An architecture for indoor location-aided services based on collaborative industrial robotic platforms

Octavian Arsene, Corina Postelnicu, Wenbo Wang, Elena Simona Lohan, *Senior Member, IEEE,* and Dumitru Iulian Nastac, *Member, IEEE*

***Abstract*—An essential component in the intelligent wireless processing for the future industrial halls will be the data labelling with location information. The location information will facilitate not only the remote control and autonomy of the industrial robots and sensors, but it will also enable predictive control and maintenance, increased productivity, and increased workers' safety. The data labelling is typically a tedious and costly process when done manually or semi-automatically, and the fully automated data labelling has still to overcome several challenges that we describe in this paper. We propose a collaborative robotic architecture equipped with simultaneous localization and mapping as well as machine-learning-based algorithms. A scenario in an industrial setting is presented, in which data acquisition by robots, with various capabilities, can be used to enable location-based services for increased workers' safety and to offer timely tracking of mobile assets for an increased productivity. The robotic platform acquires data during the periods when the robots are not allocated to their main tasks. Besides, we demonstrate that the above mentioned robotic platform could benefit from machine learning, for example, the accurate estimation of positions and good adaption in different type of collected data sets.**

***Index Terms*—automated data acquisition, location-labelled data for edge processing, robotic platforms, machine learning**

I. Introduction

THE technical possibility to integrate social, mobile, and machine networks brings to reality a world where humans and objects have awareness of and can utilize and influence each others' locations, capabilities, and stories. We can now engineer the pursuit of dynamically changing goals, having in mind dynamically changing models of the world - provided that all- and only-the-relevant actors, data producers and consumers, are discovered and incorporated into the underlying formal understanding of context. The construction of maps of meaning in real time is very much needed. In emergency situations, automatic collection and interpretation of data from selected discovered fixed and mobile sensors can assist the calculation of escape routes taking into consideration sudden obstacles, e.g., in order to get quick access to injured persons. With the rapid developments of automated robots, the factories of the future will have an increased number of uncontrolled factors, such as possible abrupt changes in the velocity, acceleration, or rotation of the robots, extreme temperatures, uncontrolled waste disposal, unsafe gaseous leakages, etc. These unpredictable changes in the environment, even if rare, can have a tremendous negative impact on the production quality and the workers safety. Dynamic understanding of the environment in an industrial setting would support interventions in difficult or hard-to-reach places.

O. Arsene is with University Politehnica of Bucharest, Faculty of Electronics, Telecommunications and Information Technology, Romania.

C. Postelnicu is with IT Center for Science and Technology, Romania, an R&D company with high expertise on service and social robots .

E.S. Lohan and W. Wang are with Tampere University, Finland.

D.I. Nastac is with University Politehnica of Bucharest, Faculty of Electronics, Telecommunications and Information Technology, 1-3 Iuliu Maniu Blvd., 061071 Bucharest,Romania, e-mail: nastac@ieee.org.

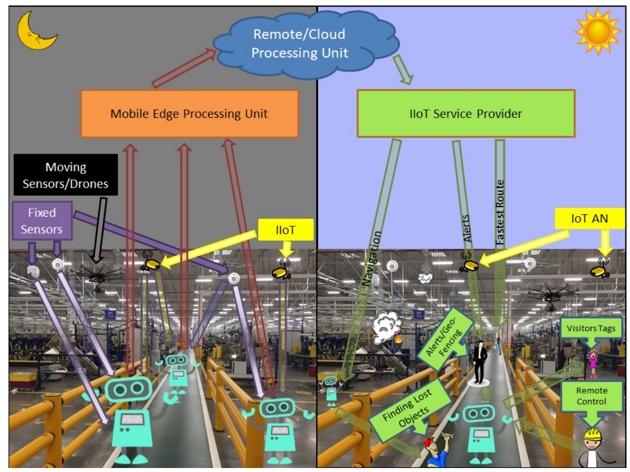

Fig. 1. Our vision of robot-based data acquisition and services in an industrial scenario.

