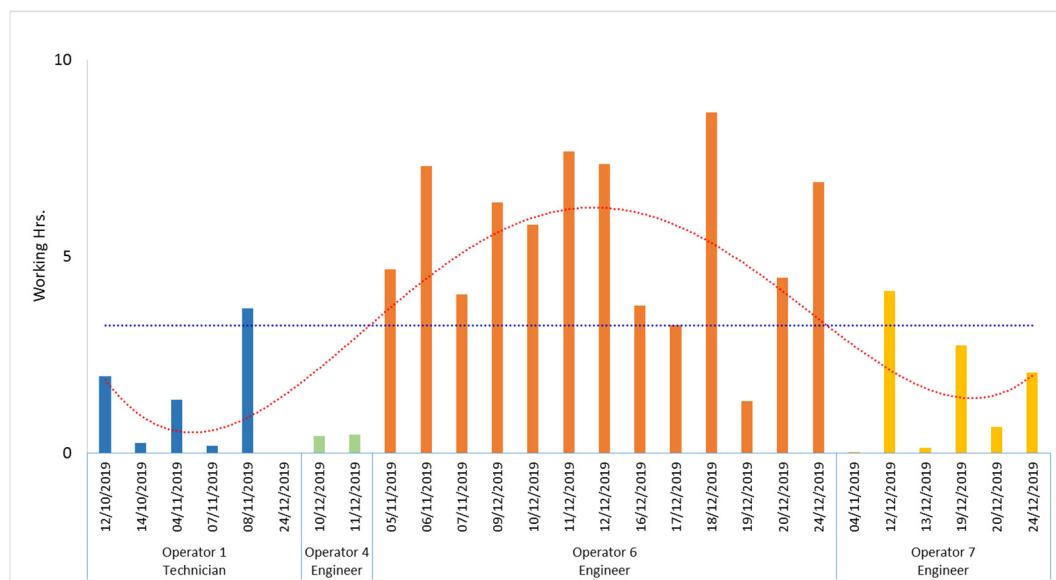


Figure 12. Customer 15, product type Complex Digital—Q4 2019.

Figure 13 shows test results for customer 20 and the product type is “Analog”. The graph shows that 4 different TOs tested this product with the greatest number of hours booked by TO number 3.

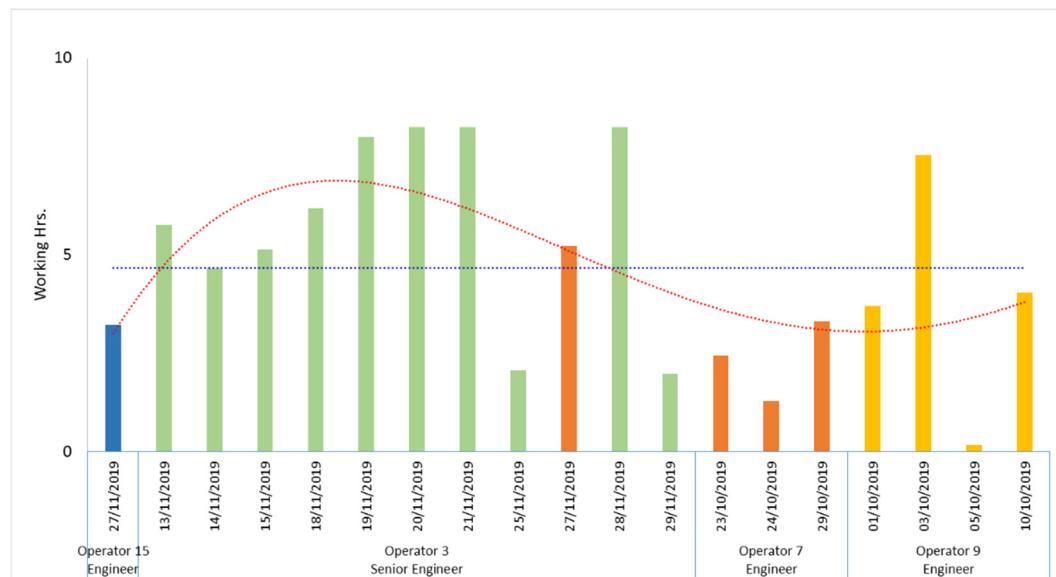


Figure 13. Customer 20, product type Analog—Q4 2019.

#### 5.4. Manufacturing Data Presented by the Test Operator (TO)

In this section, the test data are collected and shown graphically for a month. The graphs are plotted for 2 different TOs. These graphs are useful in analyzing the performance of the TOs. The same analysis can also be performed for more than a month. Graphs shown in this section are for 2 different TOs. Data are collected and presented for a month.

Figure 14 shows the test hours booked by TO number 15. During this time, TO number 15 tested products for 5 different customers as shown on the x-axis. Here 4 different product types were tested, which are grouped as shown on the horizontal axis. The TO also booked some indirect hours, i.e., during this time the TO is not testing any customer products, but spending time on a test jig maintenance.

