Article

Towards Sensor Reliability Using Internet-of-Things LiDAR Data in a Cyber-Physical System

Fernando Castaño[1,*], Alberto Villalonga[1], Rodolfo E. Haber [1], Joanna Kossakowska [2] and Stanisław Strzelczak [2]

[1] Centre for Automation and Robotics (CSIC –UPM), Spanish National Research Council, Arganda del Rey, Madrid, 28500, Spain; rodolfo.haber@car.upm-csic.es

[2] Faculty of Production Engineering, Warsaw University of Technology, Warsaw, Poland; s.strzelczak@wip.pw.edu.pl

* Correspondence: fernando.castano@car.upm-csic.es; Tel.: +34-918-70-50 (F.C.)

Abstract: Currently, the most important challenge in any assessment of state-of-the-art sensor technology and its reliability is to achieve road traffic safety targets. The research reported in this paper is focused on the design of a procedure for evaluating the reliability of Internet-of-Things (IoT) sensors and the use of a Cyber-Physical System (CPS) for the implementation of that evaluation procedure to gauge reliability. An important requirement for the generation of real critical situations under safety conditions is the capability of managing a co-simulation environment, in which both real and virtual data sensory information can be processed. An IoT case study that consists of a LiDAR-based collaborative map is then proposed, in which both real and virtual computing nodes with their corresponding sensors exchange information. Specifically, the sensor chosen for this study is a Ibeo Lux 4-layer LiDAR sensor with IoT added capabilities. Implementation is through an artificial-intelligence-based modeling library for sensor data-prediction error, at a local level, and a self-learning-based decision-making model supported on a Q-learning method, at a global level. Its aim is to determine the best model behavior and to trigger the updating procedure, if required. Finally, an experimental evaluation of this framework is also performed using simulated and real data.

Keywords: Cyber-Physical Systems; reliability assessment; Internet-of-Things; LiDAR sensor; driving assistance; obstacle recognition; reinforcement learning; Artificial Intelligence-based modelling.

1. Introduction

Nowadays, knowledge of the most appropriate sensor operating conditions and fault detection systems are among the cornerstones of scientific and technical studies for automated systems [1]. These are based upon on-line monitoring processes and additional comprehensive interpretation of sensor data, by assessing sensor reliability. Sensors are driving the rapid growth of Cyber-Physical Systems (CPSs) and the Internet of Things (IoT). Both paradigms are behind the next generation of sensor networks and unpredictable future applications, meaning that sensor reliability has become one of the most important and desirable performance indicators in the design, implementation, and deployment of future sensor networks [2].

An important reliability-related issue to be detected in autonomous systems is the failure of one network element, in order to self-correct problems such as lost data packages, and data collision, among others [3]. One possible solution is to build real-time prediction models that maximize robustness and lifetime [4]. There are, in fact, several methods for the evaluation of sensor reliability. Each component that constitutes reliability or that might affect it can be assessed individually and as a whole, through a total error band figure. There are important features to be considered such as sensitivity, range, precision, resolution, accuracy, offset, linearity, dynamic linearity, hysteresis and response time. Evaluating sensor reliability includes probabilistic and statistical data that increase estimation reliability [5]. Evidence theory can be used, such as the Dempster-Shafer theory of belief functions. Quantifying reliability implies predictions concerning sensor lifetime and failure probability. Reliability can therefore be based on both statistical and Artificial Intelligence (AI) models. Suitable probability functions must be defined, which will be used to calculate the future behavior of devices, based either on carefully controlled laboratory experiments or on thorough failure analysis while in use. A typical product will be liable to various failure modes that change over time in a characteristic manner, so that the probability functions are themselves time dependent.

The most widely used techniques for modelling predictions concerning product lifetime and failure probability are probabilistic methods. Probabilistic methods for uncertain reasoning represent another group of techniques. Probability theory predicts events from a state of partial knowledge, while Fuzzy-Logic models are applied to situations with intrinsic vagueness and uncertainty.

However, the prediction techniques are hardly limited to those mentioned above. Several clustering techniques such as nearest neighbor methods have been explored, in order to enable self-detection and self-correction capabilities [6]. Other capabilities to be considered from the perspective of reliability are self-adaptation and self-organization by embedding artificial neural networks (ANNs) in CPSs [7]. Efficient performance of multiple sensors and their online monitoring and self-correction procedures, through the application of machine learning (ML) such as Support Vector Machines (SVM) and ANNs, are very important for the reduction of maintenance costs, risk minimization associated with uncalibrated and faulty sensors, increased instrument reliability and, consequently, extended equipment life [8, 9].

With the aim of guaranteeing certain safety and security conditions in some critical applications, the verification of sensory data and subsequent data evaluation are described in this paper through the simulation of virtual and real scenarios, as well as frameworks that properly combine both scenarios.

A reliability assessment procedure is therefore described in this paper that is applicable to data captured by IoT LiDAR sensors in automotive applications: LiDAR self-testing methodology. The reliability analysis is based on the paradigm of cyber-physical systems (CPS) by distributing nodes locally and globally, as will be explained later on. Each computing node has data-processing methods and machine-learning models for reliability prediction. In addition, a run-time self-learning and decision-making model runs within a global node, in order to determine the best model and the model updating mechanism on request.

The paper will be organized into five sections. Following this introduction, the second section will present a state-of-art review of the CPS-based reliability concept for sensor system reliability using AI methods. Subsequently, the specifications and the requirements obtained from the review of CPS reliability frameworks will be summarized in section 3. A particular implementation of a CPS-based co-simulation framework will also be proposed in this section. In addition, a case study for the evaluation of an IoT sensor network using a CPS-based co-simulation framework approach will be described in section 4. In that section, the experimental results and a discussion relating to a comparative study will also be addressed. Finally, the conclusions and future research steps will be presented in section 5.

2. CPS-based reliability approach

The truly challenging aspects of sensor network reliability and its evaluation have yet to prompt an exhaustive exploration and evaluation of sensory data under critical conditions. A gap that is addressed in this study through sensors incorporated in a CPS.

2.1. Sensor reliability assesment

One approach to sensor reliability in automotive applications is to design a model-based relationship between 'model parameters'. Those parameters can be derived from process monitors while 'functional parameters' refer to both the sensor characteristics and sensor lifetime, as well as cost aspects due to process yields (see Figure 1).

With the aim of increasing the reliability of data collected by LIDAR, metrological assessment procedures must also be applied. Linear interpolation of measurements from three detectors arranged in series is a time-saving procedure for processing and reducing LIDAR data [10].

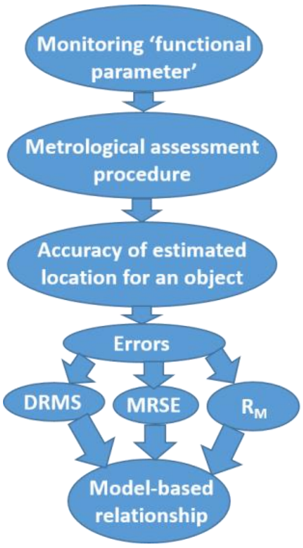

Monitoring 'functional parameter'
Metrological assessment procedure
Accuracy of estimated location for an object
Errors
DRMS
MRSE
R_M
Model-based relationship

Figure 1. Procedure for sensor reliability assessment using model-based relationship between sensor data and key performance indices.

