



**HAL**  
open science

## Towards Predicting System Disruption in Industry 4.0: Machine Learning-Based Approach

Bouziane Brik, Belgacem Bettayeb, M'hammed Sahnoun, Fabrice Duval

► **To cite this version:**

Bouziane Brik, Belgacem Bettayeb, M'hammed Sahnoun, Fabrice Duval. Towards Predicting System Disruption in Industry 4.0: Machine Learning-Based Approach. The 10th International Conference on Ambient Systems, Networks and Technologies (ANT 2019) / The 2nd International Conference on Emerging Data and Industry 4.0 (EDI40 2019) / Affiliated Workshops, Apr 2019, Leuven, Belgium. pp.667-674, 10.1016/j.procs.2019.04.089 . hal-04082223

**HAL Id: hal-04082223**

**<https://hal.science/hal-04082223v1>**

Submitted on 26 Apr 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

The 2nd International Conference on Emerging Data and Industry 4.0 (EDI40)  
April 29 - May 2, 2019, Leuven, Belgium

## Towards Predicting System Disruption in Industry 4.0: Machine Learning-Based Approach

Bouziane Brik<sup>a,\*</sup>, Belgacem Bettayeb<sup>b</sup>, M'hammed Sahnoun<sup>a</sup>, Fabrice Duval<sup>a</sup>

<sup>a</sup>LINEACT (CESI), ROUEN, FRANCE.

<sup>b</sup>LINEACT (CESI), LILLE, FRANCE.

---

### Abstract

Industry 4.0 is the most recent industrial revolution that aims to improve not only the productivity in the 21<sup>st</sup> century, but also the flexibility, adaptability, and resilience of the industrial systems. It enables the collection of real-time data from industrial systems, Thanks to the development of Internet of Things (IoT) technology. Hence, analyzing online collected data enables to deal with several industrial issues in real-time such as machines' break- or slow-downs, quality crisis, flows disruptions, etc. In traditional industrial systems, previous works focused on both scheduling and rescheduling schemes in order to improve the system performance. However, few works dealt with system disruption monitoring due to the lack of real-time data about the system running. Furthermore, the remote and constant monitoring amenities were not established yet, properly. In this paper, we propose a system disruption monitoring tool in Industry 4.0 system. Our tool focuses on system disruption related to resources localization, or when a resource is in an unexpected location. Thus, a machine learning algorithm is used to generate a prediction model of resources localization by considering the real tasks scheduling in terms of resources localization. Therefore, as real resources localization can be collected from the industrial system through the IoT network, our tool enables to detect system disruption, risk, in real-time when comparing predicted localization to the real one. Moreover, our tool is executed in a Fog computing architecture which is emerging as an extension of cloud computing to provide local processing support with good latency. The experimental results show the efficiency of our tool in terms of prediction accuracy and time complexity when compared to other machine learning algorithms, in addition to its ability to control and detect system disruption in real-time.

© 2018 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

Peer-review under responsibility of the Conference Program Chairs.

**Keywords:** Industry 4.0, IoT, Fog computing, system disruption prediction, resources localization, machine learning.

---

\* Corresponding author. Tel.: +33751520331.

E-mail address: [bbrik@cesi.fr](mailto:bbrik@cesi.fr)

## 1. Introduction

DEPLOYING digital technologies in manufacturing is originating the fourth industrial revolution, commonly called Industry 4.0 [1][2]. Industry 4.0 is based on the extensive deployment of Cyber Physical System (CPS) and aims to significantly enhance the productivity of manufacturing technologies of the 21<sup>st</sup> century, through the collection and analysis of real-time data. This transition is mainly motivated by the emergence of Internet of Things (IoT) technology in which CPS components, machines and objects, are made “smart” by connecting them to the Internet through ubiquitous sensors [3]. Hence, IoT has been gaining attraction in many industrial areas, such as manufacturing, logistics, retailing, and pharmaceuticals [3]. We note that the term of Industry 4.0 is used in Germany, while similar terms are also used such as Industrial IoT, Smart manufacturing, Made in China 2025, etc.

