UNIVERSITY POLITEHNICA OF BUCHAREST
FACULTY OF INDUSTRIAL ENGINEERING AND ROBOTICS

MAINTENANCE PLATFORM IN INDUSTRY 4.0

\- *Abstract* -

PhD Student
Georgescu (Deac) Crina Narcisa

Scientific Coordinator
Prof. Dr. Eng. Costel Emil COTEȚ

Content

Abstract

Following the new industrial revolution (Industry 4.0), we find ourselves in a period characterized by the automation, digitization, and interconnection of components in production processes and their integration into cyber-physical systems (CPS) capable of managing production in a flexible, efficient, and environmentally friendly way. These systems also support advanced maintenance strategies, providing real-time information and predictive capabilities for asset management. The configuration of the predictive solutions depends on the case studied, and the equipment used and involves a continuous improvement of them depending on the response from practice over a period. Data prediction in predictive maintenance, through Machine Learning (ML) and Deep Learning (DL), belongs to the field of "data science", is in continuous evolution and research and requires, in the same measure, continuous training of specialists in data study.

There are currently too few data-science specialists trained in statistical analysis and data prediction, especially in the industrial field. Their work must be closely linked to the work of industry specialists, as the information collected by analysts depends to a large extent on collaboration with industry engineers.

Regarding the maintenance activity, at present, in Romania, a large part of the industrial enterprises implements at most the preventive maintenance, being too little aligned with the requirements of Industry 4.0. Predictive maintenance reduces maintenance costs and the cost of purchasing spare parts, improving production quality, but requires specialized hardware and software. Current solutions on the market are still extremely expensive, few and far between and come from large multinational companies.

Too few companies store data from machines in the cloud for later management using software tools that analyse, report, forecast, or coordinate production. It is, therefore, necessary to develop CMMS (Computerized Maintenance Management System) software tools for maintenance management.

The objectives of the doctoral thesis

Starting from the analysis of the context of the fourth industrial revolution and of the existing software solutions on the market for the maintenance applied with the help of CMMS systems, IIoT devices and predictive mathematical models applied on big data (Big Data), the thesis aims to address in a broad context all types of maintenance that can be applied in these systems.

The main objective of this paper is to create a CMMS system - a maintenance platform - which is intended to be a flexible, modular, extensible tool so that it can manage all types of maintenance within an industrial enterprise: corrective, preventive, maintenance based on status monitoring, predictive maintenance, depending on the company's maintenance strategies, in one or more production locations, in a multilingual system. The platform will also allow coordination with the other departments in the company, such as the procurement department, the IT department, the production department, etc., so that the maintenance processes and implicitly the production processes are optimized in an integrated system. This main objective assumes the development of an e-maintenance software tool that can be implemented in both SaaS (Software as a Service) and on-premise mode (the software application is installed on the servers of the company that uses it) and that is addressed to all enterprises. using automated equipment and production lines in the production and processing of materials.

To achieve this main objective was necessary to address some **secondary endpoints** that include a series of theoretical analysis and conceptual developments as follows:

- the study of systems architecture cyber-physical identifies the elements to be connected to the platform
- study the concepts of maintenance applied in the industry 4.0 and the selection of those useful for the operation of the platform.
- the study of anomaly detection techniques by statistical methods and artificial intelligence

and the choice of those that will be integrated into the functions of the platform.
- identification of techniques for diagnosing defects in components / equipment and methods for predicting failure by predictive algorithms of Machine Learning and Deep Learning suitable for the proposed solution.
- selection of algorithms for predicting the remaining useful life of the appropriate equipment for the designed platform.
- the study of the predictive analysis and maintenance platforms existing on the market and the highlighting of their limitations.

From the perspective of the design and effective development of the proposed platform as the main objective, several secondary objectives are also defined:
- implementation of a monitoring and data storage system through IoT technology.
- implementation of a remote assistance system for maintenance technicians.
- monitoring the status of assets and expenses related to maintenance interventions.
- implementation of a configurator for equipment / machines
- implementation of a configurator for scheduled maintenance operations
- implementation of a scheduler for scheduled maintenance operations
- implementation of a ticketing system between company departments
- implementation of a library of ML and DL models, built and refined over time depending on the cases in industry
- implementation of a training library for maintenance technicians, built over time, using the CMMS system.
- building technical libraries that can be easily accessed by technicians or engineers in scheduled or unscheduled maintenance operations.

The structure of the doctoral thesis

The thesis is structured in 6 chapters:

In Chapter 1 are presented the field of the thesis, the motivation of the research topic and the objectives of the doctoral thesis.

Chapter 2 analyses the structure and components of a CPS system necessary for the design and development of the maintenance management platform and identifies the main types of maintenance that should be addressed within the platform.

Relevant aspects are addressed regarding the
- concept of the smart factory and its implementation according to the automation pyramid.
- the evolution of predictive maintenance and intelligent production information systems.
- the cyber-physical system for maintenance management in the context of Industry 4.0, in the 5C (CPS), 7C (CPSS) architecture and its evolution (D-CPSS).
- a summary of the history of the four generations of maintenance and their techniques.
- the evolution of the maintenance concept: reactive, preventive maintenance, maintenance based on condition monitoring, predictive maintenance, total productive maintenance, maintenance focused on reliability, maintenance performed with CMMS systems.
- introduction of the concept of PHM (Prognostics and Health Management).
- comparison between the maintenance concepts presented.

Chapter 3 explores several algorithms that could be used in the platform for operating condition management and equipment and component forecasting. A series of algorithms are presented for:
- detecting the anomaly: by multivariate statistical analysis, by reducing the dimensionality with PCA analysis, using artificial neural networks with supervised and unsupervised learning.
- model-based fault diagnosis using methods based on parameter identification, methods based on parity equations, methods based on observers, methods based on neural networks for residual generation.
- data-based diagnostics using classification techniques using traditional Machine Learning

algorithms: decision trees, vector-supported machines, k-Nearest Neighbour algorithm, artificial neural networks.

- data-based diagnostics using classification techniques using Deep Learning algorithms, such as autoencoders, CNN convolutional neural networks, recurrent RNN neural networks, hybrid networks.
- prognosis of defects and remaining useful life of components / equipment (RUL).
- forecasting techniques - experience-based forecasting approach through: AI methods of Deep Learning, survival models, stochastic filtering, methods of similarity between degradation trajectories.
- forecasting techniques - data-driven forecasting approach through linear and exponential degradation models, AI methods of Deep Learning.
- forecasting techniques - model-based forecasting approach by recursive estimators.

Chapter 4 discusses several predictive analytics and maintenance platforms available on the market: Dataiku, RapidMiner, IBM Applications, HP Haven Predictive Analytics, Microsoft Azure Machine Learning, H2O.ai, Data Robot, SAP HANA XSA, SAP Predictive Maintenance and Service, Siemens Mind Sphere, GE Predix. This study was conducted to identify the current state and possible research directions for the development of the IOTIA platform.

In Chapter 5, the main objective of the thesis is achieved by presenting the concept and development of the platform. This chapter includes:

- Introduction to systems CMMS, with which the current context in terms of maintenance and implementation of this tool e-maintenance
- concept platform, are in-line modules platform:

Interface Dashboard Module Configurator Module Administration, Preventive Maintenance Module, Mobile Maintenance Application Using AR Technology (and QR Code Implementation), Monitoring Module, Prediction Module, in which two case studies were presented - prediction of component failure in a time window, using algorithms multiclassification (Random Forest, Gradient Boosting, XGBoost, Naïve Bayes, Neural Network) and bearing fault diagnosis by vibration analysis using the Deep Learning approach (CNN, RNN, RNN CNN, LSTM CNN, LSTM)

Chapter 6 presents the final conclusions that are detached from the theoretical and applied aspects that competed in the development of the proposed platform, synth highlighting the original contributions and mentioning the prospects for further development. The list of publications during the doctoral studies is also reviewed.

IOTIA Maintenance Platform - CMMS for maintenance management in industry 4.0

The IOTIA (Internet of Things Industrial & Automation) platform is a pilot project, its purpose is to be a tool for managing all types of maintenance and to be applied to several types of industry production. The platform is scalable and can be configured in the case of companies that own several factories.

The platform structure is modular (Fig.1) and allows:

- in the section *Configurator*, defining factories, configuring assets (components, subassemblies, machines) and maintenance operations in the Planner, defining errors in the Error Glossary, defining suppliers, consumables and setting vocabulary and languages.

- in the section *Settings*, define company staff, define user accounts, set permissions and roles, configure the ticketing system for support, and other general settings.

- in the section *Administration,* instantiation and administration of production lines and their associated machines, planning and scheduling of interventions in the Intervention Generator, definition of personal training sessions.

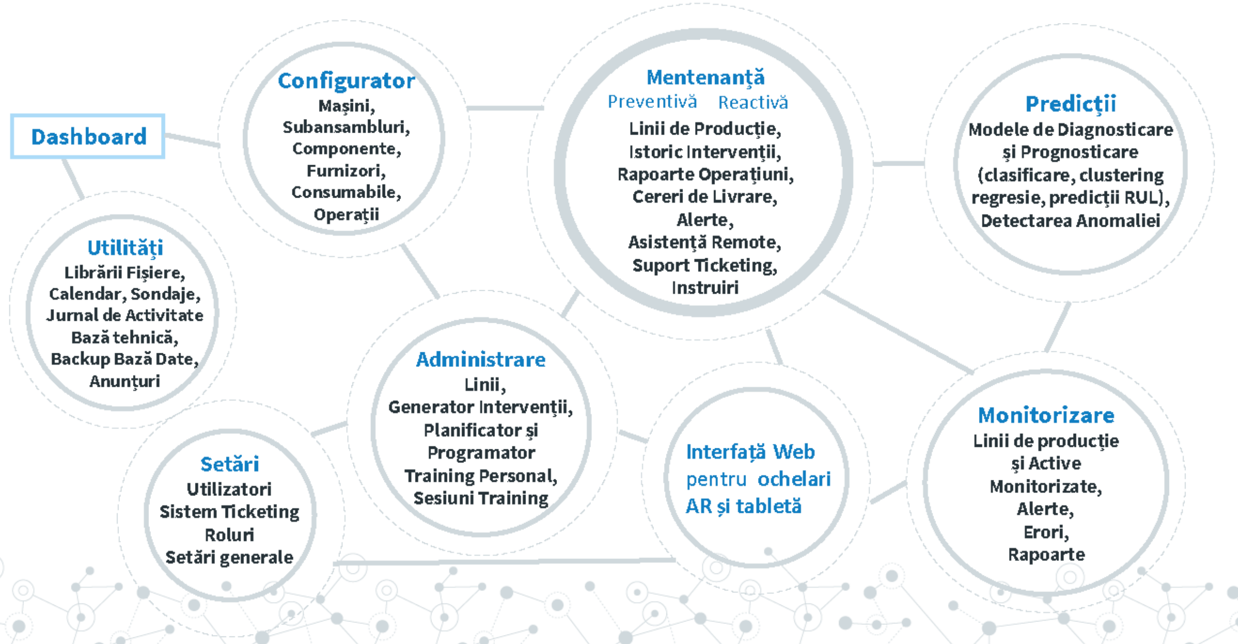

Fig. 1. Modules in the IOTIA Platform

- in the *Maintenance* section, management of interventions and preventive and reactive maintenance operations. The section manages maintenance operations for each logged-in user as a maintenance technician or director, component or consumable delivery requests, alerts, remote assistance via a video conferencing system, ticketing support, training reports,

- in the *Web interface for AR glasses and tablet* section, operation of maintenance by on-site maintenance technicians using AR maintenance application, using augmented reality technology and mobile devices,

- in the *Monitoring* section, monitoring of assets, subassemblies, and components on production lines,

- in the section *Predictions, the* implementation of predictive models for fault detection and forecasting, using classification, regression algorithms or algorithms for the detection of vibration anomalies; the models are associated with the assets / components monitored within the platform and are included in Jupiter Notebook files,

- in the *Utilities section*, defining public and private file libraries (defined by users), building the technical standardized environment, technical documentation on maintenance operations, daily activity log of platform users, ad configurator and database backup options.