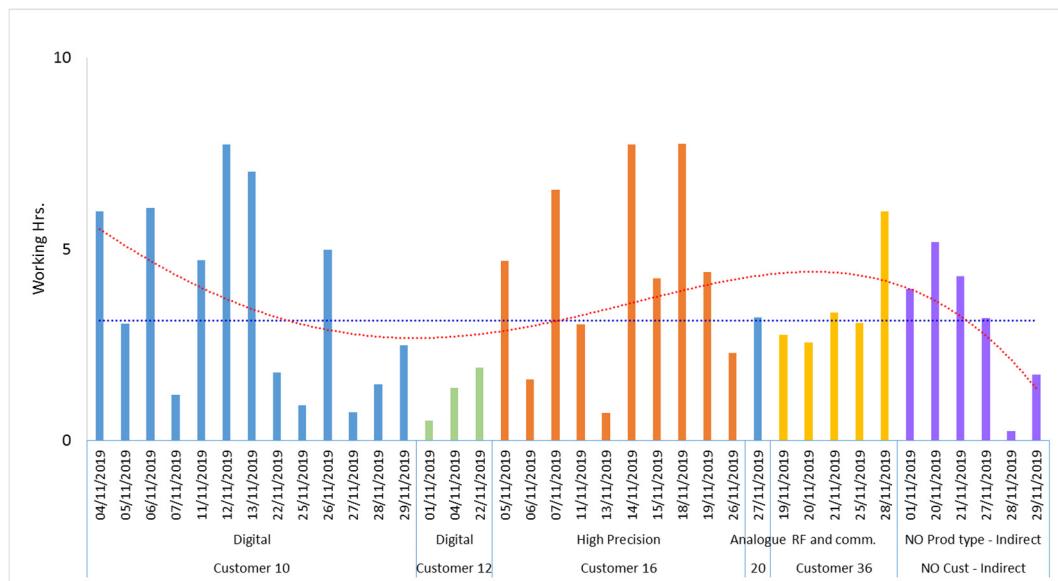


Figure 14. Test operator (TO) 15 (Engineer)—November 2019.

Figure 15 shows the test hours booked by one of the test technicians. TO number 16 tested 7 different product types for 9 different customers. TO number 16 also booked a fraction of time as indirect booking. During this time, this operator attended a customer visit.

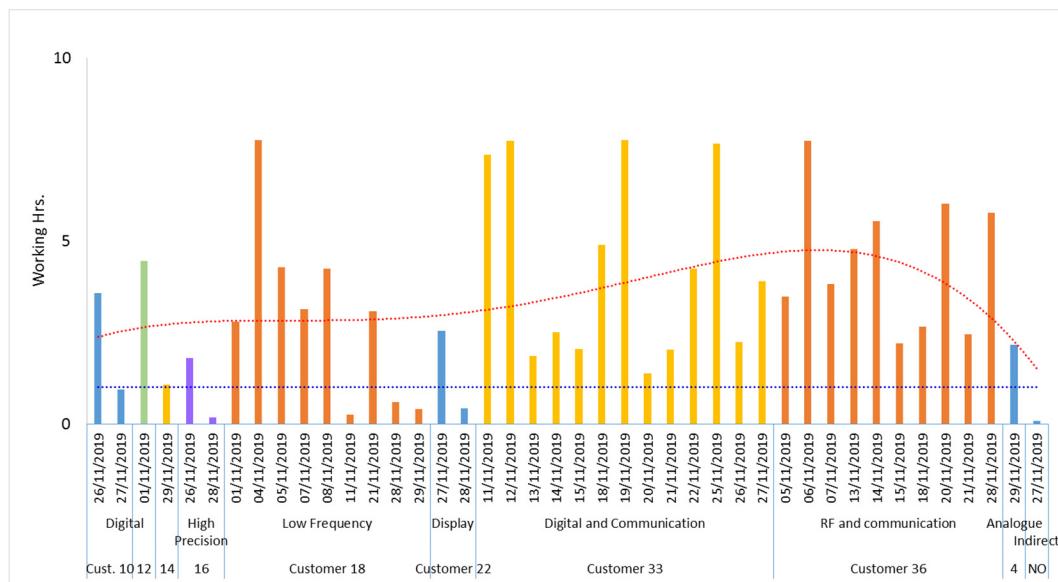
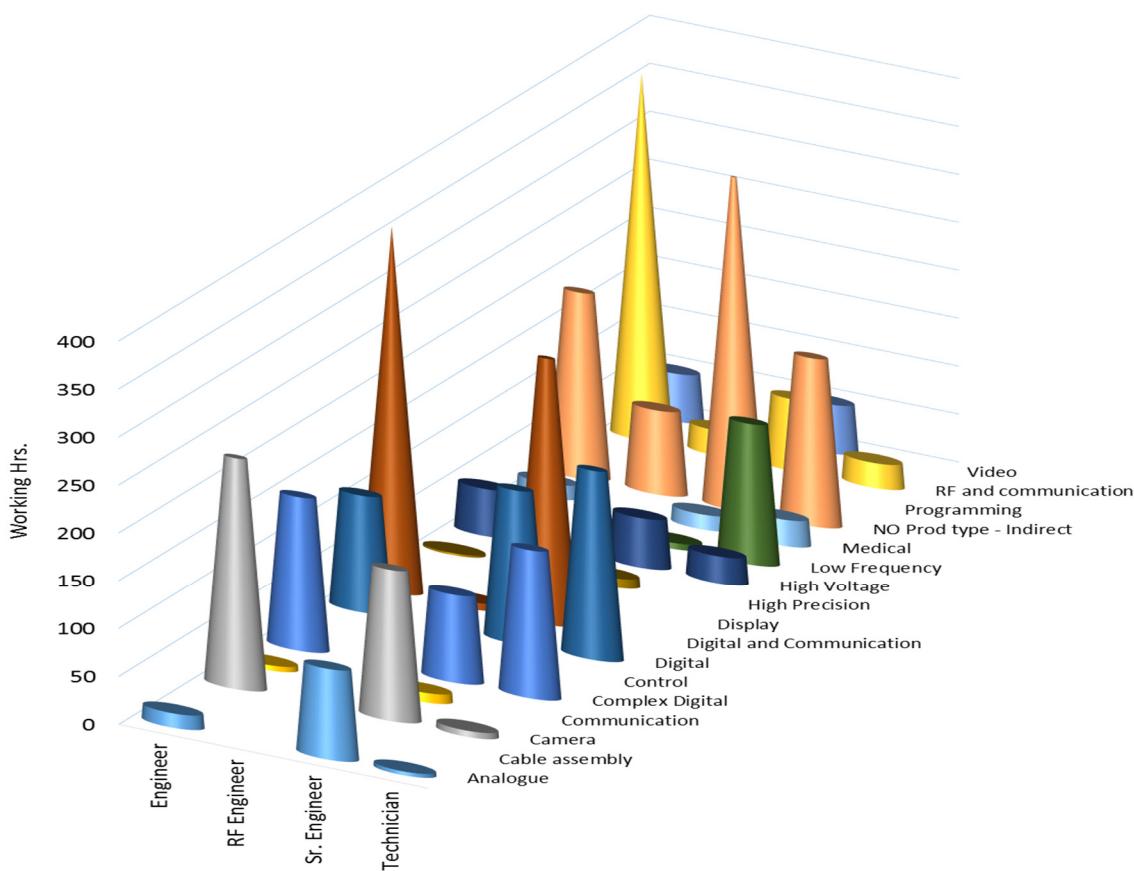