All the major sources of potential error that could influence point positioning accuracy have to be considered in the analytical derivations, in order to determine the reliability of achievable point positioning accuracy of LiDAR systems. Csanyi, May and Toth provided some of the random errors that will be considered [11]. They also provided some formulas for point positioning accuracy that were derived from the LiDAR equation, via rigorous error propagation:

$$r_M = r_{M,INS} + R_{INS}^{M}(R_L^{INS} \cdot r_L + b_{INS}) \quad (1)$$

where, r_M represents the 3D coordinates of an object point in the mapping frame; $r_{M,INS}$ represents time dependent 3D INS coordinates in the mapping frame, provided by GPS/INS; $R_{INS}^{M}$is the time dependent rotation matrix between the INS body and the mapping frame; R_L^{INS} is the boresight matrix between the laser frame and the INS body frame; $r_L$represents the 3D object coordinates in the laser frame; and, b_{INS} is the boresight offset vector.

In addition, the calculation of the accuracy of the estimated location for an object using the LiDAR sensor can be performed by other key performance indices. For example, the use of the Distance Root Mean Squared (DRMS) measure for the data that are tracked on the x-y plane (2D) and the Mean Radial Spherical Error (MRSE) measure for the data that are tracked in the x–y–z space (3D) were reported in [12, 13]. Using derivable error formulas, any given random error and scan angle in the LiDAR range can be modelled and simulated. By doing so, the factors affecting LiDAR system accuracy can be analyzed [14].

2.2. Statistical and Artificial Intelligence-based methods

Bayesian and Hidden Markov models are the most widely applied for reliability assessment under fuzzy environments [15, 16]. A Bayesian network is a directed acyclic graph consisting of a set of nodes, representing random variables and a set of directed edges, representing their conditional dependencies. The dependencies in a Bayesian network can be adaptively determined from a dataset through a learning process. The objective of this training is to induce the network with the best description of the probability distribution over the dataset and can be categorized as an unsupervised learning method, because the attribute values are not supplied in the dataset [17].

In addition to those probabilistic methods, new tools are reported in the literature, highlighting the use of Artificial Intelligence (AI) techniques and in particular, Machine Learning (ML), to solve complex situations [18]. AI techniques also provide cognitive abilities, so that performance may be improved by increasing network life-time and reliability [19]. Some of those techniques are ANN and fuzzy inference system [20, 21]. Zhang et al. proposed a soft-computing system based on Genetic Algorithm-Support Vector Regression (GA-SVR), in order to facilitate the reliability and survivability of the Structural Health Monitoring (SHM) system faced, for example, with an invalid fiber link in the sensor network [22].

3. CPS-based co-simulation framework

Some factors that can affect CPS reliability are component failure, environmental effects, task changes, and network update. A strategy for testing the reliability of CPSs and for their evaluation is proposed in [23] by analyzing both the internal and the external factors that influence their reliability. One solution could be to evaluate each element that constitutes the system: testing hardware, software, and architecture, as well as performance reliability including service reliability, cyber security reliability, resilience & elasticity reliability, and vulnerability reliability.

Behavioral simulations of CPS and IoT assume importance as a method to analyze reliability, because the mathematical modeling of those factors is so difficult [24]. Those simulations are based on addressing four main topics: node localization, energy management, network multi-objective optimization, and self-capabilities approach [25, 26].

While the reliability evaluation of physical systems is well-understood and has been extensively studied, the reliability evaluation of a CPS is of greater complexity, because software systems will not degrade and follow a well-defined failure model in the same way as physical systems. An evaluation framework is therefore necessary, in order to assess the CPSs. A framework for CPS reliability analysis that includes reliability-based runtime reconfiguration is proposed in [27]. This framework is codified in a domain-specific modelling language that provides details on operational constraints and dependences.

However, domain-specific modelling-based analysis is, in many cases, unable to compute reliability functions efficiently (e.g. in terms of failure distributions) for complex systems. To do so, a frequency-domain reliability analysis framework of transportation CPSs was described in [28]. The advantage of that method is its capability to capture higher-order moments of the system characteristics, its scalability for the analysis of the reliability of complex systems, and efficient calculations.

In addition, it is important to consider the evaluation of other aspects of the CPS, such as safety and particularly security, different aspects of which have been focused upon over the past few years. Therefore, the design of the CPS framework must address those aspects at three levels: security objectives, security approaches and security in specific applications [29]. However, not only must the cyber part be secured, but also the physical part of possible threats. A multi-cyber (computational unit) framework was compared with traditional models to improve the availability of the CPS based on the Markov model. It was efficiently evaluated, in terms of availability, downtime, downtime costs, and the reliability of the CPS framework [30].

Finally, another work considered an Internet-based computing platform in the form of a global computing node. In [31], a new cloud security management framework was introduced, based on improving collaboration between cloud providers, service providers, and service consumers for the management of cloud platform security and the hosting services. In addition, although in some applications this will not be possible, it is important to consider the possibility of introducing the human factor in the reliability analysis procedure. A human-interactive Hardware-In-the-Loop

Simulation (HILS) framework for CPS was developed in [32] to support reliability and reusability in a fully distributed operating environment.

3.1. CPS-based co-simulation framework proposed

Based on the above contributions and considering the initial list of requirements from the previous section for the deployment of an IoT sensor network, a CPS-based co-simulation framework is proposed where an IoT sensor network will supply physical data and (local and global) computing nodes for processing the sensory data.

3.1.1. Conceptual Scheme

In addition, the IoT sensor network has a global or main node composed of a knowledge database, a Q-learning method for decision-making and an AI-based model library. During the simulation and the real running, a decision-making module will decide which specific model is the best in the current instant, taking into account the data received by all nodes that make up the network.

The functionalities are distributed in different nodes, both virtual and real, according to their functions. The distributed virtual or real nodes manage the capture of sensory data and run the error prediction calculation with the required accuracy, while the global or main node incorporates the runtime model that is generated, the library, and the knowledge database (see Figure 2).