- in the *Dashboard*, the control panel with navigation and summary view of maintenance and custom activity statistics, depending on the logged in user.

The platform uses a *Model-View-Controller approach* and is a web-based platform.

Dashboard

The control panel (*Dashboard*) is the introductory panel that is configured according to the connected user with various roles or access rights: wider for administrators, general manager, IT director, maintenance managers, or smaller, for maintenance technicians, procurement operators, specialists coordinating maintenance operations, operating in one or more locations or technical support assistants, defined within the departments.

The control panel is a summary and a link to the sections of the platform. Displays a schedule of scheduled maintenance operations with reference to operation, event, or reminder sheets for specific personal tasks set by the user or system administrator. The list of pending tickets from various departments (maintenance, procurement, IT), the status of the latest requests to the purchasing department for spare parts (components) and ordering and delivery information is highlighted.

Links to monitored assets, the latest announcements from the factory departments (management / production / maintenance) and a list of actions are displayed *To do* entered by the logged in user for better management of his activities.

Configurator Module

The Configurator contains 9 configuration subsections as seen in the menu in Fig.2:

- defining platform languages,
- defining components,
- defining subassemblies,
- defining machines,
- defining suppliers,
- defining consumables,
- defining maintenance operations - the planner,
- defining the error glossary,
- defining the factories.

The platform is initiated in 6 languages but can be extended by adding other language modules.

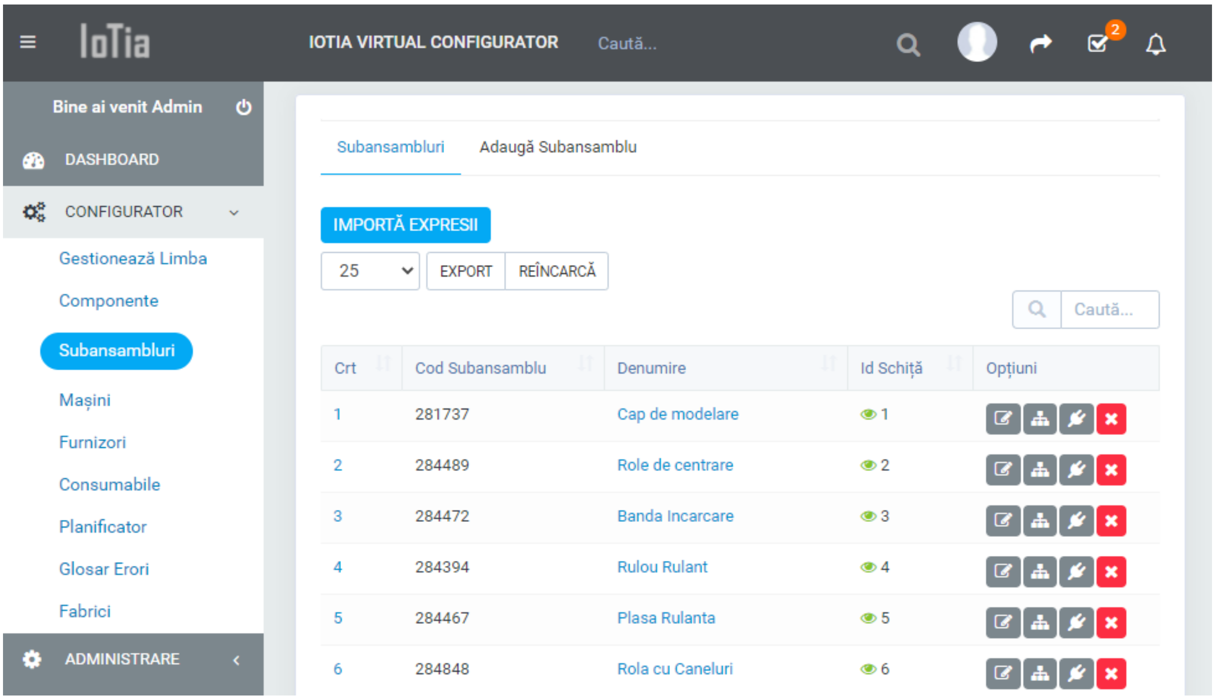

Fig.2. Configurator - Subassemblies

The first subsection - *Manage Language* allows you to insert new languages and edit sentences in all defined languages, using a CSV import procedure for the expressions to be translated, but also a local translation procedure. This feature facilitates the operation of the platform in a multicultural organization.

The second subsection - *Components* - initiates the component library. All component codes that make up the machines used in the factory according to the documentation in the manufacturer's technical books, together with other characteristics of the component, as well as a library of technical files.

The service life in hours is determined by the manufacturer's data and is recalculated based on component replacement data, as an average of the operating hours between two successive replacements and subsequently adjusted by predictive models.

After initiating the entire list of components that will be subjected to the maintenance process (planned or incidental replacement), the third subsection of the configurator, *Subassemblies*,

defines all subassemblies that make up the machines at the factory, according to the subassembly code provided by the manufacturer and the position index in the asset outline.
Each subassembly will in turn have a specific file library defined, representing multiple views of the subassembly, in sketches, available to maintenance technicians on mobile devices. The sketches show the components marked with a position indicator.

The *Machinery* section initiates the structure of all machines / equipment in the factory subject to reactive / planned or monitored maintenance activities in the cloud. Each asset is constructed from the structures of the subassemblies and components originally created.
Suppliers are defined in the next subsection, and in the sixth section the consumables indicated by the manufacturer are defined, each consumable being associated with a list of subassemblies. Thus, when scanning the AR (or QR) subassembly code, the operator can easily check the consumables indicated in the maintenance operations.

The *Scheduler* section, in the Configurator, configures with the option *New Operation*, for each subassembly (subassembly code), all the maintenance operations associated with it and existing in the technical documentation of the asset.

In addition to the maintenance operations specified in the documentation for lubrication, control, cleaning, setting, etc., with the options *Generate operations at break down* or *Generate Planned Replacements*, the Configurator administrator will generate a generic *"break-down component replacement"* operation for each component in the subassembly that needs to be replaced in the event of a breakdown and the generic operation *"planned component replacement"* for the components of the subassembly that require regular replacement (this operation has as frequency - the estimated number of hours of operation of the component).
In the name of the component replacement operation, the generator script *Scheduler - Generate Planned Replacements* - will include the component code and positioning index in the subassembly diagram for quick identification.

Once all subassembly operations have been initiated, when the actual production lines and machines are instantiated, planning for maintenance operations in *the Intervention Generator* / Administration section becomes an easier task. This is because all operations are automatically instantiated, based on the frequency set in the *Scheduler*, by a Cron script or instantiated in case of emergency maintenance, following the inspection of the maintenance technician and the validations given by the maintenance director.

The maintenance operation sheet contains the description of the procedure, the operating frequency in hours, the warning window in hours, a file library (which supports any extension accepted in the Settings section) available to operators during the intervention and all consumables allowed for the subassembly. The descriptive files of the operation procedure (in pdf or video format) will be available in the tablet application, in the Technical Library associated with the operation, when the maintenance technician scans the subassembly code, in the stage of performing the maintenance operation.

The configurator also records an error glossary (mechanical, electrical), in which the authorized user sets all the error codes defined on the subassembly / machine code. In this way, in the maintenance operation sheet, the technician will be able to add the error code that triggered the event, this being also valuable later, in prediction algorithms, in estimating the remaining RUL life of a component.

The error code is accompanied in the library with information such as potential causes and ways to fix the error, which can help maintenance technicians during interventions. The last subsection of the Configurator configures the company's factories, with contact details, locations, and representatives.

This section creates and manages the actual factory entities: production lines, production line machines, subassemblies, and components, based on the Configurator,
all of which are assigned to a factory and uniquely coded by an ID.

The *Lines subsection* displays the production lines for one or more factories, and by selecting a line you get access to a tree view with the list of machines, subassemblies and their components by suggestive buttons.
The screen of a production line displays the assets to be maintained, and the status of the machine (*Available / Pause / Maintenance / Unavailable*) is read from the PLCs and displayed in suggestive colours. Each machine instantiated on the line is accompanied by an AR (or QR) code.
The IOTIA platform has two implementation options for scan codes associated with machines and subassemblies: the AR version with ArUco codes and the version that implements QR codes. The first option has the advantage of AR technology to over-augment virtual content when scanning but has the disadvantage that it offers a limited number of markers / codes, only 2 ^ 10 = 1024 ArUco codes. This may limit the solution for industrial enterprises with hundreds of machines and thousands of subassembly codes, while the QR version is more versatile.

An ArUco code is a black and white 5x5 grid in which the first, third, and fifth columns represent parity bits. The second and fourth columns represent data bits. Therefore, there are only ten bits of data in total.

By scanning the AR / QR code, via mobile devices or using the desktop application, with the suggestive buttons you can reach a deeper view of the situation of subassemblies and related components (Fig.3), respectively of interventions on the selected asset and a view of monitored car assembly. In Figure 3, selecting a subassembly displays the history of maintenance operations, the history of component replacements, the history of component orders, and the attached file library in the Configurator.

