Reconfigurable Manufacturing: Towards an industrial Big Data approach

1st Halldor Arnarson
Department of Industrial Engineering
UiT The Arctic University of Norway
Narvik, Norway
halldor.arnarson@uit.no

2nd Bernt Arild Bremdal
Department of Computer Science and Computational Engineering
UiT The Arctic University of Norway
Narvik, Norway
bernt.a.bremdal@uit.no

3th Bjørn Solvang
Department of Industrial Engineering
UiT The Arctic University of Norway
Narvik, Norway
bjorn.solvang@uit.no

***Abstract*—As the world is moving towards more personalized and customized manufacturing, the manufacturing system needs to adapt. One method can be to integrate industry 4.0 concepts in reconfigurable manufacturing systems (RMS). This allows the manufacturing system to become more self-sustaining and flexible at the same time. There is however, a lack of research on how to integrate industry 4.0 concepts such as industrial Big Data (IBD) into RMS. This paper looks at how IBD techniques can be used on an RMS for classification and how to collect data from an RMS. A case study where five different movable platforms are identified with an accuracy of more than 85% is showcased.**

***Index Terms*—Reconfigurable Manufacturing System (RMS), Industrial Big Data (IBD)**

I. Introduction

With globalization, manufacturing companies are experiencing more fluctuations in product demand and unpredictable market changes [1]. At the same time, the manufacturing industry is moving towards more personalized production, which requires more frequent changes to the manufacturing system.

One approach to handle these changes is to have a reconfigurable manufacturing system (RMS). RMS is built on modularity and can rearrange itself based on functionality and production capacity. It provides a system that can quickly adapt to changes in the market at a reasonable cost.

RMS has the same high throughput as dedicated manufacturing lines (DML) and the agility of flexible manufacturing systems (FMS) while also being able to respond to changes in the market [2, 3]. It has been shown that taking DML and adding reconfigurability can give considerable capacity savings [4].

We are still in the early stages of research on RMS and how to implement RMS [5]. It has been found that there is a need for more research on reconfigurability towards industry 4.0 and how to use the industry 4.0 technologies in RMS [6].

Industry 4.0 brings new technologies such as simulation, autonomous robots, industrial internet of things (IIoT), cloud, cybersecurity, additive manufacturing, and Big Data (BD) and analytics [7]. These technologies enable manufacturing systems to be self-learning, controlling and aware. Industry 4.0 technologies are fundamental for the success of RMS [8], making the system more intelligent and self-sustaining.

One concept from industry 4.0 that can allow an RMS to become smarter is BD and data analysis [8]. BD for an RMS can be used to store and gather historical data, trends and status of the system and machines [8].

BD can be used to manage large amounts of data in an efficient way [9]. The emergence of the internet has led to a large amount of data being collected that can not be handled with traditional tools for analysis and processing. Data collected from machines, the environment, humans and manufacturing processes can help improve the product quality and reduce production costs [10].

BD is a term often used to describe data that is challenging to manage with traditional tools [11]. It can be defined as, *"Big Data is the Information asset characterised by such a High Volume, Velocity and Variety to require specific Technology and Analytical Methods for its transformation into Value"* [12].

BD in industry 4.0 applications can be used for business intelligence, product quality enhancement, machine health prediction, production planning and fault tolerance [11]. Combining BD techniques with a cyber physical manufacturing system can allow for monitoring of the system and real-time decisions making for production scheduling [13]. It is also important to note that BD is not only related to the volume of the data but also the velocity, variety and veracity [14].

There are differences in how BD is used in different industries. For example Industrial Big Data (IBD) requires other processing techniques than for BD in social networks [15]. IBD is related to machine generated data instead of human related data and can be collected from machine controllers, manufacturing systems and sensors [16].

As mentioned before, RMS is built on modularity and can contain multiple platforms that are working together. In this paper we will define IBD for RMS, as data collected and analyzed from the whole manufacturing system. This includes multiple platforms, machines, sensors, logistics systems, ERP system and so on. Collecting and analyzing data from one platform or machine can be considered small data, but collecting and using data from multiple platforms and machines is IBD.

This paper will look at how IBD can be applied to an RMS and have the following contributions:

- We will look at how IBD can be collected from multiple platforms in an RMS.
- Look at how the data from a mobile robot that is part of an RMS can be used with IBD techniques and how the volume, velocity, variety, and veracity effects the usability of the data.