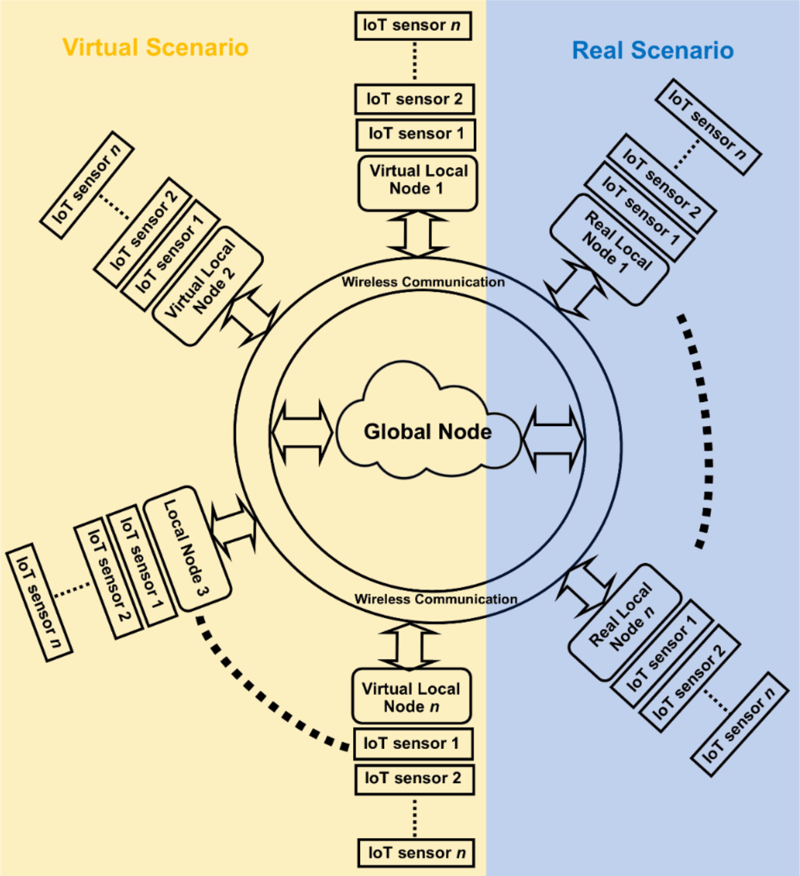


Figure 2. General scheme of the CPS-based co-simulation framework with virtual and real computing nodes and IoT sensor network.

The IoT sensors should be able to establish reliable and accurate wireless communications, ensuring that all the intrinsic challenges in an IoT network and in the different CPSs can be solved. It is achieved through the implementation of the architecture that is represented in Figure 2: a network of n nodes, each node having n IoT sensors. In addition, the computing nodes must communicate with each other and with their corresponding global node.

3.1.2. Procedure description

The framework is designed with the condition that both the real and the virtual (local) computing nodes must operate in parallel with the global computing node [33]. Data exchange between the different nodes takes place in two different ways. On the one hand, data exchange between local nodes is produced in both the virtual (3D model simulation tool) and the real scenario. On the other hand, there is the data exchange between different local nodes and the global node using the 802.11p wireless communication protocol.

There is therefore interaction between the software for both the simulated and the real environments, and external applications that are running in the main node. Figure 3 shows the schematic diagram of the exchange of information or messages within the co-simulation framework.

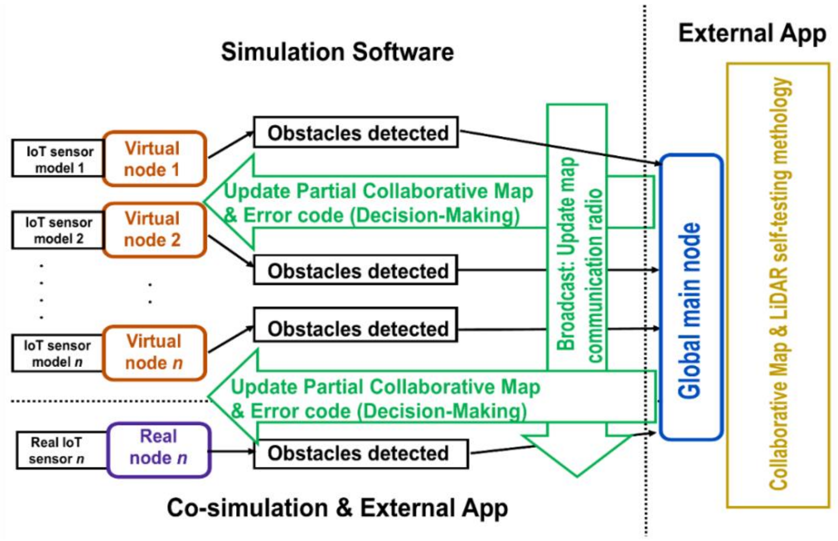

Simulation Software
External App
IoT sensor model 1
Virtual node 1
Obstacles detected
Update Partial Collaborative Map & Error code (Decision-Making)
IoT sensor model 2
Virtual node 2
Obstacles detected
IoT sensor model n
Virtual node n
Obstacles detected
Update Partial Collaborative Map & Error code (Decision-Making)
Real IoT sensor n
Real node n
Obstacles detected
Broadcast: Update map communication radio
Global main node
Collaborative Map & LiDAR self-testing methology
Co-simulation & External App

Figure 3. Conceptual flowchart showing the operation of a reliability co-simulation framework with CPS computing nodes and IoT sensors.

In the particular implementation that is described more accurately in the following section, there is a wireless exchange of messages between different nodes using the 802.11p communication protocol in the following way. First, the local nodes with their IoT sensors detect different objects and their respective properties. Secondly, this information is shared on the network through a broadcast process.

4. IoT LiDAR-based collaborative mapping – A case study

The IoT sensor network chosen to evaluate the CPS-based co-simulation framework is composed of virtual and real LiDAR sensors [34]. An Ibeo Lux LiDAR 4-layer sensor was used with the following specifications: horizontal field of 120 deg, horizontal step of 0.125 deg., vertical field of 3.2 deg., vertical step of 0.8 deg., range of 200 m, and an update frequency of 12.5 Hz. As previously mentioned, the sensor network to be evaluated is composed of IoT sensors. The sensor network therefore has IoT capabilities connected to its computing nodes. These nodes are on-board computers integrated in an autonomous vehicle with a wireless communication interface between them.

The particular implementation of the CPS-based co-simulation framework, the LiDAR-based collaborative map, is based on a co-simulation framework between two different software systems, both for the simulated part, designed in [35]. However, the contribution of this study is to include the real part in the co-simulation framework. This framework consists mainly of a computer-aided system to enable efficient interaction between the virtual scenario with virtual nodes setting in the Webots automobile simulation tool 8.6 [36] and an external application development for the computing nodes in the real scenario. The scenario in this particular case, in which the vehicles are represented as nodes, is as follows. A real vehicle (in a real scenario) and three virtual vehicles in the simulated scenario are detecting obstacles. Both kinds of vehicles share the position, object type and size of the obstacles (e.g., pedestrian, trees on the road and another vehicle). This is possible thanks to the IoT LiDAR network using an IoT obstacle detection application (see section 4.1), created in run time. Figure 4 shows the detailed diagram of the implementation of the LiDAR-based collaborative map using the CPS-based framework approach.