The main idea of Industry 4.0 is to connect the CPS, which interlinks the physical and digital worlds, components between them through an infrastructure of IoTs and Services (IoT&S) [1]. Thus, collecting analyzing data in real-time about CPS components enables to deal with several industrial systems issues. For instance, analyzing collected real-time data, enables to African gold mine to identify a major issue related to the control of one of their process steps (oxygen level during leaching). Fixing this issue has allowed increasing the yield of the process by 3.7%, which corresponds to \$20 million saved annually [4].

In fact, previous works focus on providing either robust scheduling schemes that may reduce the impacts of small system disruptions and hence avoid rescheduling operations [5], or efficient rescheduling schemes that react to unexpected events such as machine breakdowns, and revise schedules in cost-effective way [6]. However, with the emergence of Industry 4.0, system disruption monitoring is becoming a crucial research topic to improve system performance. Thus, real-time detection of situations allows a timely reaction and helps to reduce, prevent, or avoid their impacts. Therefore, real-time disruption monitoring makes it possible to improve not only the system performance but also its flexibility and adaptability, once combined with an optimised and Just In Time (JIT) decision-making tool.

Besides, Fog computing is an emerging extension of the cloud computing which enables to process data close to its source and will be responsible for decentralized decision-making, interoperability, data analytics, information security, system latency, etc. [7][8][9]. Hence, in Industry 4.0 context, deploying the Fog computing as middleware layer between Industry 4.0 and Cloud environments, can be very useful to improve the production system’s ability to prevent and to react towards disruptions.

In this paper, we deal with system disruption monitoring in Industry 4.0 based on Fog computing architecture. As there are many types, we focus on system disruption related to resources localization in the industrial system. In other words, the situation when a resource is in an unexpected location compared to the scheduling process. To do so, we generate a prediction model of resources localization that considers the real tasks scheduling in terms of resources localization at any time and during the whole scheduling process. Therefore, we will be able to predict the risk of system disruption, in real-time by comparing updated actual localization information of resources (human, tools, etc.) to predicted one.

The remainder of this paper is organized as follows. In section 2, we provide a short literature review about system disruption monitoring in industrial context as well as on Fog computing applied to Industry 4.0. In section 3, we describe our system disruption tool based on machine learning. We discuss the obtained experimental results in section 4. Section 5 concludes the paper.

## 2. Related work

In this section, as we propose a system disruption tool in Industry 4.0 and based on Fog computing architecture. We divide the related works into two main classes: works providing Fog computing deployment for Industry 4.0 and manufacturing domains, and works dealing with system disruption in industrial context.

- Fog computing for Industry 4.0

Recently, several works have been proposed to deploy the Fog computing as middleware support between Industry 4.0 and Cloud environments, in order to perform specific tasks in a timely way. In [7], the authors discussed how Fog could be exploited as a local computing support in Industry 4.0 to deal with delay, security, and indus-

trial big data mining in real time and in different industrial scenarios. They review the main challenges that the Fog can leverage along with several scenarios from industrial use-cases including mining, transportation, waste management Industry, food Industry, agriculture, etc. The deployment of a Fog computing system in a logistic center was investigated in [8], in order to provide an intelligent computing system that includes cloud center, gateways, Fog devices, edge devices, and sensing devices. The authors studied the cost-efficient deployment of Fog computing and formulated an integer programming model before solving it using genetic algorithm. In [9], the authors proposed to place a broker node at the Fog layer which is in charge of predicting future data measurements such as IoT devices energy consumption. They then studied the role of Fog computing in improving time and computational complexities.

In fact, the deployment of Fog computing in the Industry 4.0 context is still only in its early stages. The above works focus on how Fog computing can be deployed to deal with Industry 4.0 challenges. However, to the best of our knowledge, no work proposed to exploit Fog computing in order predict, remotely and in real time, system disruption in Industry 4.0.

- System disruption in industrial systems

In [10], the authors proposed a layered disruption framework to study and understand disruption consequences in an entire industrial system by considering some types of disruptions. Several industrial cases are then analyzed, via the proposed framework, using both quantitative and qualitative approaches. A proactive disruption management system was introduced in [11]. The proposed system aims at providing a robust initial scheduling which can absorb the impacts of small disruptions and hence avoid rescheduling operations. Thus, the initial scheduling considered several disruption patterns, e.g. machine breakdowns. Similarly, several proactive disruption schemes have been proposed by providing initial robust scheduling, based on predictive patterns, probabilistic expressions, Markov decision process, etc. [12][13][14].