The remainder of this paper can be structured as follows. Section II explains how the system is set up and it is connected. Section III goes through the four BD processing where the data is collected and how the data can be used to predict which platform is being moved. Discussion and conclusion in section IV and V.

II. Data processing

This section look at the system setup, how it is connected together.

A. System setup

An RMS often consists of different modules that can be assembled together based on a specific manufacturing task. For our experiments we have built a modular RMS of the following components:

- **Nachi platform:** A small anthropomorphic industrial robot
- **Scara platform:** A four joint SCARA robot
- **Conveyor lift:** A conveyor that can be moved up and down
- **Conveyor platform:** A normal conveyor to transport goods
- **3D print platform:** A creality CR-30 3D printer that can automatically print parts and move them out automatically.

These platforms can be moved and reassembled automatically with a mobile robot (MiR100). The mobile robot is equipped with a top module that has two pins that can move out to fasten itself to the platforms. It has a max payload of 100 kg but a towing capacity of 300 kg [17]. In this system, the platforms are being towed using the docking module.

A video on how the system works has been created and can be found https://youtu.be/idA-TmYP45c. The Video showcases the mobile robot picking up the five different platforms from various positions in the laboratory and assembling them to a manufacturing system. Afterward, the mobile robot takes the system apart and builds a new manufacturing system in front of the vertical storage lift (Compact lift).

B. Connection

This system consists of robots and other machines from different brands such as Nachi, MiR and Adept. These machines have different communication standards and it is time-consuming and tedious to create a separate method to collect data from each machine separately.

Another approach to collect data from the robots is to connect all machines to the Open Platform Communications Unified Architecture (OPC UA) server. The OPC UA is an IEC 62541 standard [18] and is widely accepted and used for industrial communications systems. It is seen as a reliable and secure standard for data exchange between components [19]. Connecting machines to the OPC UA server allows for a standardized method of communication and control.

The large conveyor, conveyor lift and 3D print platform use raspberry pi for control while the Scara and Nachi platforms have ubuntu machines. All these computers connect wirelessly to the OPC UA server, update data from the platforms to the OPC UA server and take data from the OPC UA server. This method allows for data collecting and monitoring while the system is running. Figure 1 shows how the system is connected with the OPC UA server.

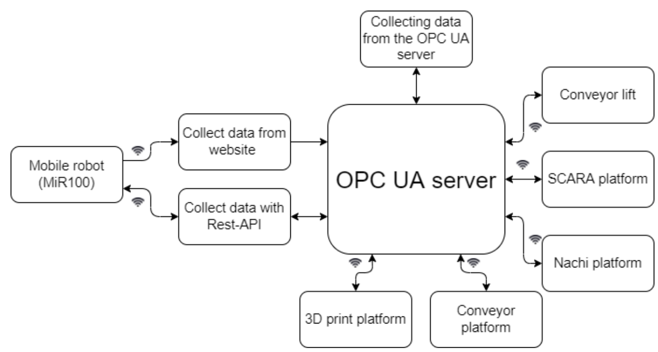

Fig. 1. The figure shows how all platforms and the mobile robot is connected to the OPC UA server.

C. Mobile robot connection

After all the platforms have been connected to the OPC UA server the mobile robot needs to be connected as well.

The mobile robot has a REST-API which can be used for simple control of the robot, gather information and changing settings on the robot. This interface uses JSON messages with get, post, and put messages. A python program is created, which works as a bridge between the OPC UA server and mobile robot. The program takes data from the mobile robot, sends it to the OPC UA server and takes control data from the OPC UA server, sends it to the mobile robot.

The REST-API gives somewhat limited information on the mobile robot and it is not possible to get data directly from the motors, power system, or sensors. If the payload is too heavy, the mobile robot will stop and give the operator the error message “motor stall detected!” or “power limit”. Thus, it can be beneficial to get data directly from motors, the power system, and various sensor data.

It is possible to get more data on the mobile robot by using a web browser and connecting to the IP address of the robot. Under "Monitoring" and then "Hardware health" you can get data on the internal Computer, Motors, Power system, Safety system, Sensors and Serial interface.

The data displayed in the browser is updated automatically every second. One method to collect this data is by using web scraping. Web scraping is a technique used to collect data from websites and save them to databases or files [20]. It can be used to extract data from websites automatically.