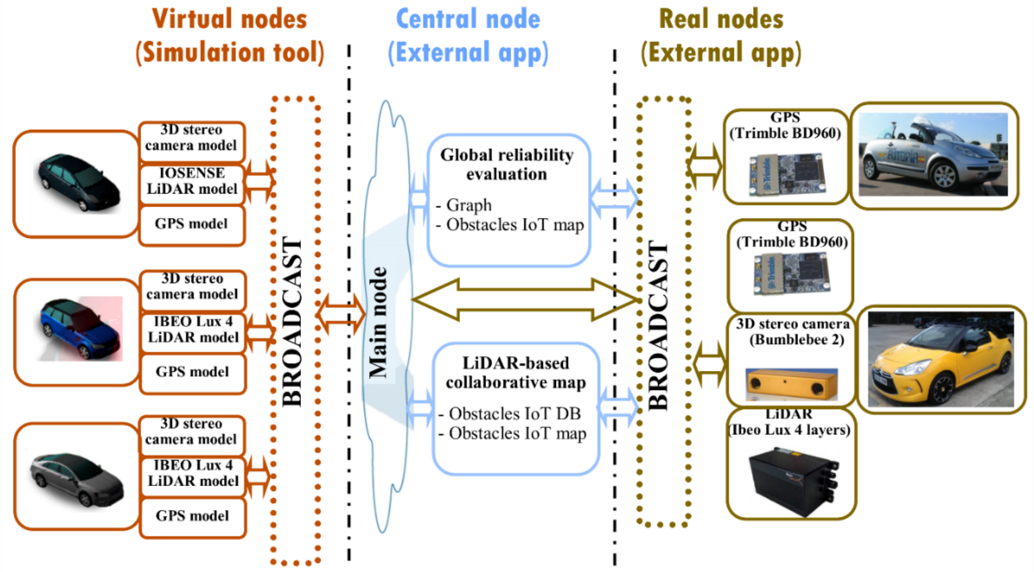


Figure 4. Detailed diagram of the implementation of the LiDAR-based collaborative map approached through a CPS-based co-simulation framework.

As previously mentioned above, the exchange of information packets between the local nodes with the main or global node is possible; thanks to the use of a communication protocol using a UDP (User Datagram Protocol) as the transport layer and a Wi-Fi 802.11p as the physical layer. The visualization of the co-simulated vehicle (real node) in the 3D virtual scenario in the Webots simulation tool is also possible. An example of the execution of the co-simulation architecture can be seen in Figure 5.

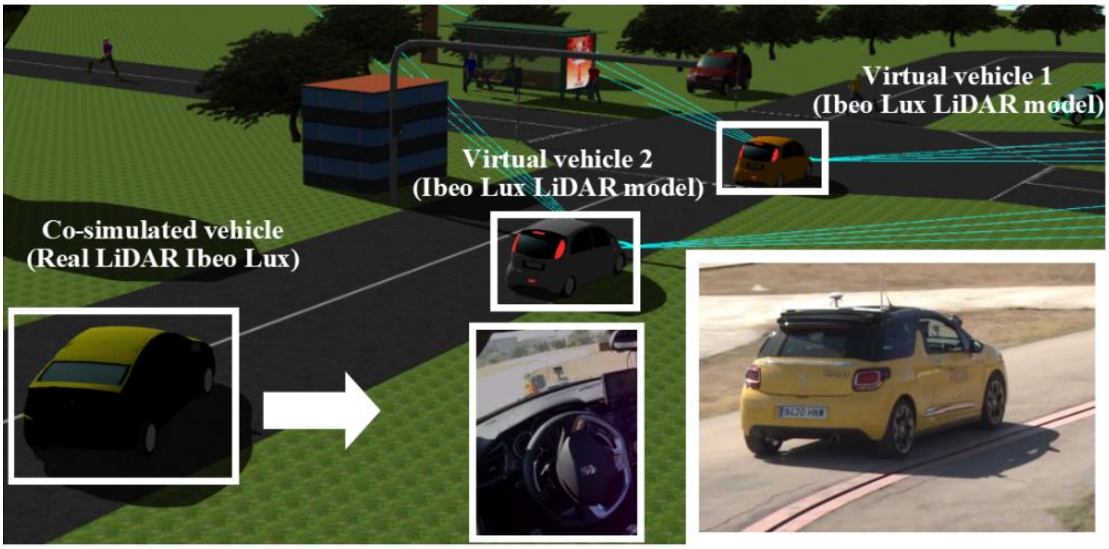
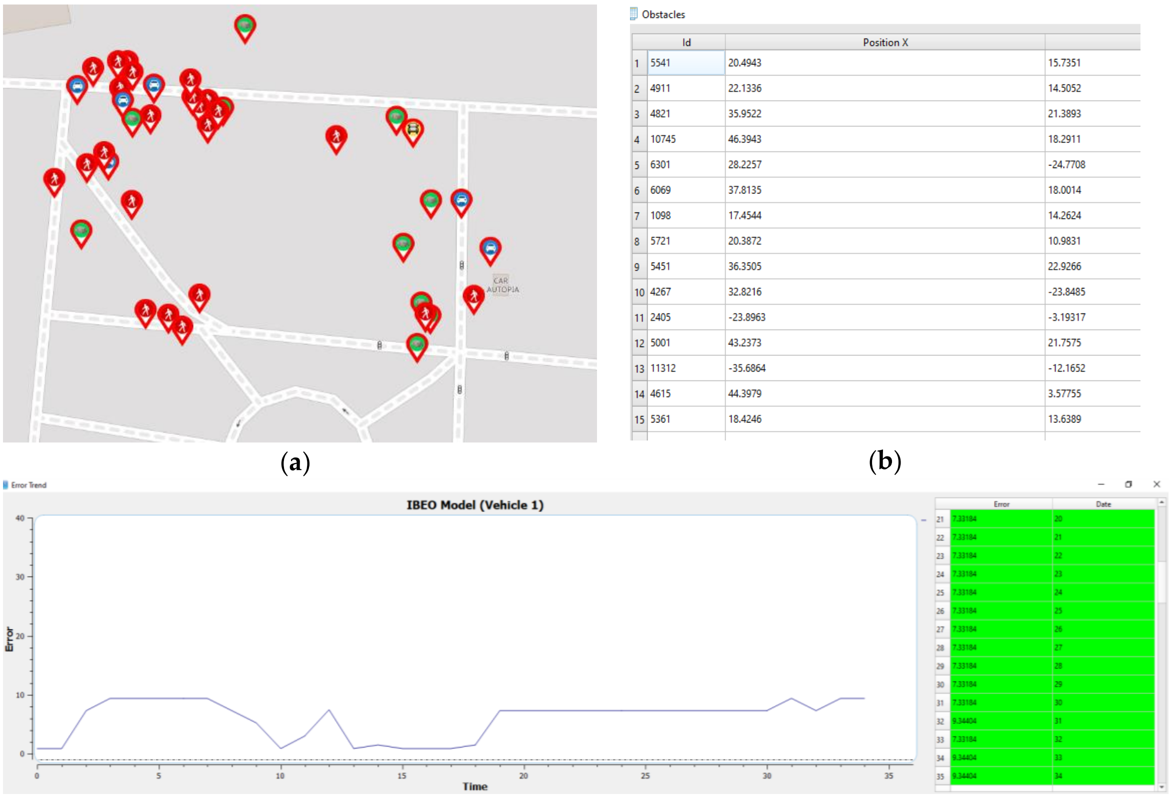


Figure 5. Data interchange between LiDAR sensors in (virtual and real) driving assistance scenarios in Webots for automobiles.

In addition, another application implemented in the IoT LiDAR-based collaborative map is the LiDAR self-testing methodology incorporated in each local computing node (autonomous vehicle), in order to evaluate the reliability of each IoT sensor in the network (section 4.2).