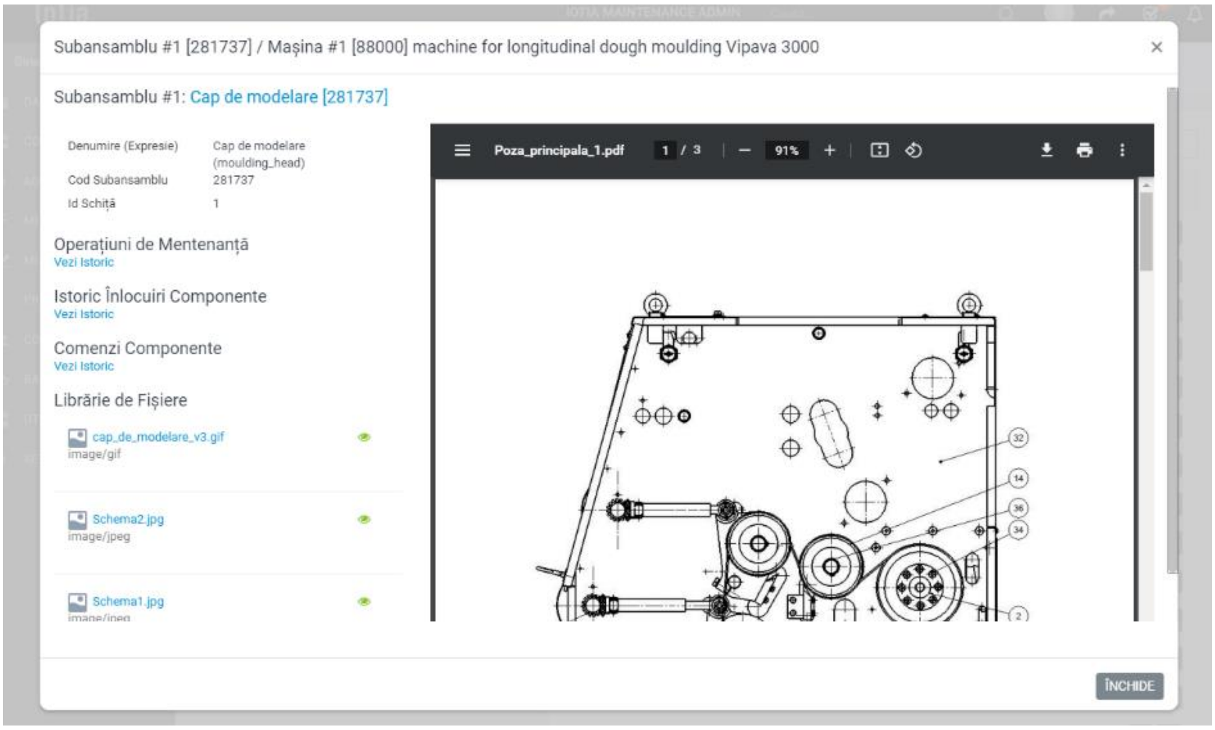

Fig. 3. Administration - Subassembly Sheet: Modelling Head (Subassembly Code: 281737)

The replacement history of the components of a subassembly will be used by the predictive model, both for planned and unplanned replacements, when severe degradation or damage occurs. In the first case -- the *Scheduled Component Replacement* history is useful for building the feature: the *number of days of component operation since the last replacement*, and in the second case -- the *Component Break-down Replacement* history is used to build the target value: *component failure*, for labelling the “failure” data of the components.

Another important feature for the predictive model, the RUL estimate of the time remaining until the component fails, is the "age of the machine" or the number of years of operation, and the machine model, expressed by Machine Code, which will be extracted from the asset sheet.

In the *Intervention Generator* subsection (Fig.4) maintenance interventions are initiated on the production lines, planned or in the event of a *breakdown*. Each intervention will include separate operations for one or more subassemblies of a machine. The component operations of the intervention are generated by the Cron script or manually, at the intervention of the maintenance director, following the technical damage inspections, depending on the *Planner*. *Operations* of *configurable* sets all maintenance operations and intervals for outgoing.

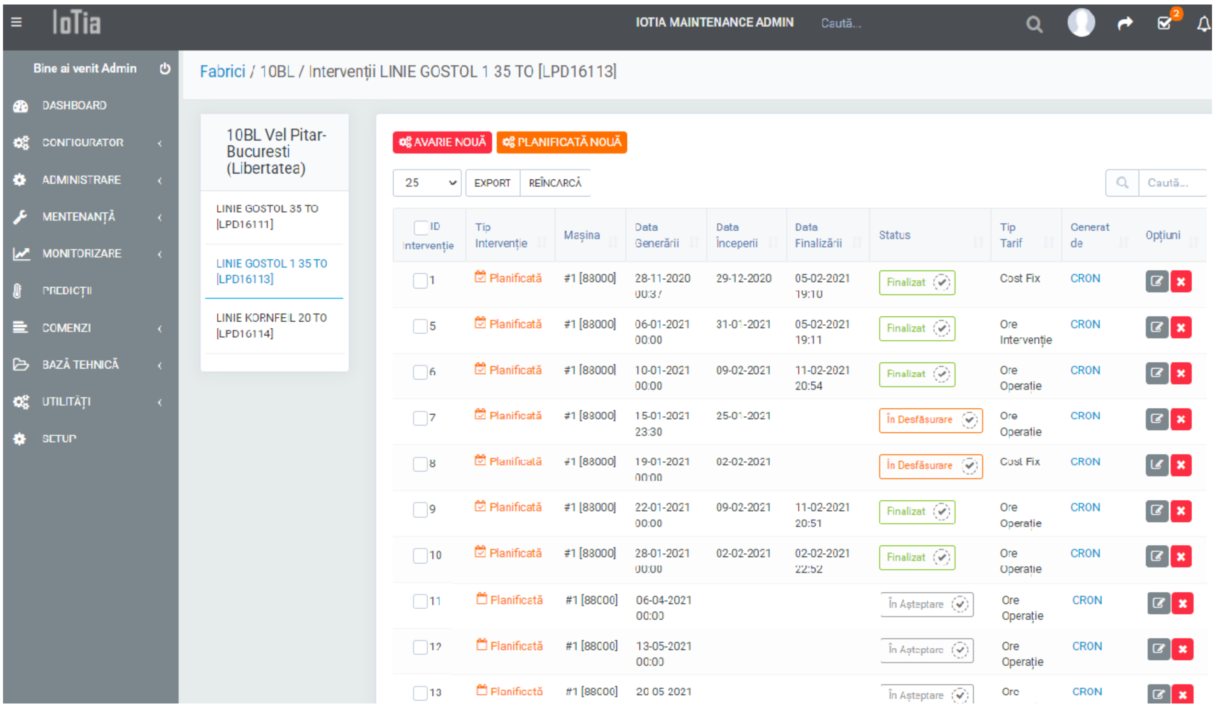

Fig. 4. Administration - Generating Interventions

- Visualization and creation of interventions on the production line

In case of emergency intervention, the “break-down” operations will be automatically selected, set in the *Configurator* for each type of defect discovered during the asset inspection and assigned to subassembly code.
Immediately after the automatic generation performed by Cron, the intervention has the status of *Pending*, and the system administrator or maintenance manager will schedule the desired operations, establishing the intervention team from the group of factory technicians or subcontractors, if they outsource maintenance services. It will also set the start date and a deadline for the completion of the intervention operations, and the intervention will enter the status of Not *started*. When the technician the execution of the first operation begins, the status of the intervention becomes: *In progress*.

For each operation generated in part by Cron, the number of hours will be estimated in the operation sheet so that, for an entire intervention, the system will be able to estimate the total number of hours required for all intervention operations. The estimated number of hours will also generate an estimated cost of the intervention (labour) depending on the type of tariff that is set per intervention: fixed cost intervention, tariff per hour of intervention or tariff per hour of operation. The number of hours performed for a given operation will be calculated based on the validation by the technician of the start and end times of an operation.

There are situations in which this time interval may also include waiting times, for various reasons (lack of components or materials needed for the intervention, lack of available staff), and the number of hours worked by the technician to be much smaller.
Real time can be captured by the timing system from the technician's interface (mobile / tablet or desktop application), by timing with the option *Start Timer* and *Stop Timer*. Thus, the method of charging for interventions and operations is very flexible through the system created, and maintenance costs can be designed flexibly in terms of labour costs, spare parts costs, consumables costs and other costs.

Each operation in the intervention sheet is marked by a Status, initially marked Not *Started*, and modified by the technician as: *In Progress, Completed, Delayed,* and a Priority Grade: *Low, Medium, High, Urgent*, set by the administrator.

Thus, it will be possible to report on the interventions and the state of operations as well as on the timing situations, both in the Maintenance Management module, in general, with broad access / edit / view rights, for the maintenance director, and restricted , in the Maintenance module, for the logged-in technicians, who will be able to view only their own operations and reports, having access to the interventions to which they have been assigned to intervene.

The IOTIA platform integrates scripts for sending email alerts, depending on the degree of priority, each time tasks are assigned to technicians or maintenance operations are undertaken by them, by communicating the status of the work.

This level of configuration of maintenance operations, by assigning a time interval to each intervention and operations specific to the maintenance team, prioritization, but also the feedback of technicians, provides a clearer picture of the status of assets and tasks and will generate a clearer view of operations in the Maintenance section, logged in users with different access rights.

Training sessions are initiated in the Administration module. In this section, the maintenance director / evaluator can create various evaluation questionnaires, and the feedback and employee training sheets can be viewed in the *Personal Training* section.

Preventive Maintenance Module
The section reflects all items set in Administration but filtered for view by permission and user type. This section contains:

• *Production lines*, by viewing assets on production lines with all subassemblies and information about component structures and defined libraries that are allowed to the user logged in to the section.

• *Interventions*, by viewing all the interventions in which the user is assigned to perform operations and operate them in the *Operation Sheets*. Each *Intervention* Form provides the assigned technician / supervisor / maintenance director / reliability engineer with a more complex view by:

- intervention summary (Fig. 5),
- list of operations and operation sheets
- time sheets,
- files attached by intervention members,
- discussions undertaken by the assigned team,
- the Gantt chart of the situation of the operations according to the assigned status / technicians, both on the production line and on the intervention, the
- tickets issued between the departments within the intervention operations and the situation of the expenses.

• *Operations*, by cumulative viewing of all operations within all interventions assigned to the connected technician / supervisor.

• *Alerts* (a list of alerts sent to the user)

• *Remote Assistance Support*, a subsection that integrates a video conferencing solution between support assistants / maintenance managers / supervisors / specialists and field technicians undertaking operations.

• Support through *Ticketing System*, with tickets open / in progress / with response / pending and closed, between the departments of Maintenance, Production, Procurement, IT, etc. The subsection allows communication between departments, each open ticket can contain a sequence of answers in chronological order and can be defined by several statuses.