The future will bring to indoor environments a plethora of collaborating robots and drones, such as the service robot Tiago by Pal Robotics [1], or the low cost TurtleBot2 [2]. These robots will go beyond the offering of guidance, notifications, and advertisements, becoming engaged in activities now carried out by humans on a daily basis, and acting if needed upon their environment. An illustration of such challenging scenario is given in Fig. 1. The factories of the future will see various industrial tasks completed by flying, terrestrial, or under-water robots co-existing with human workers. For example, drones will be accessing tall towers or furnaces for fast fault analysis and remote maintenance, or will build sound, gas, or light maps to optimize industrial functionalities; ground robots will be tracking various assets or will monitor possible ground leakages; the use of under-water robots in industries such as oil or gas industry will increasingly grow.

The potential of robot platforms as 'while-idle' or 'on-demand' data producers to feed the applications built for the everything-as-a-service paradigm must be harnessed. *On top of their regular tasks*, robots could help track lost tools, issue alerts (e.g. "machine left with power on", "danger zone")

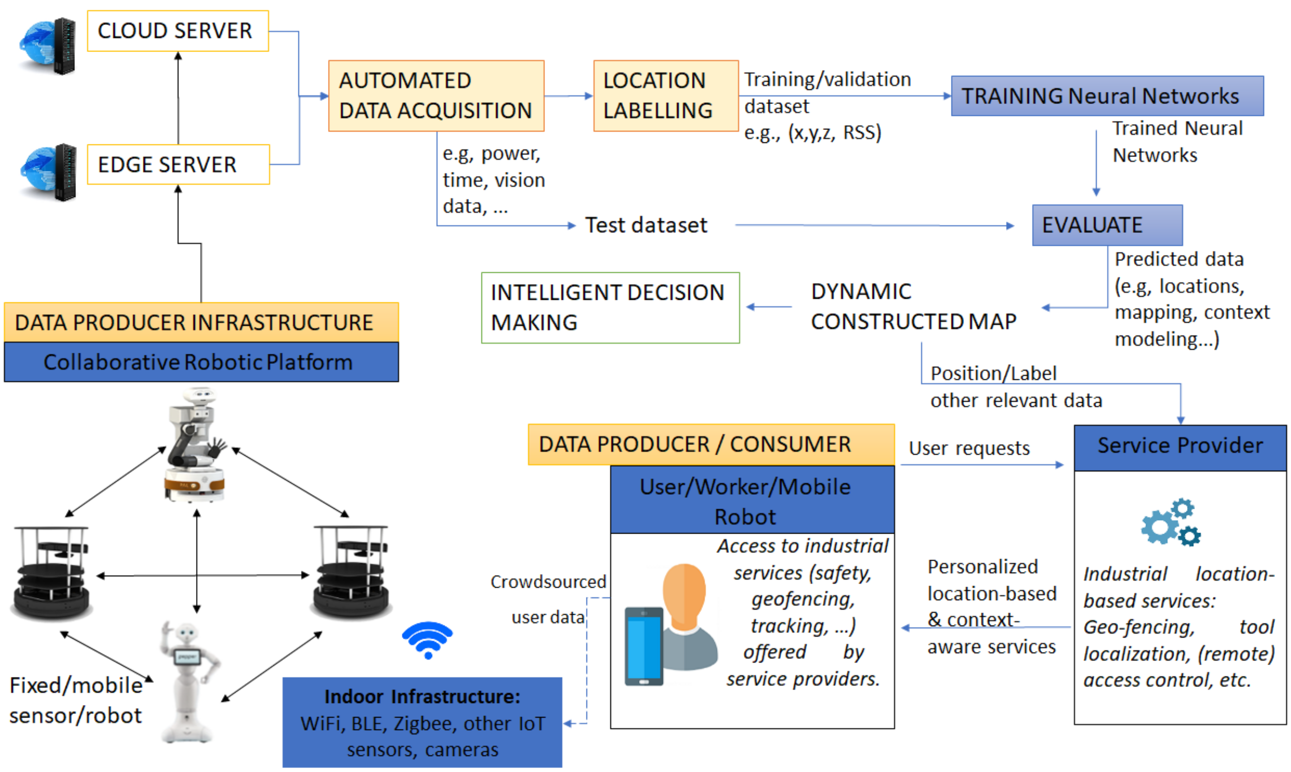

Fig. 2. Proposed architecture for future intelligent IoT factories.

and location-relevant notifications (e.g. "robot at third floor disabled/malfunctioning"), decrease the energy consumption by optimal allocation of tasks, and carry machinery and tools from one place to another. Also, industrial robots supporting the regular industrial flux of operations could assemble in an ad-hoc manner to respond to discovered exceptional situations.