4.1 Obstacle detection in the IoT application

This framework is implemented in an external application; a development in Qt 5.10, that consists of an illustrated map updated in run time (see Figure 6 (**a**)) and a database with the information on both the virtual and the real objects that are detected (see Figure (**b**)). The information contains the position, object type, and size of the obstacles, to improve on the security/safety of the object detection process with a single sensor.



Obstacles

	Id	Position X	
1	5541	20.4943	15.7351
2	4911	22.1336	14.5052
3	4821	35.9522	21.3893
4	10745	46.3943	18.2911
5	6301	28.2257	-24.7708
6	6069	37.8135	18.0014
7	1098	17.4544	14.2624
8	5721	20.3872	10.9831
9	5451	36.3505	22.9266
10	4267	32.8216	-23.8485
11	2405	-23.8963	-3.19317
12	5001	43.2373	21.7575
13	11312	-35.6864	-12.1652
14	4615	44.3979	3.57755
15	5361	18.4246	13.6389

	Error	Date
21	7.33184	20
22	7.33184	21
23	7.33184	22
24	7.33184	23
25	7.33184	24
26	7.33184	25
27	7.33184	26
28	7.33184	27
29	7.33184	28
30	7.33184	29
31	7.33184	30
32	9.34404	31
33	7.33184	32
34	9.34404	33
35	9.34404	34

(**a**) (**b**)

(**c**)

Figure 6. (**a**) Collaborative mapping; (**b**) obstacles detected database; (**c**) LiDAR data for run-time accuracy error detection.

Figure 6 depicts the visual interface of the framework that has been developed. Specifically, the collaborative map is globally updated in the main computing node. A partial area of this updated map can also be sent at the request of a local node. A set of computational procedures is in charge of adapting and transferring sensory information from Webots, virtual nodes with the Ibeo Lux sensor model, and the real node, real vehicles with the real Ibeo Lux sensor; and vice versa.

4.2. LiDAR selft-testing application

The external application also includes a LiDAR data self-testing methodology using the AI-based error-prediction models. Figure 6 (**c**) shows the graphical interface that represents the estimated error with regard to time on the left-hand side. However, on the right-hand side the admissible error threshold is observed, which if exceeded, must be requested to make decisions over the best performance of each model at any given time. Specifically, the results are focused on showing the improved performance of the IoT sensor network composed of each CPS element with each

LiDAR sensor plus added IoT capabilities. To do so, a reliability prediction model dedicated to obtaining the accuracy error in obstacle detection is incorporated in each computing node.

4.2.1. Reliability prediction models

A reliability model was generated for each IoT LiDAR sensor, both virtual and real, that predicted the accuracy error for obstacle detection. The steps to follow for the determination of those models were extracted from the methodology described in [37, 38], with a set of different training data. In this study, a model-based procedure was used with a point-cloud clustering technique, in this case Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [39]. In addition, an error-based prediction model library was described, highlighting AI-based model techniques, such as, Multilayer Perceptron Neural Network (MLP), k-Nearest Neighbors (*k*-NN), and Linear Regression (LR). A difference in the particular implementation described in this paper is that, while k-NN, MLP and LR were maintained, SVM was added as a new technique to the AI-based library [40-42].

4.2.2. Model parametrization and validation

With the aim of determining which model training strategy based on AI provides the best reliability prediction model, an experimental validation was performed. The training dataset was composed of 998 scenes for the model training and 250 scenes for the model validation. All of them were obtained from a simulation procedure. The data input consisted of geospatial statistics [13, 43] which were extracted from the point cloud supplied by the LiDAR sensor, so that the models could generate the figures of merit in terms of accuracy error: DRMS and MRSE.

The four AI-based strategies that were considered are as follows. First, a multilayer perceptron neural network with backpropagation (MLP) with two hidden layers, each with five neurons and sigmoid activation functions, and an output layer with a lineal activation function, two neurons, and 5000 epochs. The initial value of the learning rate (μ) was 10-3 with a decrease factor ratio of 10-1, an increase factor ratio of 10, and a maximum μ value of 1010. The minimum performance gradient was 10-7. The training process stop criteria were as follows: the maximum number of epochs (repetitions); goal performance minimization; the performance gradient below a minimum gradient; or, a μ value in excess of the maximum value. The second modeling technique was *k*-nearest neighbors (*k*-NN), with $k = 5$ and using Euclidean distance as the distance function. The third was a lineal regression that was also obtained by minimizing the sum of squared differences between the predicted and the observed values. Finally, a support vector machine model was fitted by means of data standardization and the radial basis function kernel.

4.2.3. Self-learning-based decision-making. Q-learning algorithm

The global or main computing node executes several parallel procedures in a specific self-learning module that uses a Q-learning algorithm. On the one hand, a dataset for training by default is present in the global node. On the other hand, a knowledge database (warehouse) is also included, which can be updated in run time with the data provided by each local node. It sets up the self-learning strategy that is run, in order to analyze the best model behavior, when new traffic situations are generated providing new point clouds (environment information).

The local node, also in parallel, simulates the reliability model and when an error is admissible the threshold will exceed 20% in one the figures of merits (DRMS or MRSE), which will mean that the current model is failing. A request is therefore made to the global module to establish whether there is a model that is working better. The decision to select the best prediction model included in the library is taken by the self-learning decision-making at each instant, according to the generalization capability and the accuracy of the model. The particular performance metrics for each are R^2 and RAE, respectively. In summary, based on this continuous information flow and the previous prediction results (knowledge database), when a request from one of the local nodes is received and a new best model behavior is detected, the current error prediction model is then commuted, between MLP and k-NN, and vice versa.

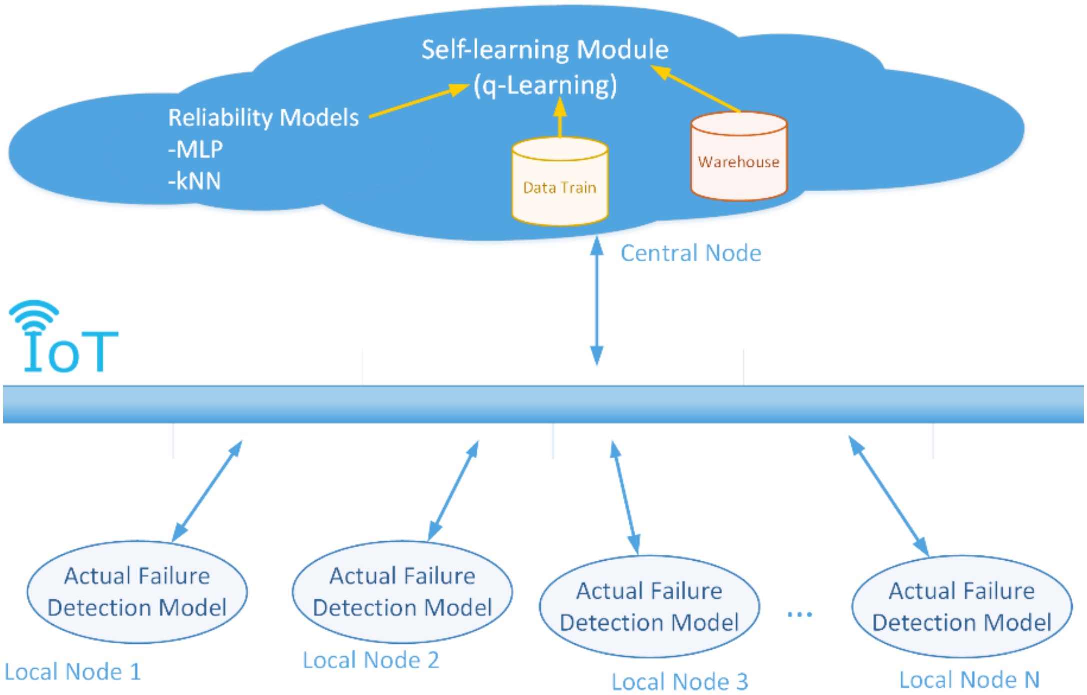


Figure 9. Flow diagram between the global node (self-learning module), IoT network, and local nodes (actual failure detection model).