Though the above works deal with system disruption in manufacturing domains. However, they do not propose to monitor system disruption in real-time. They only consider system disruption patterns when defining the initial scheduling and hence avoid the costly rescheduling operations. In addition, most of them focus on machine breakdowns as disruption pattern and no work consider system disruption related to unexpected resources localization.

### 3. System disruption prediction tool

In this section, we present our proposed tool for system disruption prediction that we executed in the Fog-computing environment. Our tool is based on prediction model that we generated based on real tasks scheduling in terms of resources localization at any time and during the whole scheduling process. Assuming that we periodically receive real localization of resources from manufacturing execution system. Hence, system disruption risk may be assessed when comparing between real localization and those predicted by our model based on the real tasks scheduling. We note that in our system, a resource refers to any entity that is required to perform tasks during the execution process such as workers, machines, robots, etc. Before we proceed, we first give an of our Fog-based architecture that we considered in our scheme.

#### 3.1. Fog-based architecture for Industry 4.0

To improve both system reliability and latency, we propose a Fog-enabled Industry 4.0 architecture that comprises three main levels (cf. Fig. 1):

- Industry 4.0: it focuses on the manufacturing system which may have different types of "things" including sensors, actuators, machines, devices, workers, etc. These things are autonomous, intelligent, and are connected with each other to perform tasks. Moreover, sensed data by these things are sent either to the Fog computing to perform real-time local data processing and tasks management, or to the Internet Cloud to make more extensive analysis of the industrial big data.
- Fog computing: it acts as middleware support that performs real-time local tasks such as the delivery of fast response and computation offloading. In our case, the Fog is responsible for risk detection in the Industry 4.0

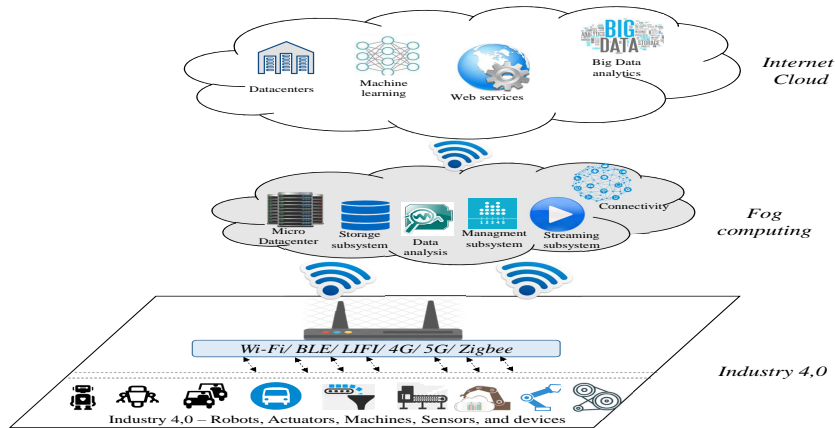


Fig. 1: Our Fog-Enabled Architecture for Industry 4.0.

system based on predicted resources localization. Hence, it will help the system supervisors to take the more adequate decisions such as rescheduling tasks, replacing the failed resource, starting maintenance operations, etc.

- Internet Cloud: the Internet Cloud is usually used to perform complex tasks which require a huge storage and computing capacities. In our architecture, the Internet Cloud environment is responsible for our prediction model creation which will be then exploited by the Fog computing to predict the resources localization in real-time.

### 3.2. Modeling the system disruption prediction

We consider a long-term dataset that includes both localization and activity information of resources over the time. The dataset contains  $n$  representative samples (rows) and  $K$  features (columns), and is divided into two subsets. The training (or learning) subset to train and generate the prediction model and testing subset to evaluate the localization prediction.