In the Authors' opinion, three of the must-haves of future smart industrial Internet of Things (IIoT) systems are: i) automatic data acquisition, in order to smartly acquire various types of needed available data; ii) automatic and low-cost localization of mobile and fixed sensors; and ii) the automatic labelling of massive amount of data in a cost efficient manner. The automated data acquisition will increase the data transmission and storage efficiency by overcoming many of the data quality drawbacks inherent to manual or semi-automatic collection of data. The location information will allow an intelligent communication to/from the access nodes, such as through location-based beamforming and location-based routing solutions, and will enable a wireless predictive maintenance through Simultaneous Location and Mapping (SLAM) procedures. The automating data labelling will support the seamless understanding of context. SLAM-capable robotic platforms equipped with machine-learning based self-positioning algorithms directly address the need to recognize the dynamic changes in the environment that are relevant to the goal for which the industrial map is built, and support possibilities to act, taking into account these changes. Synchronizing the functionalities and signals of various robots and creating efficient measures to deal with rare-but-critical anomalies would require a *fast processing of a large amount of data close to/at the edge of the operation site*.

This paper presents an architecture based on data mining for knowledge discovery in future edge-based industrial IoT and it show-case results based on a collaborative robotic platform in which data acquisition by several robots can be used for indoor localization in large and complex environments, i.e. an indoor industrial uncontrolled environment.

II. Proposed Architecture

We propose a location-based architecture with collaborative robots and mobile edge computing to better support the industrial operations in the factories of the future, as shown in Fig. 2. Our location engine relies on Machine Learning (ML) from data collected from several robots in a collaborative manner. The ML-based processing is done on an edge server for reduced complexity and computation offloading. Our location engine would enable an automatic labelling of data coming from multiple robots as well as an accurate environment mapping. Such location and mapping information would be then used to optimize the communication and synchronization between robots, using for example location-based robot control and map-based task adaptations, e.g., a robot completing its tasks fast can undertake other tasks in a queue at a certain location, or a faulty robot can be shut down fast by its neighbouring robots, which will then split between them, in a collaborative manner, the un-addressed tasks. A Mobile Edge Computing (MEC) center will aggregate and process the data acquired automatically by the robots. Through its close (physical and virtual) location to the robots, MEC will enable low and ultra-low latencies in the inter-robot communications

	HIGH-LEVEL TARGETS	CURRENT APPROACHES	PROPOSED VISION	CHALLENGES
Data acquisition	• Maximization of data quality • Agility (robustness to changes, e.g. new data types or sources)	• Dynamic path planning algorithms • Ad-hoc decentralized networks	• End-to-end automatization of the data acquisition pipeline via cooperative coordination of multi-agent systems	• Multi-agent synchronization
	• Minimization of data acquisition time • Low-latency data routing and processing • Optimal tradeoff: data size vs data acquisition effort • Capitalization of heterogeneity of data producers	• Edge-cloud architectures; distributed topology control • Compressed sensing	• Activity-level-based task sharing between robots (e.g., more intensive mapping done during low-intensity activities of the robot) • Multi-step regression mechanisms for data prediction, with small-sized patterns repeated at regular intervals for verification and updating • Location-based network topology optimization • Location-based labelling to aid the synchronization processes from heterogeneous access nodes	• Designing dynamic ad-hoc networks with moving access nodes • Application-dependent optimality metrics • Synchronization mechanisms in time sensitive networks
Data labelling	• Scalability • Higher and higher levels of generalization on unknown data	• Manual labelling • Machine learning (traditional methods) • Small-scale modeling of large-scale effects	• End-to-end automatization of the data labelling pipeline; • Deep learning over traditional ML techniques	• Acquisition and maintenance of sufficiently large training dataset
Edge computing	• Edge computing under mobility constraints • Agility	• Efficient scheduling to optimize edge server capacity	• Labelled or location-aware data reference data collected from low-cost sensors (e.g., received signal strength-based labeling) to facilitate the off-loading process to the edge	• Integration of new data producer types (e.g. sensors) • Accuracy of statistical channel models for statistical-based estimators
Context knowledge extraction	• Scalability • Robustness to changes in environment • Ability to quickly cover wide areas	• Dynamic radio maping of environment (SLAM, crowdsourcing)	• Collaborative fast robots/drones	• Joint estimation of robot rotation and robot and access nodes positions in highly dynamic environments

Fig. 3. Challenges, solutions, and future vision in the automated data acquisition.

and in the industrial services supported at the uncontrolled industrial environment.