5. Experimental results

5.1. Reliability model-based validation

Table 1. Key performance indices based on plane (DMRS) & space (MRSE) figure of merits.

Model	MAE		RMSE		RAE		RRSE		R^2	
	DMRS	MRSE	DMRS	MRSE	DMRS	MRSE	DMRS	MRSE	DMRS	MRSE
MLP	0.0046	0.0035	1.275	1.270	0.187	0.188	0.395	0.392	0.933	0.933
kNN	0.002	0.0002	1.014	1.010	0.114	0.114	0.371	0.365	0.963	0.961
LR	0.6781	0.6530	2.305	2.285	0.701	0.695	0.782	0.788	0.434	0.435
SVM	0.4735	0.4740	2.072	2.065	0.442	0.447	0.773	0.773	0.692	0.684

Table 1 shows the evaluation results obtained during the initial validation of each reliability model. Five error-based performance indices and two classification criteria were considered in the validation process: Mean Absolute Error (MAE); Root Mean Squared Error (RMSE); Relative Absolute Error (RAE); Root Relative Squared Error (RRSE); and, the coefficient of determination (R^2). Only, the models generated with *k*-NN and MLP returned R^2 results higher than 90%.

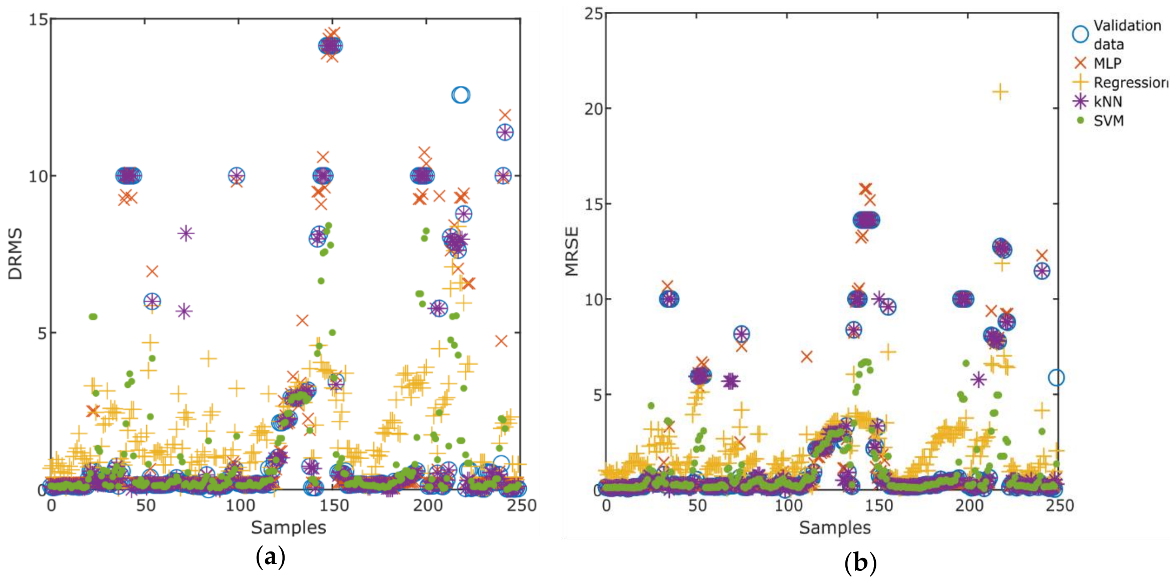


Figure 7. Behavior representation of LiDAR error on the plane for each model with regard to the validation data.

Figure 7 illustrates the behavior representation of the LiDAR error on the plane (DRMS) for each model (MLP, LN, KNN and SVM) with regard to the validation data. The AI-based modeling techniques that showed the best performing were MLP and KNN, according to the comparative study of the four modelling strategies, with a percentage improved performance comparable to the other two models of around 30%. Those model types will be chosen for the validation of the decision-making module.

5.1. Self-learning for decision-making evaluation

Finally, a simulation in order to determine the quality of the Q-learning method in the automatic selection of the best prediction model was performed. The reward function is chosen for setting the best possible Q-value in 100 different scenes. Therefore, the function to update the Q-values is [44]:

$$Q_{t+1}\left(s_t,a_t\right)=Q_t\left(s_t,a_t\right)+\alpha_t(s_t,a_t)(R_{t+1}+\gamma\max_{a\in A}Q_t\left(s_{t+1},a\right)-Q_t(s_t,a_t)) \quad (2)$$

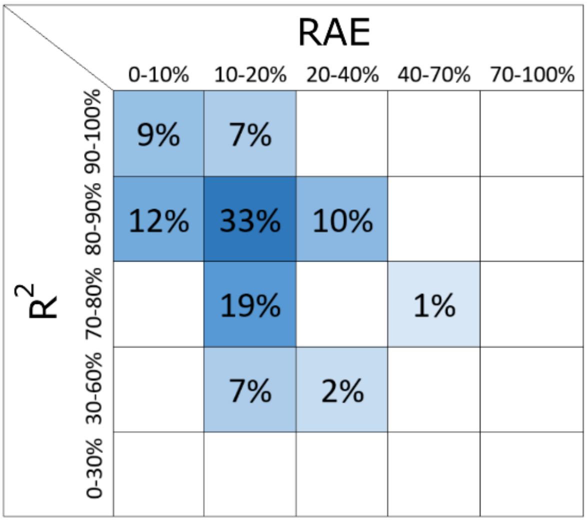
where, s_t is the state in time t; a_t is the action taken in time *t*; $R_{(t+1)}$ is the reward received after performing action a_t; α_t is the learning rate; and, γ is the discount factor which trades off the importance of sooner-versus-later rewards. Table 2 lists the error reward matrix based on knowledge of the behavior of those prediction models.

Table 2. Q-learning reward matrix for error admissible threshold.

R²	**RAE**				
	0 – 10%	**10 -20%**	**20 – 40%**	**40 – 70%**	**> 70%**
90 – 100%	1	0.9	0.8	0.5	0.2
80 – 90%	0.85	0.8	0.65	0.4	0.15
70 – 80%	0.7	0.6	0.5	0.3	0.1
30 – 60%	0.5	0.4	0.3	0.2	0.05
0 – 30%	0.3	0.2	0.15	0.1	0.01

The decision-making was based on two of the main performance indexes of model quality. First, the coefficient of determination (R²) was taken into consideration, as it provides a measure of the generalization capacity of the model. The Relative Absolute Error (RAE), which is a measure of model accuracy, was the second parameter.