Figure 15. Test operator (TO) 16 (Technician)—November 2019.

### 5.5. Manufacturing Operator Performance—Product Wise

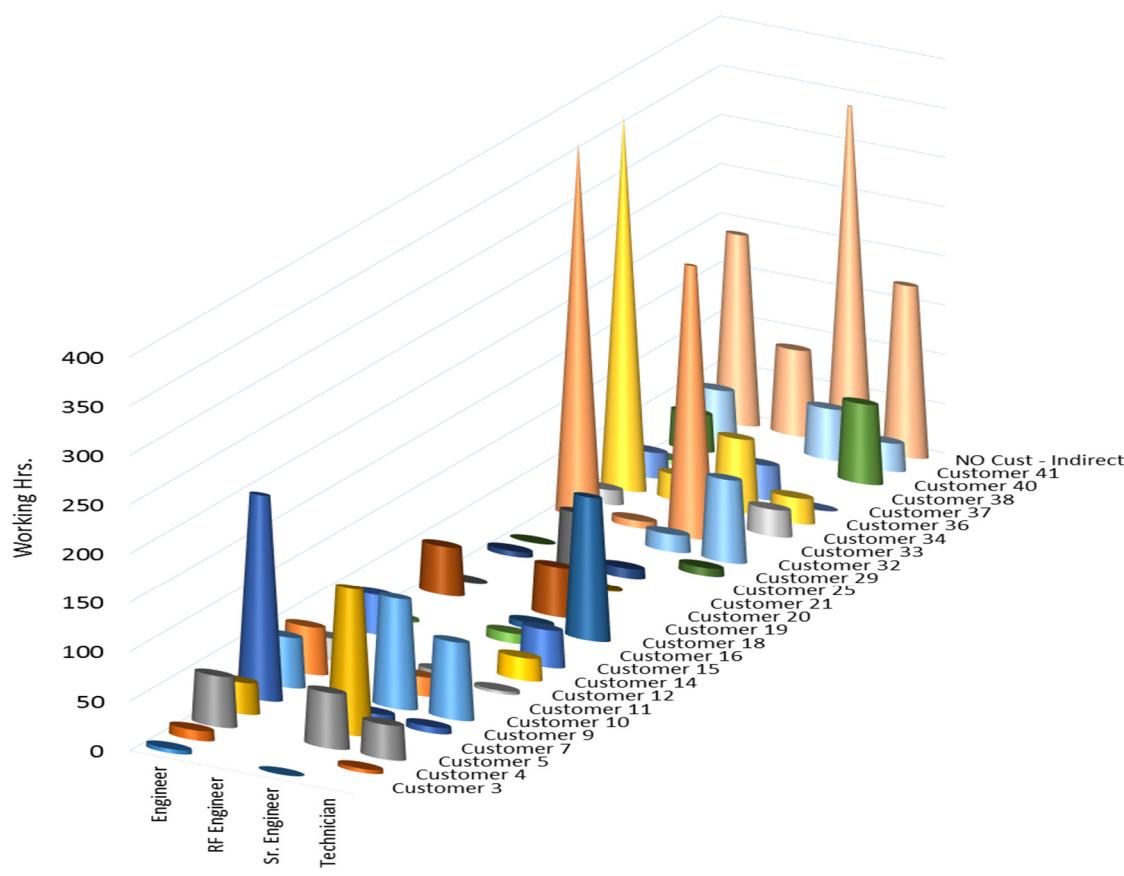
In this section, the test data collected are presented for the range of products tested during a quarter of the year. In Figure 16, testing carried out and hours booked by TOs within the categories technician, engineer, senior engineer, and RF engineer are shown on the vertical axis. The TOs and product range are listed on the other 2 axes. The total products tested were within the 14 product types. During this period, the TOs have also booked some indirect hours, i.e., time booked for non-products. Among the 4 different TOs, RF test engineers booked approximately 3% of the total hours booked due to the specialized RF testing carried out on a limited RF product range. Engineers and senior engineers booked approximately 75% hours between them. The share of hours is not fixed and can vary depending on the number of staff employed within each category, order intake, quantity, and test times of the products tested, etc.