To predict resources localization in industrial systems, we are mainly based on supervised machine learning algorithms which aims at making the most accurate predictions possible. Supervised algorithms consider both input and output (desired) data to learn a mapping function,  $F(x)$ , from a  $k$ -dimensional input vector,  $x = (x^{(1)}, \dots, x^{(k)})^T \in \mathbb{R}^k$ , to an output  $y$  (or  $y = F(x)$ ). Thus, we distinguish two classes of supervised learning: regression and classification for continuous-response values and categorical response values, respectively. Our  $y_i$  is categorical in nature, as we search to predict, over time, the room  $ID$  in which each resource is located. Therefore, our problem in a classification problem.

We used Random Forest algorithm to create a classification model of resources localization (rooms) that considered resources localization during the whole execution process. This can be very useful to predict the system disruption based on both timely real localization of resources and what was experienced in the past. The random Forest algorithm is one of the most used algorithms, because of its simplicity in addition to the fact that it can be used for both regression and classification predictions.

In fact, the random Forest algorithm builds a set of decision trees randomly selected from training data. It then merges the vote from different decision trees to get the final class of the test data.

### 3.3. Data collection

Due to the lack of real datasets from industrial systems, especially those of localization information, we are based on long-term localization data of 5 people wearing four tags (sensors) and performing different activities at home [15]. These data were collected each  $10^{-1}$  second during one day. By considering workers as resources, this dataset is

suitable data for our problem because it was collected in order to detect emergency situation based on occupants' behavior (mobility), while we aim to detect system disruption in an industrial environment, and also based on resources localization (workers mobility).

The dataset comprises five features as input data including people ID, Tag ID, timestamp, date format (*dd.MM.yyyy, HH : mm : ss : SSS*), and people activity (walking, falling, sitting, etc), and the *x*, *y*, and *z* coordinates of the tag (people) which we translate to room *ID* as output data. The size of input space is [*n*; *k*]. The size of output space is [*n*; 1] which corresponds to the room *ID* value associated to the five input features.

### 3.4. Data preparation process

To achieve good results, the format of data has to be in proper manner in terms of format, scale, type, deleting null values, etc., before feeding it to the machine-learning algorithm in order to generate a classifier.

- Pre-processing. We are based on Min-Max Normalization method to put the input feature on the same scale and range. This method transforms each feature value to a new interval between 0 and 1:

$$v' = \frac{v - \min_f}{\max_f - \min_f} \quad (1)$$

Where *v* is the current value of feature *f*,  $\max_f$  and  $\min_f$  are, respectively, the maximum and minimum values of feature *f*.

- Data splitting. The data is splitted into two subsets: training subset to produce the classifier model and test subset (evaluation) to evaluate the classification accuracy of the trained model. Usually, the splitting is done with a ratio of 70 to 80 % for training and 20 to 30 % for evaluation.

### 3.5. Machine learning algorithm

As we mentioned before, we used Random Forest algorithm to create the classification model of workers' localizations. Then, we evaluate the accuracy of our model using several metrics (see subsection 4-1).

## 4. Performance evaluation

In this section, we present the experiments study that we performed to evaluate our both classification model and Fog computing-based architecture.

### 4.1. Experiments setup of classification model

To implement our Random Forest algorithm, we used scikit-learn library which provides simple and efficient tools for data mining and analysis on Python. The main parameters of our implementation settings are shown in TABLE 1. Thus, we are based on the following metrics to evaluate classification models:

- Modeling and predicting time: reflecting the time complexity of the machine learning algorithm to create the classification model and to make new predictions by using training and test sets, respectively.
- Classification Accuracy (C.A): it is measured as the number of correct predictions from all predictions made.
- Mean Squared Error (*MSE*): it represents the sample standard deviation of the differences between predicted and real (observed) values. For each test sample, we predict the localization value (Room ID), and evaluate the performance against the real localization value in terms of the *MSE* as follows,  $MSE = \frac{\sum_{i=1}^m (y_i - \hat{y}_i)^2}{m}$ , where *m* is the length of the test set,  $\hat{y}_i$  indicates the predicted localization value, and *y<sub>i</sub>* is the real localization value of the test data sample *i*.

Table 1: Implementation parameters.