• *Training*, by viewing the training sessions of the connected user or viewed on the entire team / factory, depending on permissions and role.

The elements of this module are only related to preventive and break-down maintenance, status-based maintenance and predictive maintenance are available in separate modules.

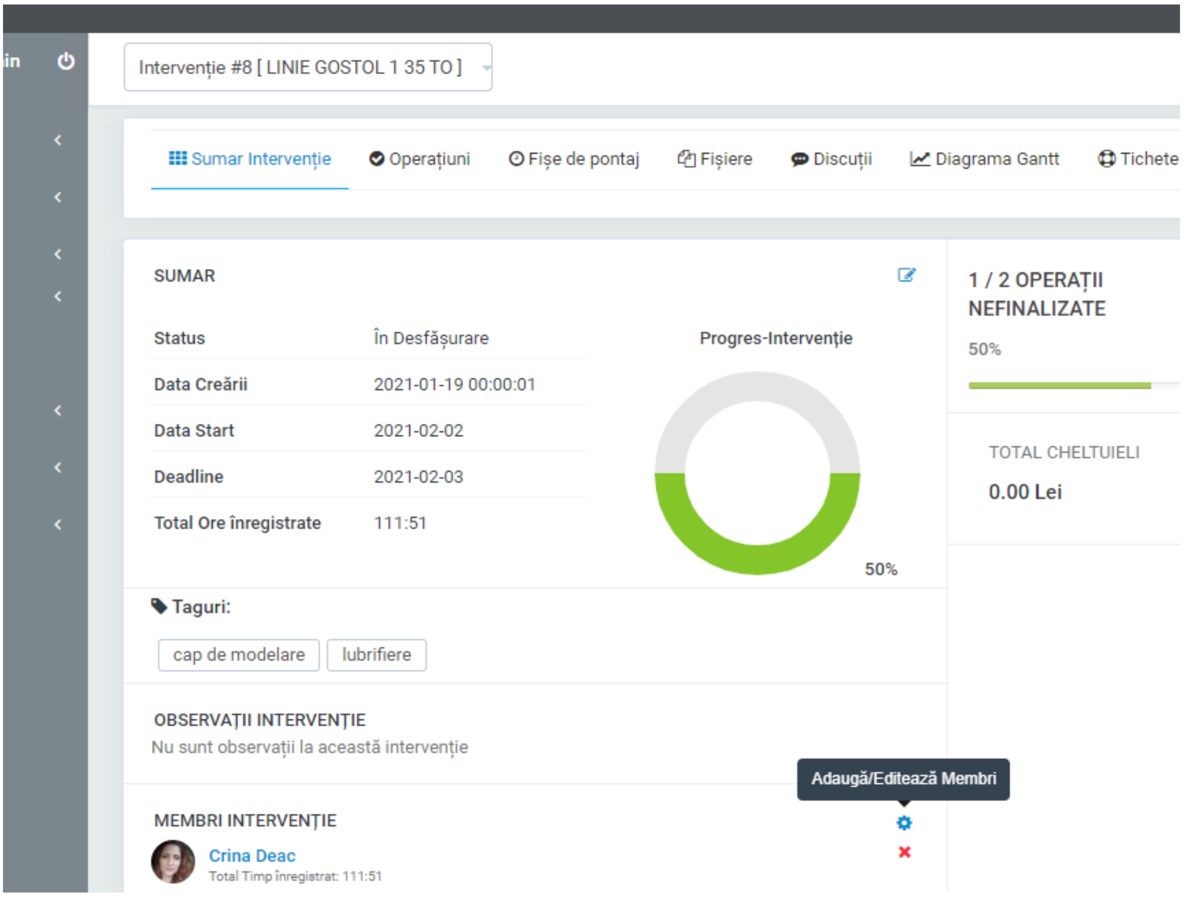

FIG. 5. Preventive Maintenance - Intervention Summary

The Maintenance module allows the user to view all information related to assets and interventions in two ways: either from within the platform (via the desktop application) or using a web-based AR application designed for tablet resolution and AR Glasses. The pilot project used the Vuzix M400 model for AR glasses.

The mobile application for maintenance with AR technology

The mobile application with AR technology (Deac et al, 2017) was initially launched as a research topic by the partner FESTO Didactic and consisted of creating a mobile web-based application for maintenance, using WEB and AR technologies open-source. It was later included in the IOTIA platform as a mobile web interface for field technicians, being connected to the desktop software application, but using QR codes instead of ArUco codes due to the limitation of the latter number.

The purpose of this application is to provide useful information for intervention teams and to facilitate the validation of inspection sheets and operations by:

- identifying machines and their components
- viewing the status of machine components and subassemblies
- accessing information on the history of processes previous maintenance (operations performed, observations of technicians, causes of failures, team members)
- information on stock of components, suppliers, prices
- easily complete the technical inspection sheet or the operation sheet within an intervention, by validating the actions taken, materials and consumables used
- sending orders to the purchasing department for spare parts / consumables
- accessing the technical library attached to subassemblies and components (with user manuals, technical manuals, technical drawings of the location of components)
- accessing the library media containing videos about maintenance operations
- requesting support, through remote assistance, through the augmented video conferencing application.
- opening a ticket to the company's departments, using the ticketing support.

Only *web technologies* have been used to create the application, so the application can run directly in the web browser, on any type of device or operating system. The type of client-server architecture ensures the centralization of information and the ease of maintaining libraries. The major challenge was to identify existing web technologies that could support this implementation.

We started with the basic principle of creating an augmented reality application, namely:

- downloading a video stream from a webcam
- analyzing AR identification markers
- generating 2D or 3D augmented content
- displaying the video stream and augmented content

The *WebRTC / getUserMedia* version was chosen for the video stream from the camera, which is a new HTML5 technology that does not require the installation of any additional drivers or plugins. The application interface was created using HTML5, CSS and JavaScript, and PHP and MySQL were used for the server application.

ArUco libraries have been chosen for AR markers, for which there is already implementation in JavaScript (ArUco libraries are based on OpenCV technology).

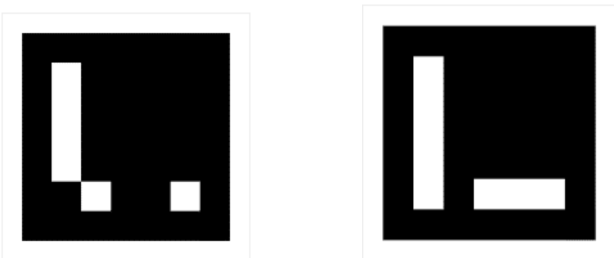

Fig. 6. Markers AR - ARUCO

WebGL 2.0. (Web graphics library) technology has been applied for augmented 2D or 3D content implemented in the *THREE.js* library, which allows the creation, import and display of 3D animations using JavaScript, without the need to import plugins.

The principle of operation is as follows:

- The technician scans with the tablet, phone or laptop, using the video camera, the markers placed on each machine (the representative marker of the machine or the markers of its subassemblies).

- When identifying the markers, the identification information of the machine / subassembly and their minimum description (image, name, code) are superimposed on the image provided by the camera.
-The technician approaches the target subassembly, and at that moment, following the identification based on the AR marker, all the useful information related to it is loaded in the web interface.

Fig. 7. Scan interface

The *Start (Home)* interface includes:
- *My Operations*: includes the list of all operations assigned to the technician
- *My Inspections*: includes the list of all inspections performed by the technician
- *Calendar:* contains tasks assigned to the technician on calendar data
- Button *Start Scan* selects the camera (front or back)

Interface *Scan* (SCAN) (Fig. 7) includes:
- the scanning screen where the AR application is implemented,
- link to the *Library section* where a technical library is selectively loaded, depending on the scanned AR marker,
- link to the *History Maintenance s*ection, which will load information related to the maintenance history of the scanned subassembly (subassembly code) with the operation sheets,
- link to *Procurement*, which will load the history of delivery requests and purchases of components of the scanned subassembly.

Clicking on each tab displayed in the *Library* screen opens diagrams, pdf instruction manuals or demonstration videos related to the maintenance operations performed on the scanned subassembly, defined in the *Configurator* section in the *Planner* library, once the operations are defined on subassembly code or in *Subassemblies* bookstore.

The screen *Maintenance History* contains the history of maintenance operations performed on the scanned subassembly, displayed by status: Not *Started / In Progress / Completed / Postponed* with detailed, one-click information about each operation.
The interface includes access to remote support through the RTC Web-based video conferencing application, which is included in the web application but can also be launched by

AR glasses, and includes ticketing support for opening a ticket to the purchasing, maintenance, production or IT.

To perform the assigned operations, the technician will call the symbol *Home*, exit the AR interface, and access *My Operations*. Here you will be able to view a list of statuses with all the operations on subassemblies that have been assigned to it in *the Intervention Generator*. By selecting an operation, the operation file will be opened (Fig. 8), and the code of the subassembly to which the operation refers will be updated in the memory and the data related to it will repopulate the tabs: *Bookstore*, *Maintenance History*, *Purchases*.