Figure 9 shows the Q-learning classification error matrix. As previously shown, the best model had a RAE between 0 and 20% and a R² above 80% in 61% of the scenarios. The system was able to guarantee models with a greater capability of generalization in 71% of the scenarios, based on a coefficient of determination that was over 80%. In total, reliability can be predicted with an RAE of less than 40% and an R² of over 70% in 90% of the scenarios, which demonstrates the quality of the models. The models presented a low generalization, with a coefficient of determination of less than 70% in only 9% of the scenarios and a RAE greater than 40% in only 1%. Therefore, the Q-learning method that evaluates reliability on the basis of the prediction error model at each instant worked appropriately when determining the best model that represented the LiDAR performance to a high degree of accuracy and that guaranteed the required levels of safety and reliability for automotive applications.



R^2 \ RAE	0-10%	10-20%	20-40%	40-70%	70-100%
90-100%	9%	7%			
80-90%	12%	33%	10%		
70-80%		19%		1%	
30-60%		7%	2%		
0-30%					

Figure 9. Q-learning classification error matrix.

5. Conclusions

A method and an accompanying procedure have been presented in this paper for evaluating the reliability of IoT sensors in a CPS. A co-simulation platform has been designed for that purpose where virtual and real sensors can interact during run time through different simulations under appropriate safety conditions. The co-simulation framework was composed of distributed computing nodes within an IoT network, at both global and local levels.

A case study that consists of a LiDAR-based collaborative map has been proposed, in order to validate the CPS-based co-simulation framework. Real and virtual computing nodes with the corresponding sensors shared the position, object type, and size of the obstacles, to improve the security and safety of the autonomous driving when detecting objects with this framework in run time. The assessment of the proposed method was divided into two parallel procedures. First, at local level, each reliability model evaluated the condition of the IoT LiDAR sensor. Secondly, at a global level, a self-learning strategy for decision-making determined the most appropriate behavior of models in the reliability model library, also in run time. The Q-learning method was selected for this unsupervised self-learning strategy.

The comparative study of four strategies (MLP. SVM, k-NN and linear regression) in the reliability modelling library was then performed. In summary, the MLP and the k-NN methods outperformed the other two strategies considered in this study. Based on the previous results, a final experimental evaluation was presented, in order to determine the quality of the Q-learning method for automatically selecting the best reliability model. A Q-learning method evaluated the reliability models, in order to perform the analysis, in a simulation study with 100 different scenarios. Based on this procedure and the prediction results of the Q-learning method, when a request from one of the local nodes is received, a new model behavior is detected, and the current error prediction model is then commuted. Overall, all the reliability models performed very well, according to their generalization capability.

Therefore, the proposed CPS-based co-simulation framework has served to assess the performance of the IoT LiDAR network very accurately, guaranteeing safety and reliability in this autonomous driving case study. These promising results pave the way for future work that will validate the proposed method under real autonomous driving conditions.

Author Contributions: R.E.H., J.K. and S.S reviewed all technical and scientific aspects of the article. A.V. and F.C was in charge of the implementation of software application, the library models and the reinforcement learning algorithm. F.C. and A.V. designed and implemented the scenario, the external application and the LiDAR self-testing procedure, and drafted the paper.

Acknowledgments: This work was partially supported by the project Power2Power: Providing next-generation silicon-based power solutions in transport and machinery for significant decarbonisation in the next decade, funded by the Electronic Component Systems for European Leadership (ECSEL-JU) Joint Undertaking and the Ministry of Science, Innovation and Universities (MICINN), under grant agreement No 826417. In addition, this work was also funded by the Spanish Ministry of Science, Innovation and Universities through the project COGDRIVE (DPI2017-86915-C3-1-R). Preparation of this publication was also partially co-financed by the Polish National Agency for Academic Exchange (NAWA) through the project: "Industry 4.0 in Production and Aeronautical Engineering (IPAE)".

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

References

1. Beruvides, G.; Quiza, R.; Del Toro, R.; Haber, R. E., Sensoring systems and signal analysis to monitor tool wear in microdrilling operations on a sintered tungsten-copper composite material. *Sensors and Actuators, A: Physical* **2013,** 199, 165-175.
2. Kabashkin, I.; Kundler, J., Reliability of Sensor Nodes in Wireless Sensor Networks of Cyber Physical Systems. *Procedia Computer Science* **2017,** 104, 380-384.
3. Godoy, J.; Pérez, J.; Onieva, E.; Villagrá, J.; Milanés, V.; Haber, R., A driverless vehicle demonstration on motorways and in urban environments. *Transport* **2015,** 30, (3), 253-263.
4. Hao, X.; Wang, L.; Yao, N.; Geng, D.; Chen, B., Topology control game algorithm based on Markov lifetime prediction model for wireless sensor network. *Ad Hoc Networks* **2018,** 78, 13-23.
5. Guo, H.; Shi, W.; Deng, Y., Evaluating Sensor Reliability in Classification Problems Based on Evidence Theory. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **2006,** 36, (5), 970-981.
6. Zhang, H.; Liu, J.; Pang, A.-C., A Bayesian network model for data losses and faults in medical body sensor networks. *Computer Networks* **2018,** 143, 166-175.
7. Serpen, G.; Li, J.; Liu, L., AI-WSN: Adaptive and Intelligent Wireless Sensor Network. *Procedia Computer Science* **2013,** 20, 406-413.
8. Nsabagwa, M.; Mugume, I.; Kasumba, R.; Muhumuza, J.; Byarugaba, S.; Tumwesigye, E.; Otim, J. S. In *Condition Monitoring and Reporting Framework for Wireless Sensor Network-based Automatic Weather Stations*, 2018 IST-Africa Week Conference (IST-Africa), 9-11 May 2018, 2018; 2018; pp Page 1 of 8-Page 8 of 8.
9. Chuan, Y.; Chen, L., The Application of Support Vector Machine in the Hysteresis Modeling of Silicon Pressure Sensor. *IEEE Sensors Journal* **2011,** 11, (9), 2022-2026.
10. Rodrigues, C. E. D.; Da Silva, R. P. M.; Nunes, R. A., Metrological assessment of LIDAR signals in water. *Measurement: Journal of the International Measurement Confederation* **2011,** 44, (1), 11-17.
11. Yang, X.; Quan, Z.; Dan, J., The effect of INS and GPS random error on positioning accuracy of airborne LIDAR system. *Journal of Computational Information Systems* **2011,** 7, (11), 3795-3802.
12. Castaño, F.; Beruvides, G.; Villalonga, A.; Haber, R. E., Self-tuning method for increased obstacle detection reliability based on internet of things LiDAR sensor models. *Sensors (Switzerland)* **2018,** 18, (5).
13. Maalek, R.; Sadeghpour, F., Accuracy assessment of ultra-wide band technology in locating dynamic resources in indoor scenarios. *Automation in Construction* **2016,** 63, 12-26.
14. Dan, J.; Yang, X., Modeling of range and scan angle random errors from airborne LIDAR. *Journal of Computational Information Systems* **2013,** 9, (7), 2747-2753.
15. Wu, H. C., Bayesian system reliability assessment under fuzzy environments. *Reliability Engineering and System Safety* **2004,** 83, (3), 277-286.
16. Wu, H. C., Fuzzy Bayesian system reliability assessment based on exponential distribution. *Applied Mathematical Modelling* **2006,** 30, (6), 509-530.
17. Du, K. L.; Swamy, M. N. s., Probabilistic and Bayesian Networks. In 2014; pp 563-619.
18. La Fe-Perdomo, I.; Beruvides, G.; Quiza, R.; Haber, R.; Rivas, M., Automatic Selection of Optimal Parameters Based on Simple Soft-Computing Methods: A Case Study of Micromilling Processes. *IEEE Transactions on Industrial Informatics* **2019,** 15, (2), 800-811.
19. Gheisari, S.; Meybodi, M. R., A new reasoning and learning model for Cognitive Wireless Sensor Networks based on Bayesian networks and learning automata cooperation. *Computer Networks* **2017,** 124, 11-26.