**Figure 16.** Product wise operator performance—Q1 2019.

### 5.6. Manufacturing Operator Performance—Product Wise

In this section, the test data collected are displayed for all the customers whose products were tested during a quarter of the year. In Figure 17, the test hours booked by TOs within the categories technician, engineer, senior engineer, and RF engineer are shown on the vertical axis. The TOs and customer names are listed on the other 2 axes. During this quarter, testing is carried out for 25 different customers.



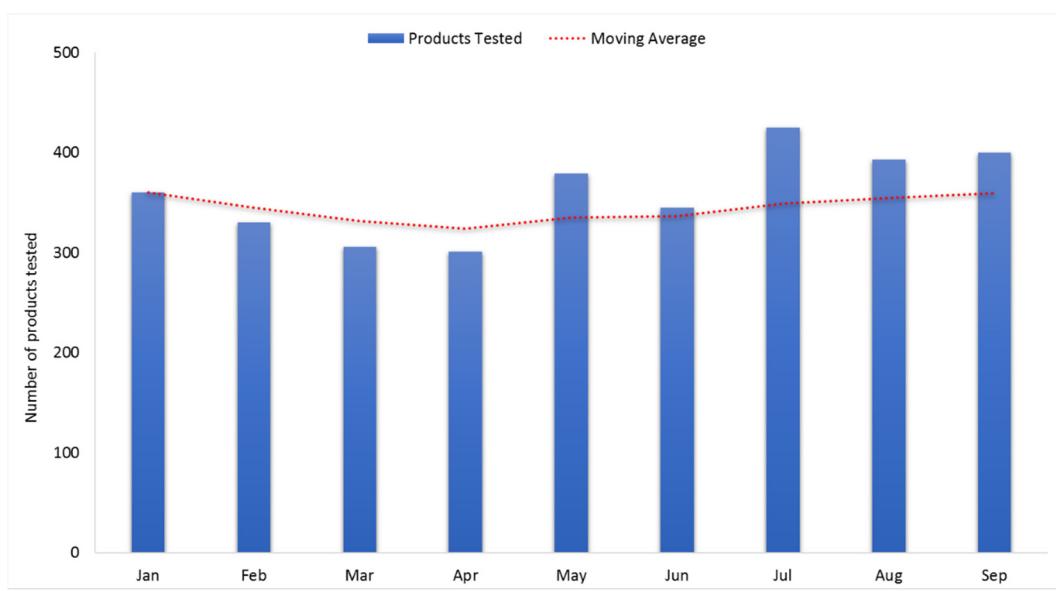
**Figure 17.** Customer wise operator performance—Q1 2019.

### 5.7. Key Performance Indicators (KPIs)

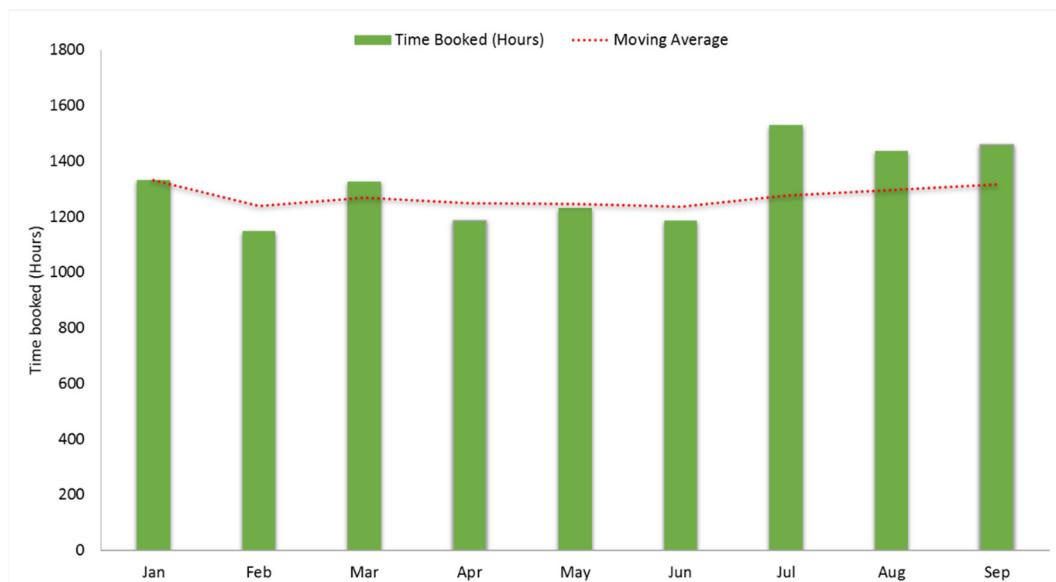
Manufacturing companies present their performance via KPIs. These KPIs highlight the problems and shortcomings as well as show areas where improvements are made. In [47], the authors discussed and highlighted the importance of KPI. In [48], the authors defined KPIs to monitor performance. This is a standard method of presenting the performance of any system. Using the proposed system, the manufacturing companies can present their results via four KPIs, which are performance, capacity, distribution, and availability. In this article, 2 separate graphs for the KPI “Capacity” are presented for monthly electronic products tested and monthly test time booked. The results are for the first three quarters of the year, where months are listed on the *x*-axis.