A second python program is created to collect the data from the web browser and put it in the OPC UA server. The program uses Selenium [21] which opens a chrome browser and goes to the IP address of the mobile robot. Then opens the "Hardware health" pages and extends all the windows to show the information from the mobile robot.

From the "Hardware health" page, there are 237 variables. For the sake of simplicity, saving time and processing power, only a set number of variables were chosen and limited to 84 variables. These variables are constantly read from the website and updated to the OPC UA server. The variables collected were on the motors, power system and sensors. A screenshot from the data in the OPC UA server can be seen in figure 2.

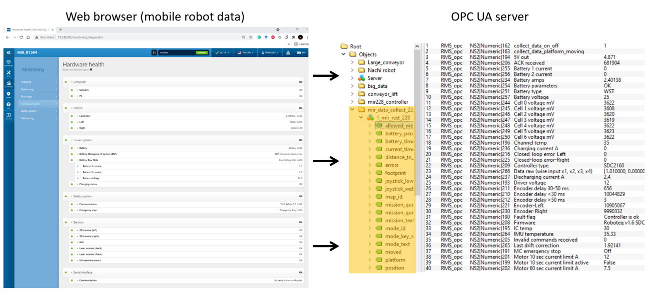

Fig. 2. The figure showcases the data taken from the web browser and transferred to the OPC UA server. The program UA Expert [22] is used to view OPC UA server, where the yellow square shows the data from the browser.

D. Data characteristics

When the data has been collected and processed it can be analyzed by looking at the characteristics. In the case of BD, it can be divided into four main characteristics volume, velocity, variety and veracity [14] [23]. Where:

- **Volume:** Is related to the amount of data/information being gathered. There is a need for algorithms that can handle and process large amounts of data in real time [14].
- **Velocity:** Is related to the speed at which the data is created, gathered and streamed [24].
- **Variety:** Data variety is related to the different data sources where the data is collected [24]. In industrial manufacturing systems, this often relates data from sensors, machines and other manufacturing equipment.
- **Veracity:** Is related to the quality accuracy and correctness of the data collected [25].

To evaluate the data from the mobile robot it can be beneficial to have the update rate and size of the data coming from the mobile robot. A simple python program is created, that measures how often the variable pitch from the gyroscope and the current of the left motor was updated. The data is updated every second. The data size collected was also measured, for all the 237 variables the size is around 2208 bytes.

When relating these four characteristics to the mobile robot:

- **Volume:** With the 237 variables, the data collected when the system is running is 2208 bytes per second. This can be considered as low volume and since this is data collection is from one module in an RMS, it will fall under small data instead of IBD. It should be noted that having multiple mobile robots or platforms would then create IBD.
- **Velocity:** The velocity of the data is rather slow (ones a second) and there is a chance of missing important readouts from the sensors. This can also affect the response rate from the sensors and it is more beneficial to have a higher update rate of the data.
- **Variety:** A total of 237 variables can be collected from the mobile robot on the internal Computer, Motors, Power system, Safety system, Sensors, and Serial interface. There is data on almost all components in the mobile robot. This data gives detailed insight into what the mobile robot is doing and the status of the mobile robot.
- **Veracity:** As mentioned, the data from the mobile robot isn't made readily available by using their REST-API. There is no information on how the data is collected, transported, and the update rate of different data. It is hard to say something about the accuracy or correctness of the data, but a low update rate (velocity) can affect the data quality. With low velocity of the data the system can miss readouts from the sensors, which reduces the

III. Case Study

After setting up and connecting the system, the next step is collecting the data and using it to do classifications. This paper focuses on data collection for the mobile robot since it is used to reconfigure the system. The experiment is conducted in a controlled environment and only focuses on one platform. Thus, it will therefore fall under small data and not IBD. Regardless, we can still use the IBD techniques and see how the main characteristics of IBD can affect the data's usability in making predictions.