20. Haber, R. E.; Alique, J. R., Nonlinear internal model control using neural networks: An application for machining processes. *Neural Computing and Applications* **2004,** 13, (1), 47-55.

21. Ramírez, M.; Haber, R.; Peña, V.; Rodríguez, I., Fuzzy control of a multiple hearth furnace. *Computers in Industry* **2004,** 54, (1), 105-113.

22. Zhang, X.; Liang, D.; Zeng, J.; Asundi, A., Genetic algorithm-support vector regression for high reliability SHM system based on FBG sensor network. *Optics and Lasers in Engineering* **2012,** 50, (2), 148-153.

23. Li, Z.; Kang, R. In *Strategy for reliability testing and evaluation of cyber physical systems*, IEEE International Conference on Industrial Engineering and Engineering Management, 2016; 2016; pp 1001-1006.

24. Vo, M.-T.; Thanh Nghi, T. T.; Tran, V.-S.; Mai, L.; Le, C.-T. In *Wireless Sensor Network for Real Time Healthcare Monitoring: Network Design and Performance Evaluation Simulation*, Cham, 2015; Springer International Publishing: Cham, 2015; pp 87-91.

25. Cacciagrano, D.; Culmone, R.; Micheletti, M.; Mostarda, L., Energy-Efficient Clustering for Wireless Sensor Devices in Internet of Things. In *Performability in Internet of Things*, Springer: 2019; pp 59-80.

26. Hossain, S.; Fayjie, A. R.; Doukhi, O.; Lee, D.-j. In *CAIAS Simulator: Self-driving Vehicle Simulator for AI Research*, International Conference on Intelligent Computing & Optimization, 2018; Springer: 2018; pp 187-195.

27. Nannapaneni, S.; Mahadevan, S.; Pradhan, S.; Dubey, A. In *Towards Reliability-Based Decision Making in Cyber-Physical Systems*, 2016 IEEE International Conference on Smart Computing, SMARTCOMP 2016, 2016; 2016.

28. Liu, X.; He, W.; Zheng, L. In *Transportation cyber-physical systems: Reliability modeling and analysis framework*, National Workshop for Research on High-Confidence Transportation Cyber-Physical Systems: Automotive, Aviation and Rail. November, pp 18-20.

29. Lu, T.; Zhao, J.; Zhao, L.; Li, Y.; Zhang, X., Towards a framework for assuring cyber physical system security. *International Journal of Security and Its Applications* **2015,** 9, (3), 25-40.

30. Parvin, S.; Hussain, F. K.; Hussain, O. K.; Thein, T.; Park, J. S., Multi-cyber framework for availability enhancement of cyber physical systems. *Computing* **2013,** 95, (10-11), 927-948.

31. Almorsy, M.; Grundy, J.; Ibrahim, A. S. In *Collaboration-based cloud computing security management framework*, Proceedings - 2011 IEEE 4th International Conference on Cloud Computing, CLOUD 2011, 2011; 2011; pp 364-371.

32. Kim, M. J.; Kang, S.; Kim, W. T.; Chun, I. G. In *Human-interactive hardware-in-the-loop simulation framework for cyber-physical systems*, 2013 2nd International Conference on Informatics and Applications, ICIA 2013, 2013; 2013; pp 198-202.

33. Artuñedo, A.; Del Toro, R. M.; Haber, R. E., Consensus-based cooperative control based on pollution sensing and traffic information for urban traffic networks. *Sensors (Switzerland)* **2017,** 17, (5).

34. Godoy, J.; Haber, R.; Muñoz, J. J.; Matía, F.; García, Á., Smart sensing of pavement temperature based on low-cost sensors and V2I communications. *Sensors (Switzerland)* **2018,** 18, (7).

35. Castaño, F.; Beruvides, G.; Haber, R. E.; Artuñedo, A., Obstacle Recognition Based on Machine Learning for On-Chip LiDAR Sensors in a Cyber-Physical System. *Sensors* **2017,** 17, (9), 2109.

36. Michel, O., Cyberbotics Ltd. Webots™: professional mobile robot simulation. *International Journal of Advanced Robotic Systems* **2004,** 1, (1), 5.

37. Castaño, F.; Beruvides, G.; Villalonga, A.; Haber, R. E., Self-Tuning Method for Increased Obstacle Detection Reliability Based on Internet of Things LiDAR Sensor Models. *Sensors* **2018,** 18, (5), 1508.

38. Castaño, F.; Beruvides, G.; Haber, R. E.; Villalonga, A., Time-To-Failure Modelling in On-Chip LiDAR Sensors for Automotive Applications. *Proceedings* **2017,** 1, (8), 809.

39. Huang, F.; Zhu, Q.; Zhou, J.; Tao, J.; Zhou, X.; Jin, D.; Tan, X.; Wang, L., Research on the Parallelization of the DBSCAN Clustering Algorithm for Spatial Data Mining Based on the Spark Platform. *Remote Sensing* **2017,** 9, (12), 1301.

40. Banerjee, T. P.; Das, S., Multi-sensor data fusion using support vector machine for motor fault detection. *Information Sciences* **2012,** 217, 96-107.

41. Niu, G.; Yang, B.-S.; Pecht, M., Development of an optimized condition-based maintenance system by data fusion and reliability-centered maintenance. *Reliability Engineering & System Safety* **2010,** 95, (7), 786-796.

42. Castaño, F.; Beruvides, G.; Villalonga, A.; Haber, R. E., Computational Intelligence for Simulating a LiDAR Sensor. In *Sensor Systems Simulations*, Springer: 2020; pp 149-178.

43. Premebida, C.; Ludwig, O.; Nunes, U. In *Exploiting LIDAR-based features on pedestrian detection in urban scenarios*, 2009 12th International IEEE Conference on Intelligent Transportation Systems, 4-7 Oct. 2009, 2009; 2009; pp 1-6.

44. Beruvides, G.; Juanes, C.; Castaño, F.; Haber, R. E. In *A self-learning strategy for artificial cognitive control systems*, 2015 IEEE 13th International Conference on Industrial Informatics (INDIN), 22-24 July 2015, 2015; 2015; pp 1180-1185.