In Figure 18, the number of monthly electronic products tested are shown via bar charts. The moving average is also shown in the graph. It can be seen from the graph that fewer products were tested during February, March, and April. The number of products tested depends on the customer orders booked and staff holidays. Using this KPI, companies can plan what resources are required in the future. Here, the results are presented monthly but can be changed as required.

In Figure 19, the monthly time booked and the moving average is shown. Manufacturing companies at the time of receiving customer orders, select an estimated test time for the UUT. This estimated test time is used as a benchmark for checking capacity and performance. Manufacturing companies can also set a target threshold and can compare this with moving average plots.



**Figure 18.** KPI (key performance indicator)—Capacity (monthly electronic products tested).



**Figure 19.** KPI—Capacity (monthly test time booked).

### 5.8. Machine-Learning-Based Automatic Recommendations

In this section, machine-learning-based automatic recommendation feature is validated through test cases. At the end of testing, UUT test results are stored in the database, and in case of failure, a fault is also entered by test software automatically. TO can also enter any observation, etc., as free text using the interface. The following four recommendations are automatically generated by the proposed system.

#### 5.8.1. Recommendation 1

In the first test case, test results of some failed UUTs are analyzed by the test software. It is found that the first 5 UUTs of a batch failed, but after that, the UUTs continued to PASS the test. The machine-learning algorithm, to come up with a solution, review the fault, which is automatically entered by the test software, and found that a connector J5 is not making proper contact. It is recommended that the test jig is faulty, so we need to perform

maintenance and replace connector J5. It would have taken a lot of time to review this fault manually, but using machine learning, this is completed quickly.

#### 5.8.2. Recommendation 2

In the second test case, TO reported through the text box that, although the UUT PASS the test but, one component on the test jig is getting hot. The machine-learning algorithm search the comments automatically entered by test software and the text entered by TO. As all the UUTs PASS the test, so no software entered comments are found. In the next iteration, the machine-learning application search comments entered by TO and found the details of the components getting hot and recommended to modify the test jig and use heatsink.

#### 5.8.3. Recommendation 3

In the third test case, the machine-learning algorithm review test results of a UUT batch. It is found that one TO took 5 minutes average test time per UUT, while the second TO took 9 minutes average test time per UUT. This is a significant difference, where the second TO took almost twice the time. The machine-learning algorithm recommend that the first TO should be used when testing this specific UUT, and the second TO requires training before testing this product again.

#### 5.8.4. Recommendation 4

In the final test case, the machine-learning algorithm review the average test time for a UUT. It is found that this is on average 3 minutes more than the estimated test time, i.e., test time per UUT agreed with the customer. Due to this, the CEM company is losing revenue. The automated recommendation, in this case, is to discuss the issue with the customer and propose a new test time.

### 6. Discussion

Collecting quality test data can help the CEM companies to understand the areas where improvement is required or the areas that are creating a bottleneck. These companies can take actions such as redesigning their test jigs, changing the test sequence, adding or removing certain tests, specific and focused training for staff and have a clear idea regarding recruitment, investment in test equipment, etc., [49]. Based on the review, several limitations are identified in the existing systems used within the CEP test domain of the CEM industry.

A new CEP is normally required to be tested as part of the manufacturing process. A test site is required for this purpose, which includes, COTS test equipment, hardware interfaces, test jigs, test software, etc. Due to a lack of a universal system, the CEM companies are required to carry out all the activities for setting up a test site, which takes a lot of time and effort. The proposed universal system provides a solution for this and saves time and cost for the CEM companies, while maintaining consistency and improving quality. To automate testing a test software is also required to be developed, which can take a lot of time depending on the complexity of the UUT. The next issue is the validation of this software application, often it is observed that due to lack of time and urgency to market the product, proper software validation is ignored, which means a faulty product is shipped. Using the proposed universal system, the CEM companies are not required to develop the test application for every new product and just need to integrate their test site. Having a universal software application for testing various CEPs is an important step, but this takes us to the next problem, which is to identify and maintain a library of software sub-routines for test measurements. Due to a variety of UUTs, the type of test measurements also varies depending on COTS or in-house developed test equipment.

The next challenge is to have a process or approach that can automatically decide, based on the input from the TO, what test measurements are required and how these measurements should be taken based on the UUT. A machine-learning approach can

resolve this problem, but for a supervised machine-learning technique to work efficiently, a lot of historical data including test code for taking test measurements is required. To solve this issue, around 100 test sites are reviewed for a variety of CEPs. This helped in creating a knowledgebase for COTS test equipment, UUT hardware interfaces, and test measurements. The system is designed in such a way that new device drivers and measurement code can be easily added. The test software takes input from the TO, selects the required code from the database, finalizes the test sequence, and the test software is generated quickly. Finally, for the above concept to work, a universal hardware interface is required. This hardware interface should have all the common interfaces to connect to any electronic product test site. This also requires a library of COTS and other test equipment drivers for the proposed system to work. Limitations and features of these interfaces such as data rate, number of devices that can be connected, power requirements are also considered. A vast majority of the COTS test equipment has one of the control ports used in the universal hardware interface.