A. Case study/idea behind the system

As mentioned before, the mobile robot is used to move the platforms around in the laboratory. All platforms in the system have a marker in front which the 3D cameras can detect and the marker allows the mobile robot to position itself accurately relative to the marker. However since all the platforms are on wheels, they can easily be tilted or moved out of position (e.g by a human operator). If a platform is moved by someone other than the mobile robot, the system will loose the position of the platform. Thus, there is a scenario where the mobile robot is driving around looking for platforms that have been moved out of its predicted position. Now the idea is to collect and use BD from the motors, power system and other sensors in the mobile robot in order to identify which platform is found. All platforms in the system vary in weight, size, wheel diameter etc. and requires an unique set of forces in order to be moved around.

In this paper six cases are considered, one for each of the five platforms and a sixth when there is no platform attached. The goal is to use data from the motors, battery and sensors with machine learning to classify what platform is being moved.

B. *Big data processing*

BD processing can be divided into four main steps; data collection, data preprocessing, data storage and data analysis [24].

- **Data collection:** Data can come from different sensors or other devices that are connected up to the internet. Therefore the first step is to collect the data from various sources.
- **Data storage:** The data that is collected needs to be stored. With BD, the data comes from different sources and is often diverse. It might need software that is compatible with multiple data types.
- **Data preprocessing:** Is used to clean and process the data that has already been collected. Some of the data that is collected might be invalid and needs to be removed. Other data needs to be unified and structured so that it can be analyzed with data from different sources.
- **Data analysis:** The last step is data analysis. Analyzing BD can give information and insight into processes that only BD analytics can give.

These four steps are used to process the data from the mobile robot.

C. *Data collecting and storage*

Machine learning can be used for various types of classification, but it needs historical data to train/fit a machine learning algorithm. In this case, data needs to be collected while the system is running.

To collect data, a program is created that drives the mobile robot to a random position and orientation in the laboratory. The area where the robot is driving is marked in figure 3.

Data is collected while the mobile robot drives towards 30 random positions. When the mobile robot is done, the data is stored in a CSV file. The program is executed for all five platforms and then with no platform. The same data is collected a second time with 30 random positions. This data is used to validate and see if the classifier works.

A short video showcasing data collection from all platforms can be found at https://youtu.be/0Sy457WWHbM.

As can be seen in figure 3, two spots are marked with a red square. If the mobile robot drives over these areas, there is a high likelihood that the robot will get stuck because of

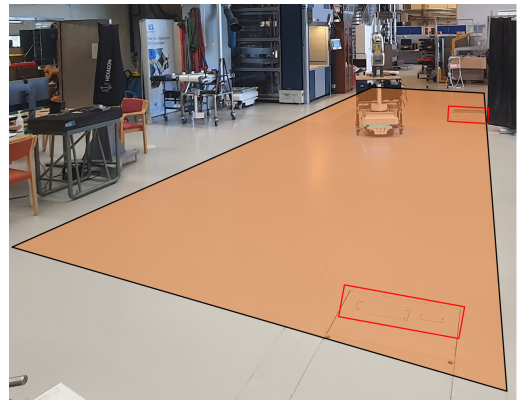

Fig. 3. Laboratory training arena: Collecting data for learning algoritm.

the pumps. If it gets stuck, it will generate the error “Right motor stall detected!”, “Left motor stall detected!” or “Motor power usage above limit!”. When the mobile robot gets stuck, the data collection needs to be restarted.

An example from the variable “Battery 1 current” can be seen in figure 4.

From the figure 4, there is a clear difference when the mobile robot is not moving a platform and when it’s moving a platform. It can be harder to see clear differences between the platforms.

D. *Data preprocessing and analysis*

To train/fit the machine learning model the python library scikit-learn [26] is used. Scikit-learn is a well known machine learning library for python [27]. It has different machine learning algorithms for classification, such as naive Bayes classifier, k-nearest neighbors and linear support vector classification.

K-nearest neighbors is a simple and popular machine learning algorithm to do classification [28]. The algorithm can classify objects using the training data to find the nearest new object based on euclidean formula [29].

In this case, the k-nearest neighbor’s method is used because of its simplicity and ease of use.

It is beneficial to have clear data, where it is possible to detect patterns. For instance, when the robot has arrived at its position, it stops and plans a new route before continuing. When the robot stops, the current of the motors and batteries becomes the same for all platforms. These measurements is removed to make the data more clear. After trimming the data, there are around 550 to 850 measurements from each of the 84 variables.

It is possible to get the current of the motors and the discharging rate of the battery. In addition, it is possible to get data on the current of both batteries and the channel temperature of the motor controller. These variables can be used to identify if the mobile robot is pulling heavy platforms.