Parameters	Values
<b>Dataset and Machine Learning</b>	
Number of occupants	5 occupants
Number of samples	164860 samples
Number of input variables	5 variables
Number of output variables	1 variable
Percentage of training set	75% of the dataset
Percentage of test set	25% of the dataset
Machine learning algorithm	Random Forest Algorithm

Table 2: Performance comparison between machine learning algorithms.

	Logistic Regression	Naïve Bayes	Random Forest	Decision Tree
<b>Modelling Time</b>	0.43 s	0.09 s	1.02 s	0.19 s
<b>Predicting Time</b>	0.01 s	0.03 s	0.05 s	0.01 s
<b>Classification Accuracy</b>	0.4266	0.4	0.97	0.901
<b>Mean Squared Error (MSE)</b>	1.4885	1.4641	0.000001	0.00002
<b>Average Ground Truth</b>	1.22	1.21	0.001	0.00447

- Average Ground Truth (AGT) : explains the average output variable value (localization) away from the ground truth value when making predictions on the test set. The AGT is deduced from MSE as follows:  $AGT = \sqrt{MSE}$ .

In addition, we compared our random forest algorithm with three other classification learning algorithms: (i) logistic Regression Classifier (LRC) [16], Gaussian Naïve Bayes Classifier (NBC) [17], and Decision Tree Classifier (DTC) [18].

#### 4.2. Experiments setup of Fog computing-Industry 4.0 architecture

We simulated our Fog computing environment for system disruption application, using iFogSim simulator [19]. We compared our Fog computing environment to Cloud-only scheme (i.e., when system disruption management is performed in Cloud environment). Thus, we evaluated efficiencies of both schemes in terms of latency, network use, and energy consumption.

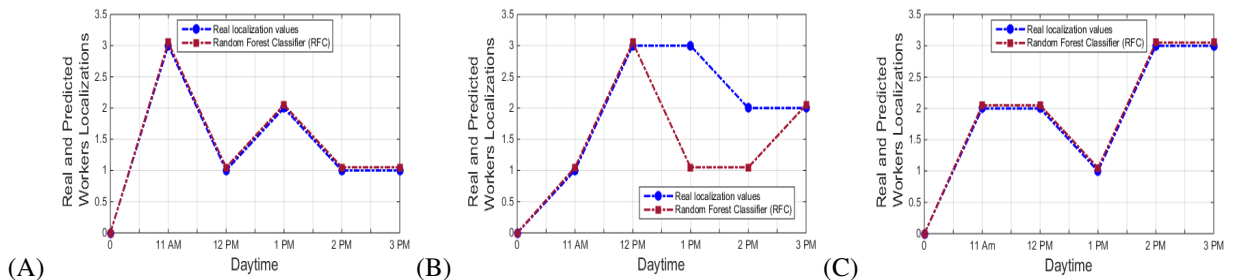


Fig. 2: Comparison between real and predicted localization information (A) Worker 1. (B) Worker 2. (C) Worker 3.



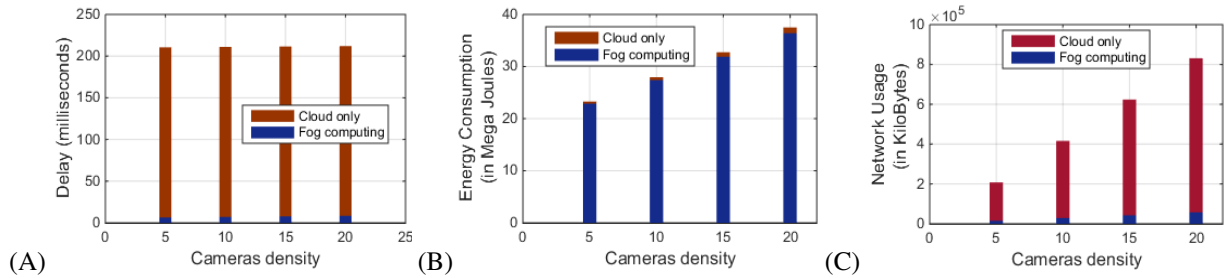


Fig. 3: Comparison between execution in Cloud computing and Fog computing: (A) Average Latency. (B) Energy consumption of deployed cameras. (C) Network overhead.