It should be noted that hundreds and thousands of UUT are tested monthly, and it is not possible to keep track and resolve failures through a manual failure analysis process. The other issue is to categorize the failures so that the relevant department is provided with details so that they can find a solution. The automated process presented here is based on a learning dataset and the faults and repair information is collected and placed into the required category. This process will speed up the fault analysis process with minimum effort. An important aspect of this research is to make sure that the proposed system can be integrated with the existing systems used in the CEM industry. Some CEM companies use the ERP systems; therefore, an interface between the proposed and the ERP systems is proposed. This is a difficult task, as the ERP systems are developed using different technologies and a universal interface requires studying different ERP systems, their interfaces, and plugins. As part of this research, several commonly used ERP systems in the CEM industry are reviewed and an interface is created. Other CEM companies use standalone systems in their test departments, so it is also considered, and the proposed system can also be deployed to work as a standalone system.

The proposed system can also be used outside the CEM industry to validate experimental setups for testing electronic circuits and systems. The researchers can interface COTS test equipment and UUT hardware to connect to their circuits and systems. The universal test software can then be used to automatically control and evaluate their designs. The proposed system is validated in two stages. Firstly, the validation of universal test hardware interface and universal test data collection sub-systems is done using two experimental test sites. These test sites are used for testing two different product types. This shows that the proposed system can be used to test a variety of CEPs. Secondly, the data analysis sub-system is validated by deploying the proposed system in a CEM environment where data are collected and analyzed, and results are presented in the form of graphs, KPIs, and machine-learning-based recommendations. The proposed system has been designed to cater to the above-mentioned issues using a user-friendly and low-cost approach.

## 7. Conclusions

A complete solution is provided in this paper for CEP testing that includes a universal hardware interface and a software application based on machine-learning. The proposed system provides COTS test equipment control, test data collection, storage, analysis, indicates KPIs, and automatically generates recommendations. The automated system can be deployed as a standalone system as well as an alternative to ERP systems in the test departments of the CEM industry. Additionally, the proposed system can be used to connect any electronic product test site through a combination of universal hardware and machine-learning-based software subsystems. This approach saves cost and time as no additional hardware interface design and test software development is required for each product. Moreover, the system is user-friendly, flexible and requires less training before

implementation. The proposed system is validated through experimental test sites of two products, a multi-channel RF amplifier, and an analog voice recorder. However, the data processing sub-system is validated by deploying it in a mid-volume CEM environment, and results are presented in graphical form. Finally, a dataset is created, and faults are categorized for the machine-learning-based recommendations and results are presented through examples to streamline the processes, improve the testing mechanism, and overcome the failures for good quality products.

In the future, the dataset size in the database can be increased by adding more test site details that will be helpful for the machine-learning algorithm for further improvement in the decision. Similarly, based on a customer requirement, the approach can be implemented to high volume manufacturing industries with more options.

**Author Contributions:** A.S. conceived and developed the proposed process; programmed its implementation; gathered data; analyzed the results; wrote the manuscript and implemented in a manufacturing environment, M.Y.I.Z. helped with data gathering and programming, debugging the application, and generation of results, P.O. proposed the research direction and helped with the manuscript structure and its revision. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research has been partially funded by Universidad de Málaga, Campus de Excelencia Internacional Andalucía Tech., Málaga, Spain.