The ML algorithms are essential to deal with massive amount of data and they are used in our architecture both for the location labeling part and for the data prediction part, such as context modeling and prediction of future trajectories of mobile nodes in the network. The heterogenous sensors and robots, relying on multiple IoT standards, such as ZigBee, LoRa, NB-IoT, etc., are collaborating wirelessly with each other, perform their main industrial operations, and contribute to the automatic data gathering and transmission to the MEC.

III. Challenges

The above outlined vision poses several challenges at various levels. Fig. 3 summarizes the challenges and foreseen solutions related to data acquisition, pre-processing and knowledge extraction via collaborative robotic platforms.

A. *Automated data acquisition*

An abundance of *data quality problems* arises when data is manually manipulated. Inconsistent logging principles cause difficulties in comparing logs. Data may be incorrectly entered in the system, or be missing. Impreciseness, such as day-level instead of second-level of granularity in time-stamp specifications, might lead to difficulties to order events. The semantics of the terminology used might be different across logs (same term used for two different concepts), if not taken from a reference ontology or taxonomy.

An automated data acquisition mode enables a design paradigm where the robots can seek out the information they need and teach themselves about their surroundings (via e.g. SLAM techniques), as opposed to being taught a manually annotated map of their environment. This requires a lot more contextual awareness and *models capable of higher levels of generalization* on unknown data.

The appropriate environment parameters from the huge amount of data available to the sensors must be selected and preprocessed to support *robustness to changes in the environment*, such as new data sources or types. The highly dynamic nature of indoor environments (changes of equipment place, doors opening and closing) should be reflected in the created indoor radio maps. There are many robot self-localization algorithms [3], [4], ranging from dead-reckoning to having visual or infra-red landmarks or using anchor nodes based on Ultra-Wide Band technology [5]. Some visual SLAM methods can automatically calculate landmarks based on visually distinctive and easily retrievable features in the environment. 2D maps are enough to achieve self-localization in indoor environments (the fewer the features the more challenging the self-localization process) [6]. Perfecting Neural Networks (NN) models by ensuring ever growing amounts of data into the input pipeline is guaranteed to achieve high-positioning accuracy.

B. *Machine learning for knowledge extraction*

Knowledge extraction relies on a set-up stage, when a regression/classification unsupervised model is trained on an initial balanced dataset, and a deployment stage, when new data is collected and automatically labelled based on the trained model. Traditional ML techniques, such as logistic regression, were originally designed for small size input training datasets. They rely on domain-specific knowledge to predefine the features to extract, and are characterized by learning curve performance plateaus that cannot be broken by simply inputting the always available additional data of noisy and complex environments into the algorithm.

Most existing deep learning applications rely on cloud-assisted training and device-level inferences. Processing tasks can be offloaded from the cloud layer by moving part of the computation to local servers (the so-called edge layer), if the size of the intermediate data (produced by the edge nodes/servers and subsequently transferred to the cloud) is smaller than the size of the input data. Edge servers could be installed for example on the ceiling Access Nodes or even on a robot with higher computational capabilities than its neighbours. A good trade-off between processing in the edge servers versus the cloud (the access point for third party service providers) should seek to optimize the utilization of server capacity, via efficient scheduling. This should minimize network traffic, but it is difficult to achieve when the edge nodes are mobile. Location-based or labelled-based processing in the edge is the solution to address mobility constraints.

C. Cross-cutting implementation concerns

Minimization of cost via capitalization of re-usability: Existing infrastructures and open-source software should be re-utilized. Received Signal Strengths (RSS) together with source coordinates (e.g. the buildings' Access Nodes (AN), the Media Access Control MAC address) form fingerprints a map can reliably be constructed from. WiFi signals, Bluetooth Low Energy (BLE) signals, older cellular signals (i.e., before 4G, as the newer ones, such as 5G cellular systems, have dedicated Positioning Reference Signals), and to a smaller extent also RFID and Ultra Wide-Band (UWB) signals are potential contributors to extracting scene fingerprints. Adapting and modifying existing open source software packages to work together can be made difficult by the lack of documentation and maintenance.