#### 4.3. Performance evaluation of classification model

Table 2 illustrates performance comparison between our machine learning algorithm and the three other ones. We observe that the random forest algorithm has a modeling time higher, 1.02, than the other algorithms. Nevertheless, it clearly reduces the prediction time compared to its modeling time and to the other algorithms. In fact, the modeling time is not that important as it is executed only once to generate the prediction model. Thus, the random forest algorithm significantly improves prediction accuracy by maximizing the classification accuracy (CA), and minimizing both *MSE* and *AGT*. Even the modeling time of the other algorithms is better than that of our random forest algorithm. However, we chose the random forest algorithm because, on one hand, it generates an acceptable prediction time complexity and, on the other hand, it clearly improves the classification accuracy, which makes it more suitable to predict resources localization in real time.

Fig. 2 compares between real localization and those predicted by our classifier of three workers and at five time points. As mentioned before, we translate the  $x$  and  $y$  coordinates of the workers into room *ID* where we define three rooms (1, 2, and 3). We also note that, in Industry 4.0, real localization can be collected from the industrial system through IoT-based network. In our case, we use the real localization from the dataset. As we observe, predicted localization are closer to the real one for both worker 1 and 3 (cf Fig. 2-A and C). However, for worker 2 and at both *1PM* and *2PM*, our classifier predicts that worker 2 must be in room 1 to perform his tasks while the IoT-based network detects that he is in room 3 (real localization). This represents a disruption situation of the industrial system and system tasks must be rescheduled in order to consider worker-2's tasks. We note that tasks rescheduling operation is beyond the scope of this paper.

#### 4.4. Performance evaluation of Fog computing-Industry 4.0 architecture

Fig. 3 compares between our Fog computing-based scheme and Cloud only scheme when varying deployed cameras in the studied factory.

Fig. 3-(A) depicts the generated average latency. We remark that both schemes generate a stable average latency as we increase the cameras density. Thus, we also observe that our Fog computing-based scheme succeeds in reducing the average latency when compared to the Cloud-only scheme, as it places the processing modules close to the network edge. Hence, the average latency does not depend on the network density but on the processing modules placement.

Fig. 3-(B) and (C) show energy consumption of cameras and network use, respectively, when also varying the cameras density. Our scheme minimizes the network use compared to the Cloud-only scheme, due to the application deployment that decreases the volume of data sent to centralized Cloud data center. We also see that both energy consumption and network usage increase as we increased the cameras density. In fact, cameras perform motion detection, which causes a large amount of consumed power and transferred data, especially when increasing the number of deployed cameras. In addition, deployed cameras almost consume the same energy in both schemes. These results can be justified by the fact that consumed energy does not depend on the application placement (Fog or Cloud) but on the number of used cameras.

Therefore, based on these results, it is obvious that random forest algorithm achieves a better and stable performance compared to the other algorithms. Thus, it improves significantly the prediction time complexity as well as



the model prediction accuracy. Moreover, our classifier model is then exploited, in a Fog-computing environment, to detect industrial system disruption in real time, and hence to give system supervisors feedbacks about the whole system running.

## 5. Conclusion

In this paper, we proposed a new machine learning-based system disruption detection scheme of Industry 4.0. Our scheme is performed in a Fog computing-Industry 4.0 environment and considers real tasks scheduling to learn prediction model of resources localization using random forest algorithm. The learned model is then exploited to detect system disruption risk in real time, when comparing real resources localization with predicted ones. Hence, our scheme may give system supervisors feedbacks in real time and helps them to make the adequate decision. Our prediction model is compared to other models created by other machine learning algorithms. Experimental results showed the efficiency of our algorithm to improve the prediction time complexity as well as the model prediction accuracy.

As a future work, we are working to collect real dataset from an industrial system for more performance evaluation of our prediction model. In addition, we are also working to propose an efficient tasks rescheduling algorithm in Industry 4.0 to deal with manufacturing systems disruptions in real time.

## Acknowledgment

Acknowledgement is made to the Normandy region and the European Union for supporting this research through the regional research RIN program by funding the project entitled LOCADYN within which this research is being performed.