**Acknowledgments:** Authors express their gratitude to the Escuela Técnica Superior de Ingeniería de Telecomunicación, and the Instituto de Ingeniería Oceánica, University of Málaga, Málaga, Spain.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Stark, J. *Product Lifecycle Management*; Springer Science and Business Media LLC.: Cham, Switzerland, 2016; pp. 1–35. [[CrossRef](#)]
2. Brusa, E.; Cala, A.; Ferretto, D. System Verification and Validation (V&V). *Emerg. Trends Sliding Mode Control* **2017**, *134*, 289–325. [[CrossRef](#)]
3. Yu, B.; Xu, X.; Roy, S.; Lin, Y.; Ou, J.; Pan, D.Z. Design for manufacturability and reliability in extreme-scaling VLSI. *Sci. China Inf. Sci.* **2016**, *59*, 1–23. [[CrossRef](#)]
4. Ungar, L.Y. Design for Testability (DFT) to Overcome Functional Board Test Complexities in Manufacturing Test. In Proceedings of the IPC APEX 2017, San Diego, CA, USA, 14–16 February 2017.
5. Burkhardt, A.; Berryman, S.; Brio, A.; Ferkau, S.; Hubner, G.; Lynch, K.; Mittman, S.; Sonderer, K. Measuring Manufacturing Test Data Analysis Quality. *2018 IEEE Autotestcon* **2018**, 1–6. [[CrossRef](#)]
6. Jain, S.; Shao, G.; Shin, S.-J. Manufacturing data analytics using a virtual factory representation. *Int. J. Prod. Res.* **2017**, *55*, 5450–5464. [[CrossRef](#)]
7. Liang, A.; Zhanyong, R. Research on Determination Method of Electronic Equipment Incoming Defect in Batch Production. In Proceedings of the 2017 International Conference on Computer Systems, Electronics and Control (ICCSEC), Dalian, China, 25–27 December 2017; Volume 2018, pp. 396–399. [[CrossRef](#)]
8. Kesim, H. Automated continuity testing of flexible backplanes using a cable tester. *2015 IEEE Autotestcon* **2015**, 269–272. [[CrossRef](#)]
9. Angrisani, L.; Ianniello, G.; Stellato, A. Cloud based system for measurement data management in large scale electronic production. In Proceedings of the 2014 Euro Med Telco Conference (EMTC), Naples, Italy, 12–15 November 2014; pp. 1–4. [[CrossRef](#)]
10. Sangat, P.; Taniar, D.; Indrawan-Santiago, M. Sensor data management in the cloud: Data storage, data ingestion, and data retrieval. *Concurr. Comput. Pr. Exp.* **2018**, *30*, e4354. [[CrossRef](#)]
11. Saez, M.; Lengieza, S.; Maturana, F.; Barton, K.; Tilbury, D. A Data Transformation Adapter for Smart Manufacturing Systems with Edge and Cloud Computing Capabilities. In Proceedings of the 2018 IEEE International Conference on Electro/Information Technology (EIT), Rochester, MI, USA, 3–5 May 2018. [[CrossRef](#)]
12. Business Management Software to Fit Your Industry | Epicor U.S. Available online: <https://www.epicor.com/en-us/> (accessed on 31 October 2020).
13. Enterprise Resource Planning | ERP for Business Software | ERP UK. Available online: <https://eu.syspro.com/> (accessed on 31 October 2020).
14. CRM and ERP Applications | Microsoft Dynamics 365. Available online: <https://dynamics.microsoft.com/en-gb/> (accessed on 31 October 2020).
15. Cloud ERP—Sage Business | Sage UK. Available online: <https://www.sage.com/en-gb/sage-business-cloud/sage-x3/> (accessed on 31 October 2020).