Then from the 84 variables, six were used to train the classifier, as listed below:

- Battery 1 current
- Battery 2 current

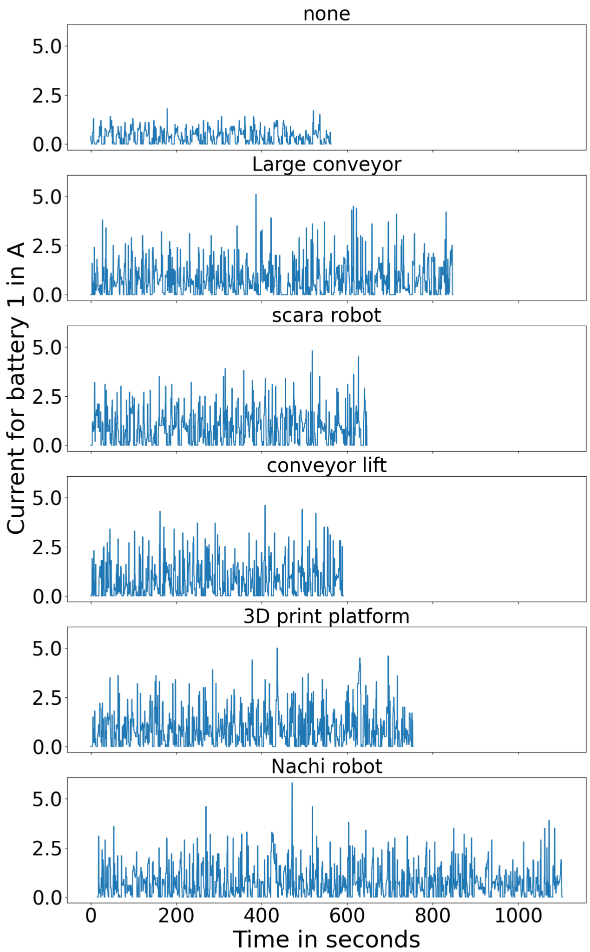

Fig. 4. The graph shows a example of data collected on for the variable "Battery 1 current". It shows data from all five platforms and where there no platform (none).

- Discharging current A
- Motor current-Left
- Motor current-Right
- Channel temp (of motor controller)

After testing different combinations of the six variables, it was found that using the "Channel temp" and "Discharging current A" gave the best classification (80-85% accuracy). It is also possible to use the battery current or motor current, giving a little lower accuracy of around 80%.

IV. Discussion

In this paper, we have focused on collecting and analyzing data from the mobile robot in the RMS. Using IBD techniques on the mobile robot gives new opportunities to do classification with the sensors. As mentioned, the experiment with the mobile robot has small data and not IBD. However, the same methods and techniques from the experiment can be transferable to the other modules and data sources. In the literature, there is most focus on the amount of data in IBD cases. However, it is also important to consider the data's velocity, variety, and veracity.

The next step is to look at how all the data from the robot arms, conveyors, 3D printers and other data in the system can be used together. Merging data from different modules and sources in an RMS can allow the system to become more intelligent with the use of IBD.

Using the OPC UA server can be a good method to connect manufacturing equipment, allowing for simple data collection. However, it should be noted that connecting robots and other machines that don't support the OPC UA standard can be time-consuming work. For instance, the mobile robot has valuable data on its website, where a web browser is needed to collect the data. Creating programs for both web scraping and the REST-API for data collection is time consuming. It is more efficient if robot manufacturers make all sensor data available with an API or through the OPC UA server.

The data from the website is only updated every second and gives a low update rate. It can be easier to find patterns between variables if they are collected more frequently. This again makes it more challenging to use for machine learning, where you are trying to find differences between data. With a higher update rate, it can be possible to estimate how much payload the mobile robot is moving. By using the same sensors used to identify which platform is being moved.

In addition, having an update rate of once per second can impact the data quality. When relating this to the four V's, we can see that having a low velocity (update rate) can affect the veracity of the data.

V. Conclusion and further work

This paper looks at different applications of IBD for an RMS in Narvik. It focuses on the sensors data from the mobile robot and how this data can be used.