Minimization of data acquisition time: Appropriate target points for the individual robots need to be chosen such that they reliably communicate to dynamically coordinate when exploring simultaneously different regions of their environment. This calls for dynamic path planning algorithms and well-organized communication protocols [7]. For the latter, ad-hoc networks are ideal candidates that provide a decentralized communication framework which does not rely on a fixed infrastructure.

Optimal distribution of processing tasks:

- Pre-processing computing tradeoff: Equilibrium must be found between the data acquisition effort (continuous vs. only during allotted time periods with less/idle activity) and the size of the data to be collected, stored and processed.
- Low latency: Set-up and data acquisition could take place off-line, in real-time, or by a combination of an off-line set-up stage and real-time data acquisition. To achieve low-latency it is preferable to leave only the necessary computing tasks to the service provision stage.

Optimal information exchange: Collaborative platforms are equipped with varying degrees of sensor accuracy (from e.g. low cost ultrasonic range finders up to high accuracy laser range finders [8]). The heterogeneity of robot teams must and can be capitalized via location labelling to synchronize processes from heterogeneous access nodes.

Real-time deployment : In dynamic environments (e.g., moving robots) trajectories will need to be recalculated and adapted often, putting strain on the hardware resources. A too low SLAM polling rate can lead to cumulative errors and missing loop-closures (i.e. imprecise/erroneous maps).

Security and privacy: Users should be able to choose whether they want to be data and application consumers, producers, or both. Relying on deep learning for edge computing will support the preservation of privacy.

Setup challenges: Some supervision is required during the initial deployment in a new dynamic environment, to avoid unexpected situations that might lead to crashes with obstacles.

IV. USE CASE

We propose a low-cost positioning engine relying on ML to support the labeling of the acquired data. The implementation details are presented as follows.

A. Setup

Data acquisition: *Semi-automatic data gathering* assumes that each data collection is triggered manually by the human operator in each node of a predefined path. The integrated platform consists of a custom advanced WLAN scanner, a laptop and a laser telemeter. The WLAN device relies on the popular ESP 8266-07 WLAN module which has an easily programmable interface (as an extension for the Arduino IDE), extra-low power consumption and powerful data processing capabilities.

The *fully automated* approach tested relies on the autonomous robotic platform TurtleBot2 (see Fig. 4) on a Kobuki mobile base, a WLAN scanner for RSS data acquisition and an antenna mounted at the approximate level of a mobile phone held by a human when looking at a screen application (approximately 120 cm from the ground). The TurtleBot series is based on the Robotic Operating System (ROS) deployed on an Intel NUC computer (Intel core i7, 8 GB RAM) running Ubuntu Linux. The mobile base, featuring two wheels and a caster, is configured as a differential drive base, so that the velocity of the wheels can be controlled independently. In addition to base sensors (bumpers, cliff sensors, wheel drop sensors), the TurtleBot2 is also equipped with an Orbbec Astra2 3D camera. An access node was associated a MAC address. Several MAC addresses can come from the same physical AN.

Open-source data: Three open-source WLAN Received Signal Strength (RSS) fingerprints datasets were studied for position-labelling. [3], [4] and [9].

Open-source software for position-labelling: The details of localization methods [10] vary depending on the global goal. In this setup, we seek to utilize low-cost solutions with available infrastructures. A customized ROS node and the Real-Time Appearance-Based Mapping (RTAB-Map) package for localization [11] are used to automatically associate the robot position with the captured RSS. The RTAB-Map package is configured to use the multiple sensors [11] available: the RGBD camera, the odometer and an additional RPLidar [12].

Fig. 4. TurtleBot with Lidar, RGBD camera, odometer and WLAN scanner.

B. Methods: Data acquisition and knowledge discovery

Data preprocessing: Preprocessing of the collected data (i.e. the RSS and associated 3D Cartesian environment coordinates) involves a scaling phase (where features are brought onto the same scale by removing the mean and scaling to unit variance), followed by Principal Component Analysis (PCA) to keep only the features with a specific cumulative explained variance and therefore to avoid data over-fitting.

PCA was employed to select 77 features out of of a total of 448, which explained approximately 50% of the total variance in the data.