## References

- [1] P. Lade, R. Ghosh, S. Srinivasan, Manufacturing analytics and industrial internet of things, *IEEE Intelligent Systems* 32 (3) (2017) 74–79.
- [2] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, B. Yin, Smart factory of industry 4.0: Key technologies, application case, and challenges, *IEEE Access* 6 (2018) 6505–6519.
- [3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: A survey on enabling technologies, protocols, and applications, *IEEE Communications Surveys Tutorials* 17 (4) (2015) 2347–2376.
- [4] What is industry 4.0, <https://www.forbes.com/sites/bernardmarr/2018/09/02/what-is-industry-4-0-heres-a-super-easy-explanation/> 14ca62909788, accessed: 2018-11-07.
- [5] J. Zhang, J. Yang, Y. Zhou, Robust scheduling for multi-objective flexible job-shop problems with flexible workdays, *Engineering Optimization* 48 (11) (2016) 1973–1989.
- [6] G. E. Vieira, J. W. Herrmann, E. Lin, Rescheduling manufacturing systems: A framework of strategies, policies, and methods, *Journal of Scheduling* 6 (1) (2003) 39–62.
- [7] M. Aazam, S. Zeadally, K. A. Harras, Deploying fog computing in industrial internet of things and industry 4.0, *IEEE Transactions on Industrial Informatics* 14 (10) (2018) 4674–4682.
- [8] C. Lin, J. Yang, Cost-efficient deployment of fog computing systems at logistics centers in industry 4.0, *IEEE Transactions on Industrial Informatics* 14 (10) (2018) 4603–4611.
- [9] G. Peralta, M. Iglesias-Urkia, M. Barcelo, R. Gomez, A. Moran, J. Bilbao, Fog computing based efficient iot scheme for the industry 4.0, in: 2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM), 2017, pp. 1–6.
- [10] K. Kilkki, M. Mäntylä, K. Karhu, H. Hämmäinen, H. Ailisto, A disruption framework, *Technological Forecasting and Social Change* 129 (2018) 275 – 284.
- [11] S. Sejdovic, N. Kleiner, Proactive and dynamic event-driven disruption management in the manufacturing domain, in: 2016 IEEE 14th International Conference on Industrial Informatics (INDIN), 2016, pp. 1320–1325.
- [12] Z. Feldman, F. Fournier, R. Franklin, A. Metzger, Proactive event processing in action: A case study on the proactive management of transport processes (industry article), in: Proceedings of the 7th ACM International Conference on Distributed Event-based Systems, DEBS '13, ACM, New York, NY, USA, 2013, pp. 97–106.
- [13] A. Metzger, R. Franklin, Y. Engel, Predictive monitoring of heterogeneous service-oriented business networks: The transport and logistics case, in: 2012 Annual SRII Global Conference, 2012, pp. 313–322.
- [14] Y. Engel, O. Etzion, Towards proactive event-driven computing, in: Proceedings of the 5th ACM International Conference on Distributed Event-based System, DEBS '11, ACM, New York, NY, USA, 2011, pp. 125–136.

- [15] B. Kaluža, V. Mirchevska, E. Dovgan, M. Luštrek, M. Gams, An agent-based approach to care in independent living, in: *Ambient Intelligence*, Springer Berlin Heidelberg, 2010, pp. 177–186.
- [16] J. R. Brzezinski, G. J. Knafl, Logistic regression modeling for context-based classification, in: *Proceedings. Tenth International Workshop on Database and Expert Systems Applications. DEXA 99*, 1999, pp. 755–759.
- [17] M. Martinez-Arroyo, L. E. Sucar, Learning an optimal naive bayes classifier, in: *18th International Conference on Pattern Recognition (ICPR'06)*, Vol. 4, 2006, pp. 958–958.
- [18] S. R. Safavian, D. Landgrebe, A survey of decision tree classifier methodology, *IEEE Transactions on Systems, Man, and Cybernetics* 21 (3) (1991) 660–674. [doi:10.1109/21.97458](https://doi.org/10.1109/21.97458).
- [19] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, R. Buyya, ifogsim: A toolkit for modeling and simulation of resource management techniques in internet of things, edge and fog computing environments, *CoRR* abs/1606.02007.