A classifier is created that uses sensor data from the mobile robot to identify which platform is being moved. Using the two variables "Channel temp" and "Discharging current A" gave the best result (80-85%) with the k-nearest neighbor's classifier. Other variables can also be used, such as the battery current and motor current, giving an accuracy of around 80%.

With the proposed system structure, a classifier can used to identify which platform is being moved.

In further work, we look at how to increase the accuracy of the classification. One method is to create a fixed path/routine for the mobile robot to drive. When the robot is driving the path/routine, data is collected, which can be used to classify which platform is being moved.

It should be noted that both classifiers can be improved by:

- Collected data over a longer period of time. More data usually means a more accurate classifier.
- More machine learning classification algorithms can be tested to improve the classifier.
- Other variables can be tested to improve the classifier.

References

[1] Yoram Koren, Xi Gu, and Weihong Guo. "Reconfigurable manufacturing systems: Principles, design, and future trends". In: *Frontiers of Mechanical Engineering* 13 (Nov. 2017). DOI: 10.1007/s11465-018-0483-0.

[2] Y. Koren et al. “Reconfigurable Manufacturing Systems”. In: *CIRP Annals* 48.2 (1999), pp. 527–540. ISSN: 0007-8506. DOI: https://doi.org/10.1016/S0007-8506(07)63232-6. URL: https://www.sciencedirect.com/science/article/pii/S0007850607632326.

[3] Yoram Koren and Moshe Shpitalni. “Design of reconfigurable manufacturing systems”. In: *Journal of Manufacturing Systems* 29.4 (2010), pp. 130–141. ISSN: 0278-6125. DOI: https://doi.org/10.1016/j.jmsy.2011.01.001. URL: https://www.sciencedirect.com/science/article/pii/S0278612511000021.

[4] Ann-Louise Andersen, Thomas Brunoe, and Kjeld Nielsen. “Investigating the Potential in Reconfigurable Manufacturing: A Case-Study from Danish Industry”. In: Sept. 2015. DOI: 10.1007/978-3-319-22756-6_34.

[5] Rajesh Pansare, Gunjan Yadav, and Madhukar R. Nagare. “Reconfigurable manufacturing system: a systematic review, meta-analysis and future research directions”. In: *Journal of Engineering, Design and Technology* (Aug. 2021). DOI: 10.1108/JEDT-05-2021-0231.

[6] Marco Bortolini, Francesco Gabriele Galizia, and Cristina Mora. “Reconfigurable manufacturing systems: Literature review and research trend”. In: *Journal of Manufacturing Systems* 49 (2018), pp. 93–106. ISSN: 0278-6125. DOI: https://doi.org/10.1016/j.jmsy.2018.09.005. URL: https://www.sciencedirect.com/science/article/pii/S0278612518303650.

[7] Lidong Wang and G. Wang. “Big Data in Cyber-Physical Systems, Digital Manufacturing and Industry 4.0”. In: *International Journal of Engineering and Manufacturing* 6 (2016), pp. 1–8.

[8] Ashutosh Singh, Piyush Gupta, and Mohammad Asjad. *Reconfigurable Manufacturing System (RMS): Accelerate Towards Industries 4.0*. Feb. 2019. DOI: 10.2139/ssrn.3354485.

[9] Krzysztof Witkowski. “Internet of Things, Big Data, Industry 4.0 – Innovative Solutions in Logistics and Supply Chains Management”. In: *Procedia Engineering* 182 (2017). 7th International Conference on Engineering, Project, and Production Management, pp. 763–769. ISSN: 1877-7058. DOI: https://doi.org/10.1016/j.proeng.2017.03.197. URL: https://www.sciencedirect.com/science/article/pii/S1877705817313346.

[10] Jihong Yan et al. “Industrial Big Data in an Industry 4.0 Environment: Challenges, Schemes, and Applications for Predictive Maintenance”. In: *IEEE Access* 5 (2017), pp. 23484–23491. DOI: 10.1109/ACCESS.2017.2765544.

[11] Maqbool Khan et al. “Big Data Challenges and Opportunities in the Hype of Industry 4.0”. In: May 2017. DOI: 10.1109/ICC.2017.7996801.

[12] Andrea De Mauro, Marco Greco, and Michele Grimaldi. “A formal definition of Big Data based on its essential features”. In: *Library Review* 65 (Mar. 2016), pp. 122–135. DOI: 10.1108/LR-06-2015-0061.