ML for positioning : We compared deterministic algorithms [3] [4], e.g. k-Nearest Neighbours (kNN) with various distance metrics with non-deterministic regression and classification algorithms. The non-deterministic approaches explored included:

- Regression on one input vector of RSS values to predict 3D Cartesian coordinates, performed on three distinct regression models (Multi-layer Perceptron (MLP) with one hidden layer, Multi-output Regressor with GradientBoosting (MOR-GB) and RandomForest (MOR-RF) estimators [13], [14]).
- A hybrid solution, relying on the assumption that the user's elevation from the floor is irrelevant if the correct floor is priorly identified: floor classification (via Multi-layer Perceptron with one hidden layer), followed by 2D regression within each identified floor area (via each of the three regression models listed above). This method is more specialized, without losing efficiency or the power to generalize.

The optimal combination of hyper-parameters was found using brute-force grid search of the smallest cross-validation error (Euclidean distance metric) [15]. K-fold cross-validation was employed to assess model performance. Predictions of the trained model on a test dataset were used to evaluate the generalization error. Overfitting was prevented via PCA-assisted dimensionality reduction, early stopping of the training procedure (if increase in cross-validation error was detected), and regularization of the network.

C. Results: Automatic data acquisition and SLAM

The above setup was deployed on the third floor of building B in the Leu campus of Politehnica University of Bucharest.

Fig. 5 (top left) illustrates the power map generated from one AN in the building. The top right part of Fig. 5 compares the performance among kNN-based, regression-based, regression-classification-based location estimators, for a small-sized building with large amount of training data (namely, the open-source dataset in [3]). The best regression model was found to be a ML algorithm based on regression. The best hybrid method results (an accuracy for the number of correctly predicted floors of 99.7%) were obtained for 50 nodes in the hidden layer and weight optimization via the Adam solver over 200 iterations. Both regression and hybrid methods have a 75% chance of predicting the correct position within 2 meters and for 4 meters the confidence threshold is at 95%.

The hybrid approach does not present noticeable accuracy advantages over the simple regression method, but its modularity makes it better suited for continuous data collection by robotic platforms than its simple regression alternative. Parts of the environment can be updated separately, without having to re-train the entire model, as it is the case for the simple 3D regressor. This saves time and resources, and allows for more constant updates of the map.

The bottom right part of Fig. 5 showcases a side-by-side plot of the real plan of the corridor (two opened doors and a pillar) versus the corridor map created by TurtleBot2 after one of the mapping sessions. Details such as open doors and pillars are well reproduced by the robot. The RATB-Map package configured with all three available sensor types proved to exhibit very high precision both in mapping featureless corridors and, more importantly, in correcting cumulative errors over time. The localization error decreased from $1.5\,\mathrm{m}$ down to less than $20\,\mathrm{cm}$.

V. Conclusion

Every object or actor in a value chain can now have knowledge of the id, history, and planned future, both for itself and surrounding objects/actors. In industrial environments for example, this is now achievable at all levels of granularity of the factory, from low-level sensors to flexible manufacturing cells, via IoT tags and web-service encapsulation of product needs and equipment skills.

Seamless human-object interactions enabled by the IoT support new products and applications on top of the everything-as-a-service business paradigm. The best of such products and applications will have the ability to immediately understand