16. SAP Business ByDesign | Cloud ERP Software | Sapphire Systems. Available online: <https://www.sapphiresystems.com/en-gb/products/sap-business-bydesign> (accessed on 31 October 2020).
17. Enterprise Resource Planning (ERP) | Oracle. Available online: <https://www.oracle.com/erp/> (accessed on 31 October 2020).
18. Rouhani, S.; Mehri, M. Empowering benefits of ERP systems implementation: Empirical study of industrial firms. *J. Syst. Inf. Technol.* **2018**, *20*, 54–72. [CrossRef]
19. Buergin, J.; Belkadi, F.; Hupays, C.; Gupta, R.K.; Bitte, F.; Lanza, G.; Bernard, A. A modular-based approach for Just-In-Time Specification of customer orders in the aircraft manufacturing industry. *CIRP J. Manuf. Sci. Technol.* **2018**, *21*, 61–74. [CrossRef]
20. Berryman, S.; Brio, A.; Burkhardt, A.; Ferkau, S.; Gharbiah, H.; Hubner, G.; Lynch, K.; Woudenberg, M. Concept of operations for test cost analytics in complex manufacturing environments. *2017 IEEE Autotestcon 2017*, 1–8. [CrossRef]
21. Dorochowicz, A.; Kurowski, A.; Kostek, B. Employing Subjective Tests and Deep Learning for Discovering the Relationship between Personality Types and Preferred Music Genres. *Electronics* **2020**, *9*, 2016. [CrossRef]
22. Horng, M.-F.; Kung, H.-Y.; Chen, C.-H.; Hwang, F. Deep Learning Applications with Practical Measured Results in Electronics Industries. *Electronics* **2020**, *9*, 501. [CrossRef]
23. Sapkota, S.; Mehdy, A.K.M.N.; Reese, S.; Mehrpouyan, H. FALCON: Framework for Anomaly Detection in Industrial Control Systems. *Electronics* **2020**, *9*, 1192. [CrossRef]
24. Kang, Z.; Catal, C.; Tekinerdogan, B. Machine learning applications in production lines: A systematic literature review. *Comput. Ind. Eng.* **2020**, *149*, 106773. [CrossRef]
25. Martinek, P.; Krammer, O. Analysing machine learning techniques for predicting the hole-filling in pin-in-paste technology. *Comput. Ind. Eng.* **2019**, *136*, 187–194. [CrossRef]
26. Šerban, M.; Vagapov, Y.; Chen, Z.; Holme, R.; Lupin, S. Universal platform for PCB functional testing. In Proceedings of the 2014 International Conference on Actual Problems of Electron Devices Engineering (APEDE), Saratov, Russia, 25–26 September 2014; Volume 2, pp. 402–409. [CrossRef]
27. Hanley, L. ChiMS: Open-source instrument control software platform on LabVIEW for imaging/depth profiling mass spectrometers. *Rev. Sci. Instrum.* **2015**, *86*, 065106. [CrossRef]
28. Yan, D.; Yang, Y.; Hong, Y.; Liang, T.; Yao, Z.; Chen, X.; Xiong, J. Low-Cost Wireless Temperature Measurement: Design, Manufacture, and Testing of a PCB-Based Wireless Passive Temperature Sensor. *Sensors* **2018**, *18*, 532. [CrossRef]
29. Chavhan, K.B.; Ugale, R. Automated test bench for an induction motor using LabVIEW. In Proceedings of the 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), Delhi, India, 4–6 July 2016; pp. 1–6. [CrossRef]
30. Hakim, A.; Khayam, U. Simulation and testing of Goubau PCB antenna as partial discharge detector. In Proceedings of the 2017 International Conference on High Voltage Engineering and Power Systems (ICHVEPS), Bali, Indonesia, 2–5 October 2017; pp. 170–174. [CrossRef]
31. Khayam, U.; Alfaruq, F. Design of Hilbert antenna as partial discharge sensor. In Proceedings of the 2016 2nd International Conference of Industrial, Mechanical, Electrical, and Chemical Engineering (ICIMECE), Yogyakarta, Indonesia, 6–7 October 2016; pp. 84–88. [CrossRef]
32. Gruwell, A.; Zabriskie, P.; Wirthlin, M. High-Speed FPGA Configuration and Testing through JTAG. In Proceedings of the 2016 IEEE AUTOTESTCON, Anaheim, CA, USA, 12–15 September 2016; pp. 1–8. [CrossRef]
33. Ramaprasad, S.S.; Rajesh, G.N.; Kumar, K.N.S.; Prasad, P.R. Fully Automated PCB testing and analysis of SIM Module for Aircrafts. In Proceedings of the 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, India, 18–19 May 2018; pp. 2016–2020. [CrossRef]
34. Pereira, G.; Puodzius, C.; Barreto, P.S. Shorter hash-based signatures. *J. Syst. Softw.* **2016**, *116*, 95–100. [CrossRef]
35. Engineer Ambitiously—NI. Available online: <https://www.ni.com/en-gb.html> (accessed on 31 October 2020).
36. Zia, M.Y.I.; Otero, P.; Siddiqui, A.; Poncela, J. Design of a Web Based Underwater Acoustic Communication Testbed and Simulation Platform. *Wirel. Pers. Commun.* **2020**, *1*–23. [CrossRef]
37. Manimozhi, A.; Nivetha, D.; Nivethitha, P. Smart Environmental Monitoring System Using Labview. *Int. J. Eng. Comput. Sci.* **2017**, *6*, 20705–20709. [CrossRef]
38. Mahmoodi, M.; James, L.; Johansen, T. Automated advanced image processing for micromodel flow experiments; an application using labVIEW. *J. Pet. Sci. Eng.* **2018**, *167*, 829–843. [CrossRef]
39. Khan, A.S.; Rajkumar, R.K.; Aravind, C.V.; Wong, Y.W.; Bin Romli, M.I.F. A LabVIEW-based Real-Time GUI for Switched Controlled Energy Harvesting Circuit for Low Voltage Application. *IETE J. Res.* **2018**, *66*, 720–730. [CrossRef]
40. Mishra, D.; Gupta, A.; Raj, P.; Kumar, A.; Anwer, S.; Pal, S.K.; Chakravarty, D.; Pal, S.; Chakravarty, T.; Pal, A.; et al. Real time monitoring and control of friction stir welding process using multiple sensors. *CIRP J. Manuf. Sci. Technol.* **2020**, *30*, 1–11. [CrossRef]
41. Sahu, G.; Vashisht, S.; Wahi, P.; Law, M. Validation of a hardware-in-the-loop simulator for investigating and actively damping regenerative chatter in orthogonal cutting. *CIRP J. Manuf. Sci. Technol.* **2020**, *29*, 115–129. [CrossRef]
42. Miranda, J.; Ponce, P.; Molina, A.; Wright, P. Sensing, smart and sustainable technologies for Agri-Food 4.0. *Comput. Ind.* **2019**, *108*, 21–36. [CrossRef]
43. Radcliffe, J.; Cox, J.; Bulanon, D.M. Machine vision for orchard navigation. *Comput. Ind.* **2018**, *98*, 165–171. [CrossRef]
44. Welcome to Python. Available online: <https://www.python.org/> (accessed on 31 October 2020).

45. Binder, J.M.; Stark, A.; Tomek, N.; Scheuer, J.; Frank, F.; Jahnke, K.D.; Muller, C.; Schmitt, S.; Metsch, M.H.; Unden, T.; et al. Qudi: A modular python suite for experiment control and data processing. *SoftwareX* **2017**, *6*, 85–90. [[CrossRef](#)]
46. Vanderplas, J.T.; Granger, B.; Heer, J.; Moritz, D.; Wongsuphasawat, K.; Satyanarayan, A.; Lees, E.; Timofeev, I.; Welsh, B.; Sievert, S. Altair: Interactive Statistical Visualizations for Python. *J. Open Source Softw.* **2018**, *3*. [[CrossRef](#)]
47. Lindberg, C.-F.; Tan, S.; Yan, J.; Starfelt, F. Key Performance Indicators Improve Industrial Performance. *Energy Procedia* **2015**, *75*, 1785–1790. [[CrossRef](#)]
48. Belkadi, F.; Boli, N.; Usatorre, L.; Maleki, E.; Alexopoulos, K.; Bernard, A.; Mourtzis, D. A knowledge-based collaborative platform for PSS design and production. *CIRP J. Manuf. Sci. Technol.* **2020**, *29*, 220–231. [[CrossRef](#)]
49. Vogl, G.W.; Weiss, B.A.; Helu, M. A review of diagnostic and prognostic capabilities and best practices for manufacturing. *J. Intell. Manuf.* **2019**, *30*, 79–95. [[CrossRef](#)]