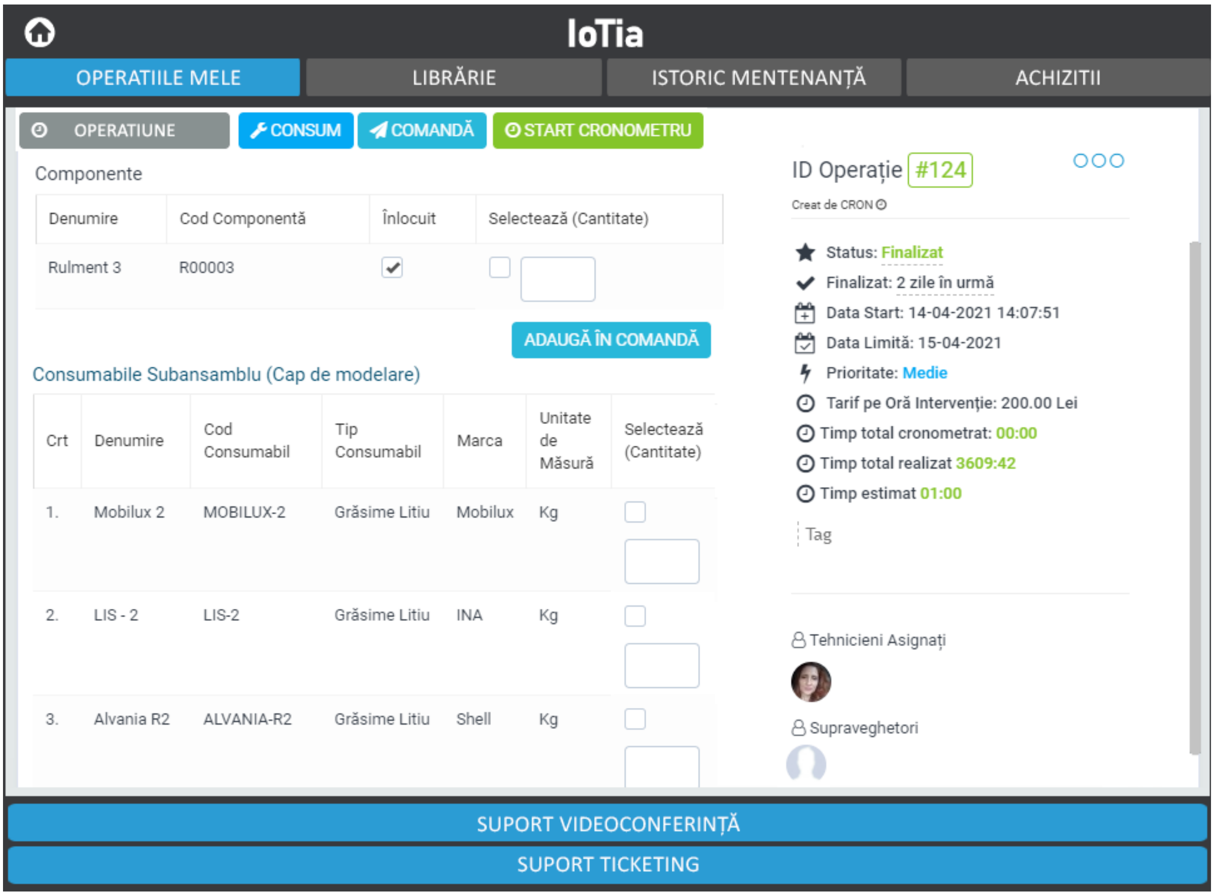

Fig.8. My Operations Interface / Operation Sheet The

technician will mark the start of the operation by selecting a new status with the option: *Mark in Progress*, and for timing will start the timer. He will check step by step the description of the operation and possibly the observations of the maintenance director. If you want technical information about the operation (subassembly diagrams, video about the operation), you will access the library for consultation or request remote support by accessing the video conferencing application.

The technician will fill *in the Consumption Form* with used consumables or replaced parts and will be able to issue delivery requests to the purchasing department (*Order*) for consumables or spare parts if he does not have the necessary for replacement.

At the end of the operation, if the operation is of the component replacement type, the technician marks the status *Completed*, and the component scheduled replacement operation is automatically marked in the database in the scheduled replacement table. The replacement operation will be recorded in the table *component failure*, recording the date, id and component code for the predictive model. In the Consumption Sheet, the consumables used, and the parts replaced in the event of failure are marked by ticking in the displayed list of components of the subassembly.

In the operation sheet, the technician will insert observation notes. If a new purchase is required, the operator will access the Delivery Request button, in which he will tick the code of the required components from the component list of the subassembly and the number of parts, thus generating a delivery request for purchases. All delivery requests will also be viewed on the platform in the Orders / Delivery Requests submenu.
The entire operation described above can also be performed from within the platform (desktop application), by selecting the Intervention Form in the Interventions section or by directly accessing the Operations subsection in List mode or Kan Ban mode.

Video conferencing support allows multiple users to connect simultaneously and includes the following features:

- High-Definition audio video communication,
- Screen sharing
- File transfer,
- text chat,
- video sharing on YouTube,
- SIP connection with other proprietary conferencing systems.

Video conferences can be recorded and uploaded to the Configurator's video library for later use by other technicians / operators.

Monitoring module

Monitoring of machines within the platform is done either based on data taken from PLCs (telemetry data, machine status, alerts, electrical or mechanical error codes), or based on sensors mounted on machines for analysis of various parameters condition: vibrations, temperatures, etc.

The pilot project used TA312-M12A sensors for measuring vibration and temperature, which were installed on a machine in the bakery industry for dough modeling - the Vipava 3000 500 long modelling machine. The

TA312-M12A sensors have the following characteristics: voltage power supply between 3-5V, sensitivity 25mV / g, a frequency response between 0.5-15000 Hz and temperature range between 3 and 121 degrees Celsius.

The machine-mounted sensors are connected to an ESP32 microcontroller via an ADC (digital to analogue converter) channel.

The ADC of the microcontroller has a SAR (Successive Approximation Register) architecture with a resolution of 12 bits. To be able to capture information in the frequency range typical of bearing vibrations (at least up to 5kHz) a double sampling frequency of at least 10kHz is required.

The TA312-M12A piezoelectric sensor used has a nominal sample specified by the manufacturer with a division of 25mV / g, a frequency response range up to 12kHz (± 10%) and a dynamic measuring range up to 50g. To cover the dynamic range, a resolution of the ADC is required which, depending on the nominal sample specified by the manufacturer, covers the maximum possible value. The ADC of the microcontroller specification meets this range: 2 ^ (12-1) / 25 = 81.92g> 50g (The minimum accepted resolution is 11bit, in which case a maximum dynamic range of 40.96g is reached, losing only 9 samples in the specified dynamic range).

Given the effective analogue bandwidth of the integrated 6kHz ADC for which it maintains a linear response based on the sampling frequency, the aim was:

- Implement a *look-up table* to normalize the frequency response, with an initial calibration routine that uses the microcontroller's DAC to implement a *feedback loop.* It is used to map all points in the frequency ranges outside the specification.
- Implementation of a wrapper polynomial correction

To improve the SNR (Signal-to-Noise Ratio), the ratio between the desired signal and the noise, to reduce noise was aimed at:

- Implementation of an analogue RC filter for the nominal sampling frequency of 10kHz
- Testing a solution oversampling
- Implementation of a low-pass digital filter to eliminate spectral noise due to reflections, with the possibility of use in conjunction with anti-aliasing oversampling
- Use of a nominal response instrumentation amplifier in the frequency range up to 10kHz for SNR improvement (desired signal-to-noise ratio)

Communication with the server is done wirelessly over IP, at the frame level through the MQTT protocol, and at the level of transmitted information the specific format *Influx DB* from the *Line protocol is used*. There is also the possibility of creating a network *mesh* between microcontrollers with a single MQTT gateway (Fig. 9) in the margin of limiting the rate of data that can be processed by each node.

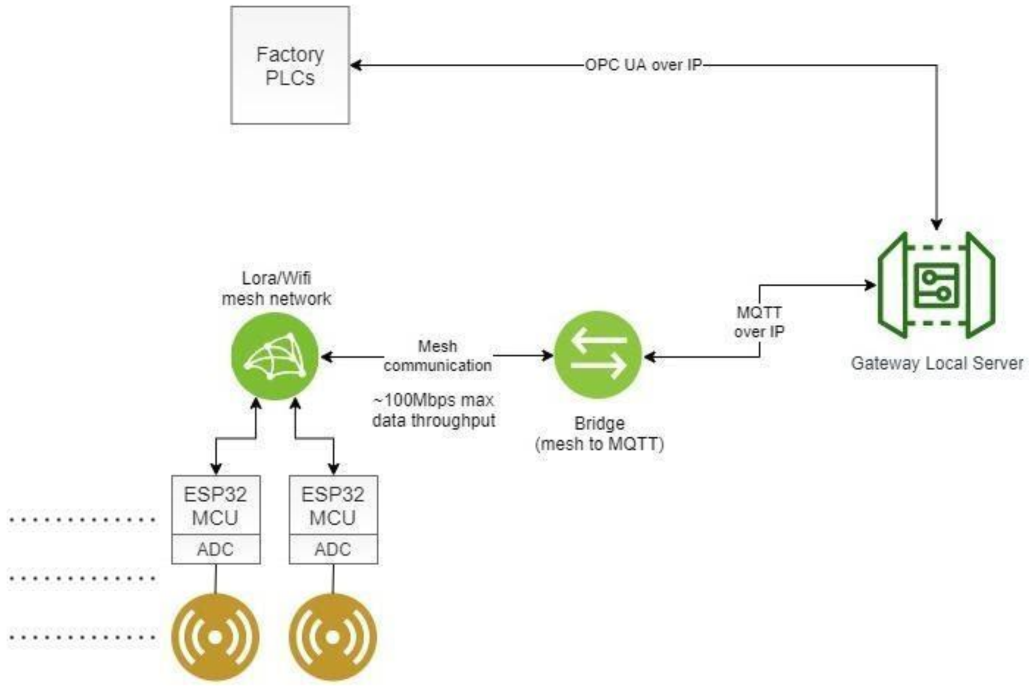

FIG. 9. Factory data transport with Mesh network

For the current monitoring solution with a high data rate, a direct connection was used with *the Wi-Fi router* that takes the data from the microcontroller and transmits it to the server. At the higher level of the communication protocol (MQTT) two implementations were tried: MQTT over TCP frame and MQTT over WebSocket's. The first implementation is used for bulk data samples, and the latter has higher performance for serial data.

Therefore, the implementation within the microcontroller can have 2 variants of transmitting samples in packages: in bulk (samples accumulated over a predefined period of time, the size of the block) and serial, in real time (each package contains a single sample) . The latter has poor performance in the event of a large data flow.

Telemetry data from machines are collected from PLCs via OPC UA, IP or other accepted protocols.is used to view the data *Grafana Dashboards*.

The monitoring section displays the production lines with the monitored machines and the monitored asset sheets. An overview of the machine is displayed by selecting a monitored machine. It contains data about the machine, such as: name, manufacturer code, ID, machine number, year of production, current status of the machine, weight (Kg), rated voltage (V), frequency (Hz), voltage connection (kW), date of installation, availability in hours, time unavailable in hours, total number of alerts, error logs, total number of critical issues by counting all break-down operations in the history of interventions, last service with last intervention (date) next intervention (time) index status of the machine.

The screen also displays a table with all the operations performed on the machine subassemblies in the intervention history, including break-down operations and planned component replacement.

Depending on the parameters monitored on each machine, an i-frame view, called Panel, generated in *Grafana*, an open-source platform for querying, viewing and alerting time series parameters and logs, displays graphs of status parameters, moving average at 6 or 12 hours, depending on the configured interval, using the ability to mediate graphs (diagrams). It can still use a *counter* time, using the feature *Legend*, to display the average, minimum, maximum, using the mediation *line* (*Plot average line*) or other aggregation data, such as Mediate and Median.

The main telemetry parameters that characterize the evolution of the car will be considered in the characteristics that will form the final set of input that will drive the predictive model.