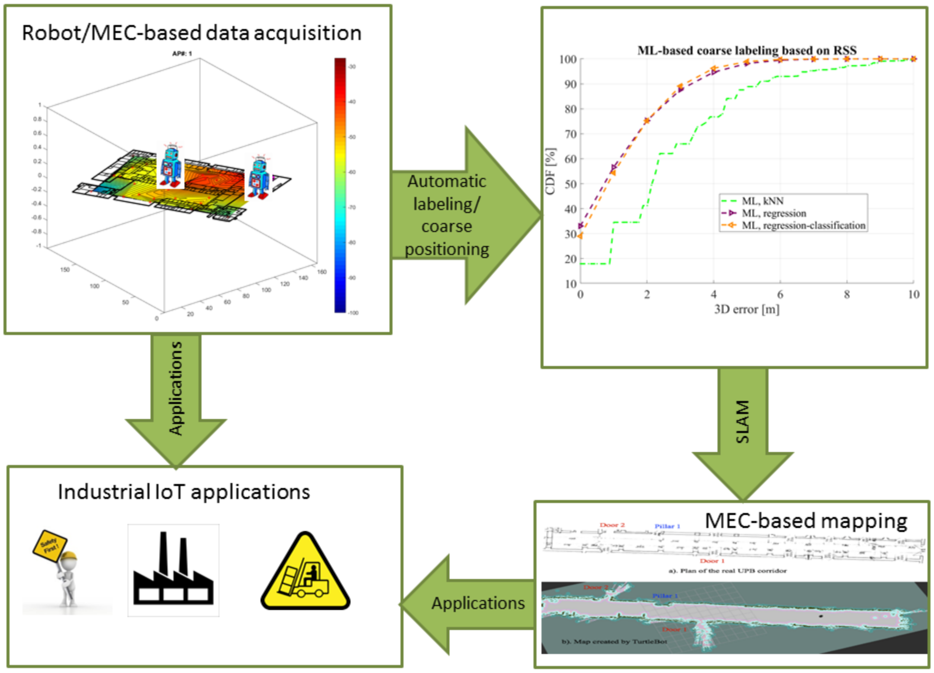

Fig. 5. The flowchart of the proposed automatic location labeling of data for intelligent edge computing in IoT.

and react to dynamic changes of their environment. To this end, the collaborative robotic platforms of the future should have capability to construct radio maps of their surroundings in real time, by selectively choosing the sensors and the data relevant for a final goal (e.g. positioning, assistance, alerting), and converting it automatically into meaningful information.

ML algorithms clearly outperform deterministic algorithms, especially where larger sets of training and test data are available. Modularity supports separate update of the different environment model parts. The larger the amount of collected data the better the environment models, allowing for significant gains in accuracy. Automating the data collection process would result in much larger amounts of available data, improving the efficiency of applications consuming this data. SLAM and machine-learning based self-positioning algorithms are good candidates to enable automated location-labelling of data for the intelligent factories of tomorrow.

Acknowledgment

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI - UEFISCDI, project number PN-III-P2-2.1-PED-2016-1779, within PNCDI III. The work has been also partially supported by Academy of Finland (grant number 313039), which is gratefully acknowledged.

References

[1] PalRobotics, "TIAGo," http://tiago.pal-robotics.com.
[2] Willow Garage, "TurtleBot 2," https://www.turtlebot.com/turtlebot2.
[3] G. M. Mendoza-Silva, P. Richter, J. Torres-Sospedra, E. S. Lohan, and J. Huerta, "Long-Term Wi-Fi fingerprinting dataset and supporting material," Nov. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1066041
[4] P. Richter, E. S. Lohan, and J. Talvitie, "WLAN (WiFi) RSS database for fingerprinting positioning," Jan. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1161525
[5] J. Li, L. Cheng, H. Wu, L. Xiong, and D. Wang, "An overview of the simultaneous localization and mapping on mobile robot," *Proceedings of International Conference on Modelling, Identification and Control*, pp. 358–364, 2012.
[6] Z. Kong and Q. Lu, "A brief review of simultaneous localization and mapping," *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, pp. 5517–5522, 2017.
[7] J. Chen and L. Li-Ren, "Path planning protocol for collaborative multi-robot systems," *Proceedings of Computational Intelligence in Robotics and Automation, 2005. CIRA 2005*, pp. 721–726, 2005, espoo, Finland.
[8] D. Fox, W. Burgardz, H. Kruppa, and S. Thrun, "Collaborative multi-robot localization," *KI '99 Proceedings of the 23rd Annual German Conference on Artificial Intelligence: Advances in Artificial Intelligence*, pp. 255–266, 1999.
[9] LOMAR, "UPB dataset," www.lomar.pub.ro/gb/rezultate/index.html.
[10] A. Khalajmehrabadi, N. Gatsis, and D. Akopian, "Modern WLAN fingerprinting indoor positioning methods and deployment challenges," *CoRR*, vol. abs/1610.05424, 2016. [Online]. Available: http://arxiv.org/abs/1610.05424
[11] M. Labbe and F. Michaud, "Online Global Loop Closure Detection for Large-Scale Multi-Session Graph-Based SLAM," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 2661–2666.
[12] SLAMTEC. Rplidar a1. [Online]. Available: http://www.slamtec.com/en/lidar/a1
[13] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.
[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[15] S. Raschka, D. Julian, and J. Hearty, *Python: Deeper Insights into Machine Learning*. Packt Publishing, 2016, ch. 6.