[13] Yoram Koren, Xi Gu, and Weihong Guo. “Reconfigurable manufacturing systems: Principles, design, and future trends”. In: *Frontiers of Mechanical Engineering* 13.2 (June 2018), pp. 121–136. ISSN: 2095-0241. DOI: 10.1007/s11465-018-0483-0. URL: https://doi.org/10.1007/s11465-018-0483-0.

[14] Xianyu Zhang, Xinguo Ming, and Dao Yin. “Application of industrial big data for smart manufacturing in product service system based on system engineering using fuzzy DEMATEL”. In: *Journal of Cleaner Production* 265 (2020), p. 121863. ISSN: 0959-6526. DOI: https://doi.org/10.1016/j.jclepro.2020.121863. URL: https://www.sciencedirect.com/science/article/pii/S0959652620319107.

[15] Jihong Yan et al. “Industrial Big Data in an Industry 4.0 Environment: Challenges, Schemes, and Applications for Predictive Maintenance”. In: *IEEE Access* 5 (2017), pp. 23484–23491. DOI: 10.1109/ACCESS.2017.2765544.

[16] Jay Lee, Hung-An Kao, and Shanhu Yang. “Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment”. In: *Procedia CIRP* 16 (2014). Product Services Systems and Value Creation. Proceedings of the 6th CIRP Conference on Industrial Product-Service Systems, pp. 3–8. ISSN: 2212-8271. DOI: https://doi.org/10.1016/j.procir.2014.02.001. URL: https://www.sciencedirect.com/science/article/pii/S2212827114000857.

[17] Mobile Industrial Robots. *MiR100*. URL: https://www.mobile-industrial-robots.com/en/solutions/robots/mir100/. (accessed: 023.11.2020).

[18] OPC Foundation. *Unified Architecture*. URL: https://opcfoundation.org/about/opc-technologies/opc-ua/. (accessed: 07.09.2021).

[19] Peter Drahoš et al. “Trends in industrial communication and OPC UA”. In: *2018 Cybernetics Informatics (K I)*. 2018, pp. 1–5. DOI: 10.1109/CYBERI.2018.8337560.

[20] Bo Zhao. “Web Scraping”. In: May 2017, pp. 1–3. ISBN: 978-3-319-32001-4. DOI: 10.1007/978-3-319-32001-4_483-1.

[21] *selenium*. [Online; accessed 23. Sep. 2021]. Sept. 2021. URL: https://pypi.org/project/selenium.

[22] *UaExpert "UA Reference Client" - Unified Automation*. [Online; accessed 24. Sep. 2021]. Sept. 2021. URL: https://www.unified-automation.com/products/development-tools/uaexpert.html.

[23] Pengcheng Zhang et al. “Data quality in big data processing: Issues, solutions and open problems”. In: *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. 2017, pp. 1–7. DOI: 10.1109/UIC-ATC.2017.8397554.

[24] Pengcheng Zhang et al. “Data quality in big data processing: Issues, solutions and open problems”. In: *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. 2017, pp. 1–7. DOI: 10.1109/UIC-ATC.2017.8397554.

[25] Gema Bello-Orgaz, Jason J. Jung, and David Camacho. “Social big data: Recent achievements and new challenges”. In: *Information Fusion* 28 (2016), pp. 45–59. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2015.08.005. URL: https://www.sciencedirect.com/science/article/pii/S1566253515000780.

[26] *scikit-learn: machine learning in Python — scikit-learn 0.24.2 documentation*. [Online; accessed 23. Sep. 2021]. Sept. 2021. URL: https://scikit-learn.org/stable.

[27] Muhammad Nur Fikri Hishamuddin et al. “Improving Classification Accuracy of Scikit-learn Classifiers with Discrete Fuzzy Interval Values”. In: *2020 International Conference on Computational Intelligence (ICCI)*. 2020, pp. 163–166. DOI: 10.1109/ICCI51257.2020.9247696.

[28] Rashmi Agrawal. “K-nearest neighbor for uncertain data”. In: *International Journal of Computer Applications* 105.11 (2014).

[29] Okfalisa et al. “Comparative analysis of k-nearest neighbor and modified k-nearest neighbor algorithm for data classification”. In: *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*. 2017, pp. 294–298. DOI: 10.1109/ICITISEE.2017.8285514.