Fig. 10. Visualization of telemetry data with Grafana

Predictive Maintenance Module

This section contains predictive models developed in the platform based on data accumulated from sensors and PLCs and transmitted to the central server as well as data stored in CMMS from maintenance teams or other imported databases. Predictive models are written in Jupyter Notebook, an open-source web application called the platform, which shares documents that include code, views, and additional descriptions.

The platform uses notebooks to analyse exploratory data, stored in a library organized in folders and files, running a notebook server. Connections are made to the central server that hosts the unified database and to the local servers in the factory that store telemetry data received from sensors or PLCs. Data is read and uploaded to the *Pandas* library. Other libraries are used for graphical data display, visualization, scientific calculations (*Matplotlib, Seaborn, SciPy library)* and modelling.

Models are built using high-level APIs such as Keras for implementations *Deep Learning*, scikit-learn for ML implementations in Python, pre-processing, model selection, classification, regression, clustering, and scaling.

Predicting component failure in a time window using multi-classification algorithms. Study the case.

Detecting the fault, diagnosing faults, and knowing the likelihood of a particular machine failing due to components in the next period are crucial topics in Industry 4.0 analysis. In this way, we can prevent major failures in advance and act before the event occurs through timely maintenance, with limited resources, in a more cost-effective way, improving the quality and supply chain.

In this implementation we want to know the probability that a certain car will fail due to the failure of a critical component in the next time frame.

Planned replacement operations are known to run at regular intervals when some components may still work. Knowing the probability of component failure in a matter of days, you can control the cost of maintenance by performing preventive maintenance at the right time, close to failure and extending the operation of the component as much as possible but preventing major damage.

The planned replacement operations within the preventive maintenance are performed *"just in time"*, and the corrective maintenance interventions are reduced, preventing critical breakdowns and long downtime. Moreover, knowing the time of failure, the inventory of existing spare parts in stock or the requests for delivery to the purchasing department can be resolved in a timely manner, before the intervention.

Because there are several critical components in a machine, determining the probability of component failure over time is reduced to a multi-classification problem using time series and machine learning algorithms.

A multi-classification model will be based on data collected from several similar production lines (from one or more factories), and each sample of data marked by a *timestamp* will be labelled with the status of critical components: The "critical component" that failed, if the sample describes a failure state, or "normal", if the sample is a normal state in which none of the critical components failed.

For each type of machine, several component codes can be identified - critical components, which can be considered *target labels* in the predictive model.

Telemetry data captured in the monitoring process from PLCs, such as voltage, pressure, rotation, vibration, will be used in the model feature set, along with other features, including a set of aggregate measurements, calculated on a *lagging window* 6 or 12 hour (rolling average, standard deviation, minimum or maximum value in that range).

Other features will be added to this feature set, such as error logs received from machines while they are still operational. Only those errors (error ids) that are crucial in predicting a future failure will be chosen from the error glossary, and because they cannot be mediated, being categorical values, the total number of errors of each type will be considered in the window. set delay, for each machine.

Also, the number of days since the last replacement for each component code, from each machine ID, will be calculated using the maintenance history in the Maintenance section, looking for break-down replacement operations and the history of planned component replacements, knowing that the time interval since the last replacement represents the number of operating hours of the component, which is highly correlated with its degradation level.

The target variable, the component that failed, will be extracted from the intervention history *break-down*, from the component replacement operations to the failure, from where the component codes, the machine ID and the timestamp that marks the moment of failure for each machine will be extracted. Another useful feature that could complete the input data set provided to the model is the age of the machine taken from the technical data sheet of the machine (Year of production / Date of installation), knowing that the number of hours of operation of the machine is correlated with the degree of Wear.

Based on these characteristics extracted from historical data, predictive models learn degradation patterns and will predict future outcomes with some probability. The validation of the proposed methods was done by comparing the actual failures in the test set with those predicted by the model driven on the training set. The confusion matrix was calculated showing the predicted faults in the column and the actual failures of the components in rows.
Several classification models will be considered in the prediction: models based on decision trees or models based on neural networks. The model with the best results in the evaluation metrics will be chosen in the end to make the prediction on the data set in real time, in the process of monitoring the machines.

Different data sets, taken from the Microsoft Azure cloud, were used to illustrate the solution, with records of the operational history from a set of 500 machines of 4 types, over a period of operation of approximately one year.

Analyzing the evaluation metrics, the XGBoost classifier obtained the highest score F1 = 0.83, being the best classifier for the study data.

XGBoost (*Extreme Gradient Boosting)* uses the principle of gradient boosting, but with differences in modeling details. XGBoost uses a more complex form of adjustment of the model for over-fitting control and achieves better performance.
The objective function to be optimized is the Loss function regularized by the term Ω according to the formula:

$$Obj^{(t)}(\Theta) = L(\Theta) + \Omega(\Theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^{t} \Omega(f_i) \quad (1)$$

The complexity of the decision tree is given by the adjustment term (f) and is defined for XGBoost as follows:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} \omega_j{}^2 \quad (2)$$

where ω is the vector of the tree leaf scores, T is the number of leaves.

XGBoost trees can have a variety of end nodes.

The XGBoost algorithm better supports processing *multi-core*, which reduces drive time.

Table 1. Confusion matrix for XGBoost classifier

Predicted failure	component1	component2	component3	component4	none
failure					
component1	1495	34	4	10	386
component2	20	2014	16	9	475
component3	7	10	608	4	141
component4	21	30	1	983	189
none	196	255	144	127	178 366

Accuracy = 0.988795
Accuracy = 0.851703
Recall = 0.810682
F1 = 0.830687

From the studies performed and the results obtained, the Random Forest algorithm can also be used, for which a score of F1 = 0.76 is obtained, being faster than the Gradient Boosting classifier.

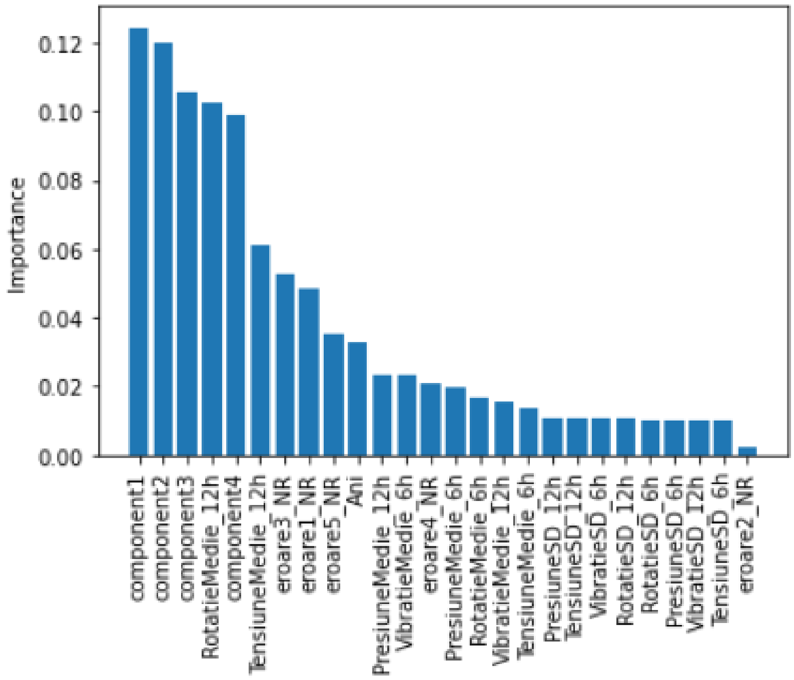

Fig. 11. Importance of *Features* for the XGBoost Model

The neural network did not perform as well as the decision trees, not having enough examples of failures to learn the degradation behavior of the components, the score being only F1 = 0.55.

Diagnosing bearing defects by vibration analysis, using the Deep Learning approach. Case Study.

The use of labeled data to learn expressive representations of normality / abnormality is crucial for accurate detection of anomalies. The models proposed in this case study are supervised learning models, with good classification accuracy, which learn directly from the raw time series data provided by accelerometers. The only condition is that the samples are labeled correctly.

The case study uses a set of deep learning algorithms to diagnose bearing defects, which will detect the degradation pattern of the vibrations measured by the sensors and will finally classify the types of defects labeled in the drive set.

The method presented is for point anomalies - carrying defects in the early stages and cannot be used for group anomalies. It also focuses on detecting anomalies from single data sources, not multiple heterogeneous data sources.

The study tests several neural networks on the open-source data set CWRU (Case Western Reserve University), using multivariate time series.

We tested: deep neural networks (DNN) composed of fully connected layers, convolutional neural networks (CNN) defined by various groups of convolutional layers and *pooling* layers (restriction), recurrent neural networks (RNN), Long short-term memory networks (LSTM) built on recurring layers and hybrid networks - combinations of RNN and LSTM with convolutions. All models are evaluated and compared.

Finally, the study obtains the best results on the RNN networks enhanced with convolutions and then by averaging the result with the second best obtained model (CNN), it obtains an accuracy in the test data set of 98.50 %.

The method captures the vibration dynamics of both bearings in two simultaneous time series and the dependence between the two series, with more complex labeling, for more reliable predictions.

To validate this deep learning neural network-based diagnostic methodology, we used data provided by *Case Western Reserve University* (CWRU - https://csegroups.case.edu /bearingdatacenter/pages/12k-drive-end-bearing-fault- data), coming from vibration sensors mounted on a motor in an experimental stand (Fig. 12).

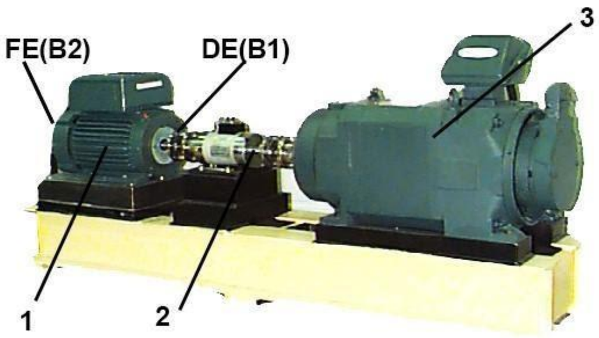

Fig. 12. CWRU Experimental Stand

The stand used to purchase the bearing data set consists of a 2 hp electric motor (1), a torque transducer (2), a dynamometer (3) and an electronic control module. Test bearings support the motor shaft. Punctual failures were introduced, using a machine with electric discharges on the test bearing (B1), with defect diameters of 0.007” and 0.021” at two different rotational speeds: 1797 rpm, engine load (hp) 0 and 1772 rpm, motor load (CP) 1.

Defects in the outer ring are stationary defects because the outer ring is fixed.
As a result, the placement of the fault relative to the bearing area of the bearing has direct implications on the vibrational response of the system. DE (B1) drive end bearing has defects in the outer ring, located in the loading area in the 6 o'clock position, at 3 o'clock (orthogonal to the loading area) and at 12 o'clock (opposite the loading area) (Fig. 13).

0.007” and 0.021” point defects were also placed in the B1 bearing, inner ring and rollers.

FIG. 13. Placement of faults relative to the loading area (6), opposite the loading area (12) and orthogonal to it (3).

All data sets were collected using accelerometers attached to the magnetically based motor housing and placed in the 12 o'clock position, both at the drive end DE (B1) at bearing B1 and at the fan end FE (B2) at bearing B2, for each rotational speed. Data were collected at a *sampling rate* of 12,000 samples/ second for both bearings.

Fan end bearing B2 is considered normal, with no defects in the study, but the proposed model may well model defects in B2, which involves adding more labels, indicative of fan end defects (B2), to the samples provided.

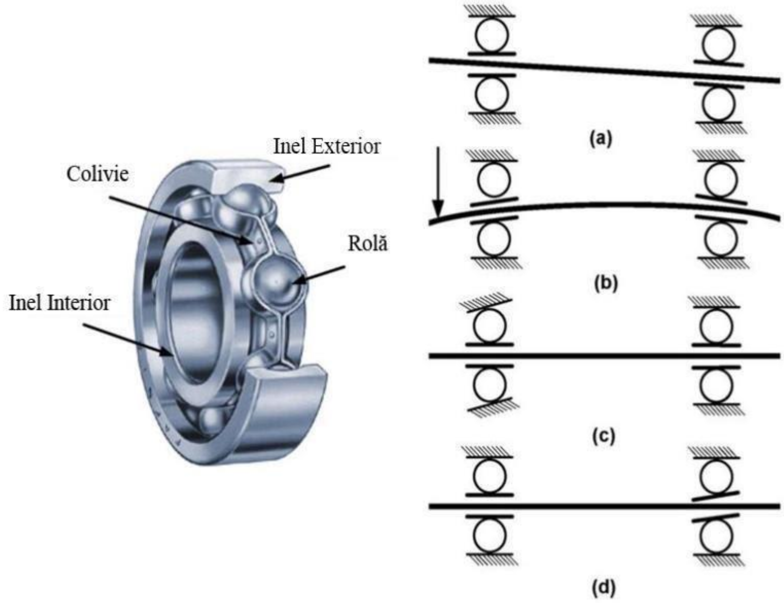

Fig. 14. Four misalignment scenarios that can cause bearing failures:
a.Non-aligned shaft, b. curved shaft, c. Outer ring inclined, d. Defective in inner ring

	B1	B2
0	-0.277602	0.040885
1	-0.044345	0.069855
2	0.117603	0.337767
3	-0.145055	0.251682
4	-0.111430	0.108891
...	...	...

Fig. 15. Defective data set [mV / g] in inner ring in bearing (B1), defective diameter 0.007” and normal fan end bearing (B2), Load 1

The data sets comprising the acceleration bearings at the drive end (B1) and the fan end (B2) are considered separately for two fault diameters, 0.007” and 0.021”, in two different load cases - 0 hp engine load and 1 hp engine load, for 5 types of faults. The sets contain defects in the inner stroke, defects in the rollers and 3 types of defects in the outer stroke: centered, orthogonal, respectively opposite to the loading area.
Also, two sets of data for normal behavior were considered for each task, resulting in 11 classes / labels in the end (Table 2).
This means that the study will be performed for each task separately (task 0 and task 1) using 11 data sets each (11 time series), a total of 22 data sets provided by the CWRU:
- two data sets for normal bearing behavior, without defect, for each load, five data sets for the five types of defects, with a diameter of 0.007 ”, at load 0, five data sets for the same five types of defects, but having a diameter of 0.021” , at load 0, five data sets for the five types of faults, with a diameter of 0.007 ”, at load 1 and five data sets for the five types of faults with a diameter of 0.021” and load 1. Each data set will be labeled, in each case of loading, according to Table 2.

Table 2. Sets labeled on Load 0

Data type (B1)	Data type (B2)	Nr. Samples (B1, B2)	Label
Normal	Normal	243938	Label 0

Defect in inner stroke 0.007”	Normal	121265	Label 1
Defect in outer stroke centered 0.007”	Normal	121991	Label 2
Defect in outer stroke orthogonal 0.007”	Normal	122281	Label 3
Defect in outer stroke opposite 0,007”	Normal	122281	Label 4
Defect in rollers 0.007”	Normal	122571	Label 5
Defect in inner stroke 0.021”	Normal	122136	Label 6
Defect in outer stroke centered 0.021”	Normal	122426	Label 7
Defect in outer stroke orthogonal 0.021”	Normal	121701	Label 8
Defect in outer stroke opposite 0.021”	Normal	121846	Label 9
Defect in rollers 0.021”	Normal	121991	Label 10

The time series will be further segmented into samples to be supplied to the proposed models. We need to know the length of the time series. The total number of data recorded by accelerometers in the original data sets can be seen in Table 5.23. A total of 1464427 samples, included in all 11 files, will be the buffer size of the study data set for task 0.

At task 0, the engine speed is 1797 rpm (RPM) and because the sampling rate is of 12000 / s, means that 12000-point data are recorded in one second and we have a speed of 30 rot / s, with 400 data points contained in a period of rotation. At load 1, the rotational speed is 1772 rot / min (RPM), at the same sampling rate, this means 29.53 rot / s, so for both cases the rotational speed will be approximately 30 rpm.

Multivariate time series were formed, considering parallel sequences of data provided by accelerometers mounted on B1 and B2 (Fig. 16):

$$\begin{array}{cc} \vdots & \\ \mathrm{B1} & \mathrm{B2} \\ \vdots & \end{array}$$

Fig.16. Composition of the multivariate time series (series of two variables, one variable in B1 and one variable in B2)

Each of the 11 multivariate time series will be divided into three parts (70%, 15%, 15%) and then the parts of the series will be concatenated. The first 70% of each time series will be concatenated to obtain the training data set, the next 15% will be concatenated to form the validation data set, and the last 15% of all 11-time series will be concatenated. be concatenated to compose the test set.

Validation set will help prevent over-involvement *(overfitting)* set training throughout the ages drive and define the best neural network.

The test data set will be used to calculate the accuracy of the model classification.

After the preparation of the three data sets, the data were segmented by dividing them into sequential batches of quarter rotation, consisting of windows of 100 data points that were served to the model as two-dimensional feature vectors, in tensors of length 100, by using a segmentation function.

At the length tensor 100, the function will map the label, which is another length tensor 11, marking the classes (one class for normal and ten classes for defects), obtained by coding *one hot* of the labels. The function will mix the samples thus formed and will form batches of 20 tensors of this type (Fig. 17):

(<tf.Tensor: shape = (20,100,2)>, <tf.Tensor: shape = (20,11) >)

$$\begin{array}{ll} [[[B_1, B_2], \ldots {\scriptstyle 100} \ldots, [B_1, B_2]], & [[L_0, L_1, \ldots, L_{10}], \\ \quad\quad \vdots\ {\scriptstyle 20} & \quad\quad \vdots\ {\scriptstyle 20} \\ \ [[B_1, B_2], \ldots {\scriptstyle 100} \ldots, [B_1, B_2]]], & [L_0, L_1, \ldots, L_{10}]] \end{array}$$

Fig. 17. A set of 20 tensors - Segmentation of the two-dimensional data set

in two - dimensional tensors, length 100

This segmentation of the two-dimensional data set will feed the CNN, RNN, RNN-CNN, LSTM-CNN, LSTM models proposed in the study, all accepting a multidimensional input, batches of 100 two-dimensional tensors. These models will fit a total of 100 + 100 features on a single label.

A DNN model will also be trained, in this case reducing the two-dimensionality by concatenating the two segments from the two sensors, due to the one-dimensional input accepted by the DNN network, obtaining a one-dimensional tensor of 200 points.

$[L_0, L_1, \dots, L_{10}]$ is the vector encoding *one hot* of the label,
$[L_0, L_1, \dots, L_{10}] = [\mathbf{1}, 0,0,0,0,0,0,0,0,0,0,0,0,0]$ means the representation of the normal state, class 0, without defects, having 1 in the first position of the vector, at index 0.

$[L_0, L_1, \dots, L_{10}] = [0,0,0,0,0,\mathbf{1},0,0,0,0,0,0,0]$ represents the roll defect of 0,007 ”,
which is class 5, having 1 on the 6th position, at index 5 of the vector.

When local defects occur in the outer / inner ring and in the tread (cracks, holes), the interaction between the tread and the tread results in discontinuous, time-varying, and uneven contact forces, which generate a specific signature in the signal. vibrational.

It was possible to observe the signature of the vibrations specific to each type of defect and size and the way in which the defect of one bearing induces vibrations in the second bearing without defect, through the shaft.

For each load we tried to define and train several models.

Some models required a change in structure, more layers, or more training times when switching from Load 0 to Load 1 to get comparable results, but the best models worked. the same for both cases.

The objective function (called the *Cost* or *Loss function*) used for all the proposed models was *Categorical Cross Entropy*, which calculates the loss of a sample by calculating the following sum:

$$C(y_i, \hat{y}_i) = \textit{Categorical cross-entropy} = -\sum_{i=1}^{dimensiune\ ieşirii = 11} y_i \cdot \log \hat{y}_i \tag{3}$$

where $\hat{y}_i$ is the scalar value at position i (of node i) predicted at the model output, y_i is the corresponding (real) target value, and the output size is the number of scalar values at the model output (11 values for 11 classes).

The aim is to minimize the objective C function by using the gradient reduction method. The shift to the minimum is determined by the gradient of the objective function:

$$\nabla C = \left(\frac{\partial C}{\partial \omega 1}, \frac{\partial C}{\partial \omega 2} \dots, \frac{\partial C}{\partial \omega n}, \frac{\partial C}{\partial b}\right) \tag{4}$$

The iterative method of optimizing the objective function was performed with the *Adam* optimizer instead of the *Stochastic Gradient Descend* (SGD) optimizer, which got better results. The Optimizer *Adam* calculates individual learning rates, adaptive, for each weight, which it adapts separately to each epoch (6), learning is improved, while SGD maintains a learning rate - a step of single drop (alpha) for all weight updates and does not change during training.

The activation function chosen in the last output layer is *Softmax* function in all models, because in this case we wanted to predict only one defect in each sample, the data sets in the study containing only one type of defect.

With the function *Softmax*, the probability $P(c_j | x_i)$ of a class c_j (where $c_j \in \{0,1,\dots, 10\}$), given the sample x_i (which is a tensor), is dependent on the probabilities of the other classes, as in formula (5):

$$P(c_j|x_i) = \frac{e^{z_j}}{\sum_{k=0}^{10} e^{z_k}} \quad (5)$$

where $z_j = \sum \omega_{ij} a_i + b_j$ is the sum of the linear combination of activations in the previous layer and the bias of the unit j in the final layer.

In subsequent tests, changing the activation function to the *sigmoid* function, in the last layer, for the prediction of several simultaneous defects, gave equally good predictions.

For the study in Python, the following were used: TensorFlow 2.0 and NumPy libraries for scientific calculation, Matplotlib Pyplot for graphical data display, Pandas for reading csv files and data set composition in DataFrame, sklearn for metrics and Keras for defining learning models deep.

All models were first trained in 100 epochs to select the best learning rate - the best *learning rate* for finding the size of the initial gradient step for the Adam optimizer.

A *callback* function was used to calculate the best *learning rate, denoted by lr_schedule,* which best reduces the loss, its value being adjusted at each time according to formula (7). *lr_schedule* is a configurable hyperparameter, it controls the speed with which the model learns, it is the step of moving to the minimum of the objective function at each iteration, by updating the weights according to the formulas (5.9):

$$\omega = \omega + lr\_schedule \frac{\partial C}{\partial \omega} \quad (6)$$
$$b = b + lr\_schedule \frac{\partial C}{\partial b}$$

In order to choose the best *learning rate* , a graphical representation of it was made according to the value of the loss obtained at each epoch and the value *lr_schedule* was chosen that minimizes the loss the most (Loss), according to formula (7).

lr_schedule = tf.keras.callbacks.LearningRateScheduler (
*lambda epoch: 1e-7 * 10**(epochs / 20))* (7)

The 6 models built for both load cases will have as evaluation metrics: categorical accuracy (ratio between correctly predicted number of samples and total number of samples), loss and F1 score.

RNNCNN Model - Recurrent Neural Network with Convolutions

An RNNCNN model was constructed by alternating two sequences consisting of two recurrent layers and two 1D convolutions. The first two convolutions have a *max-pooling layer* that restricts 3 times the number of parameters and the next two convolutions in the second RNN sequence are followed by a global constraint layer (*global average* pooling), followed by a last dense layer with 11 units neuronal, for the prediction of the 11 classes.

Twelve 5-core filters were used for each convolution, and the number of memory cells was changed from 100 to 50 because 100 cells over-trained the model. The results obtained were remarkable.

After selecting the best *learning rate* starting, from *1e-5* to task 1 and 7e-5 to task 0, the model was trained over a period of 1000 epochs in both cases.

A total accuracy of 99.75% was obtained on the training set, respectively 98.40% on the validation set, on task 0 and a total accuracy of 100% on the training set and 96.62% on the validation set, at load 1.

When evaluating on the test set, a very good accuracy of 98,54% was achieved at load 0 and 97,33% at load 1, respectively, this result surpassing all other models tested.

The recognition of the 0.007” diameter defect in the roll - Label 5 - has been improved, obtaining a Recall of 87% at load 1 and 99% at load 0 (Table 3).

Table 3. Classification report Model RNNCNN, on test set, Load 0 and Load 1

Label	Precision load 0	Recall load 0	F1 load 0	Precision load 1	Recall Load 1	F1 Load 1
Label 0	1.00	1.00	1.00	1.00	1.00	1.00
Label 1	1.00	0.99	1.00	1.00	1.00	1.00
Label 2	0.99	0.97	0.98	0.99	0.96	0.97
Label 3	1.00	1.00	1.00	1.00	0.98	0.99
Label 4	0.99	0.96	0.98	0.95	0.96	0.96
Label 5	1.00	0.99	0.99	0.93	0.87	0.90
Label 6	0.99	0.97	0.98	0.98	0.99	0.99
Label 7	0.93	0.98	0.95	0.95	0.98	0.96
Label 8	0.97	1.00	0.99	0.99	0.99	0.99
Label 9	1.00	0.96	0.98	0.99	0.98	0.99
Label 10	0.95	1.00	0.97	0.86	0.91	0.88
Accuracy		Load 0	0.985		Load 1	0.973
F1		Load 0	0.99		Load 1	0.98

The accuracy also for Label 5 is 93% at Load 1, which means that 93% of the total predictions for Label 5 are true, the confusion matrix also illustrating this result in Fig. 18 (158 samples correctly detected and 11 assigned to other classes).

Predicted Actual	0	1	2	3	4	5	6	7	8	9	10
0	725	0	0	0	0	0	0	0	0	0	0
1	0	182	0	0	0	0	0	0	0	0	0
2	0	0	175	0	3	1	0	2	0	0	2
3	0	0	0	179	0	0	3	0	0	0	0
4	0	0	0	0	175	0	0	5	0	0	2
5	0	0	0	0	0	158	0	0	0	1	23
6	0	0	0	0	0	0	181	0	0	0	1
7	0	0	2	0	0	0	0	178	1	1	0
8	0	0	0	0	0	0	0	2	180	0	0
9	0	0	0	0	0	2	0	1	0	180	0
10	0	0	0	0	6	9	1	0	0	0	166

Fig. 18. Confusion matrix on the test set in the RNNCNN model at Load 1

The highest performances were obtained by the RNNCNN and CNN models, according to the summary of the scores in Table 3.

A function in Python is written, to make a weighted average *elementwise* between model predictions for each class, trying to further improve the score. The function is based on an array containing the best models and the test set.

Table 3. Model Summary

Model	Total accuracy On test set Load 0	F1 On test set Load 0	Total accuracy O ntest set Load 1	F1 On test set Load 1
1. CNN	94%	0.96	92%	0.94
2 DNN	89%	0.93	79%	0.82
3. RNN	90%	0.94	89%	0.91
4. RNNCNN	98.50%	0.99	97.35%	0.98

5. LSTM	79%	0.85	66%	0.68
6. LSTMCNN	86.61%	0.91	82.29%	0.85

At load 0, the mediation model obtained a maximum accuracy of 97.94% and an F1 score = 0.987 by mediating the CNN and RNNCNN models, without downgrading the RNNCNN, and at load 1, the mediated model improved the prediction by approximately 1%, eventually reaching an accuracy of 98.19% (Table 4).

Table 4. Summary of the Best Models

Best Models	**Accuracy Test Set Load 0**	**F1 Test Set Load0**	**Accuracy Test Set Load 1**	**F1 Test Set Load1**
RNNCNN	**98.50%**	0.990	97.35%	0.980
Mediation Model (CNN, RNNCNN)	97.94%	0.987	**98.19%**	0.987

Original Contributions

The original contributions are structured according to the objectives of the thesis, set out in the first chapter of this paper. The main contribution of the research conducted in the thesis is the design and development of a collaborative web-based platform for maintenance management, a CMMS tool to manage several types of maintenance: reactive, predictive, maintenance based on status monitoring and predictive maintenance.

This research successfully designed, developed, and implemented the IOTIA web-based platform and conducted a series of studies to identify methods based on machine learning to diagnose and predict the status of monitored equipment and detect an anomaly. However, the research work during the years of the doctoral study consisted not only in the realization of the platform but also in the completion of the current state in the field, the analysis of mathematical concepts underlying artificial intelligence algorithms, the analysis of traditional methods of statistical data analysis to identify the anomaly, diagnostics and forecasting. The platform also integrates modern methods that include artificial intelligence and IoT technology for diagnostics, forecasting with ML and DL techniques and forecasting with various methods of prediction failure and RUL estimation of an equipment/component.

Thus, the main personal contributions of the thesis based on the secondary theoretical and conceptual analysis embedded in implementation platform are:

1. Conceptual analyzes were conducted on smart factory solutions in the context of Industry 4.0, the cyber-physical system and the CPS system architecture that would allow the creation of a CMMS interface capable of connecting and communicating at all levels. The aim of the research was to choose an architecture for the CPS system based on which the maintenance platform developed in the thesis can be implemented.

2. The generation-specific solutions of maintenance over time were analyzed and the maintenance techniques within them were presented. The evolution of the maintenance concept was studied, presenting the main types of maintenance that can be applied in Industry 4.0 as well as their limitations. The purpose of this research was to identify the maintenance concepts applied in Industry 4.0 and which can be integrated into the IOTIA platform as well as the necessary changes to remove some of their limitations.

3. Current methods based on machine learning were analyzed using ML or DL artificial intelligence algorithms, as well as statistical methods for detecting the anomaly in the monitored data from the sensor measurements. The aim of the research was to select methods

for detecting the anomaly according to the particularities that may be encountered in the industry, for use in the platform: some methods require more resources for calculation and are more sensitive to noise, other methods depend on the size of the training set and give results when we have a large number of samples, greater than the number of features, while other methods scale well for large values of the number of features. Some methods require data labeling (abnormal data and normal data) and can be adopted when we have a large number of anomalies in the training set, other methods based on unsupervised learning, such as Autoencoder networks, do not use data labeled to make a classification, but use the model's drive on wear-free equipment (we have no anomalies), so that we can later detect deviations when the equipment begins to degrade.

4. The performance and limitations of modern fault diagnosis techniques have been investigated with algorithms that in addition to detecting the anomaly also detect the cause, using the data-based approach and the model-based approach. In the data-based approach, with a wider applicability in the industrial sphere, classification diagnostic techniques have been studied based on traditional machine learning algorithms and deep learning algorithms. The aim of this research was to select modern DL algorithms to a large extent for diagnosing defects within the IOTIA platform, as they have a higher representation power compared to traditional ML algorithms and the ability to automatically extract and select representative features from the monitored data set. In this regard, in Chapter 5 a case study was performed for the diagnosis of incipient defects in bearings with deep learning algorithms, by vibration analysis. The raw data transmitted by the sensors (accelerations) were used, and the characteristics were extracted directly from the neural networks.

5. A series of research were performed for the selection of RUL prediction techniques using ML and DL suitable for the development of the proposed platform. To predict the remaining useful time of the equipment, several forecasting techniques were presented in paragraph 3.4.2 of Chapter 3 which is based on three approaches: the model-based approach, the experience-based approach, and the data-based approach. The purpose of researching these techniques was to select them to be embedded in the design of the platform and used according to industry cases. Direct RUL DL direct estimation methods are very handy when we have a rich history of the evolution to failure of a large number of similar equipment / components from which DL neural models are able to learn different failure patterns specific to various stages of wear. But the implementation of these models requires the archiving of huge amounts of data over a long period of time (years), from several similar sources and the knowledge of the state of degradation of historical samples (RUL) at the time of historical archiving. This is also true for similarity models that estimate the RUL based on the similarity of the degradation trajectories of the historical courts. The method requires the calculation of the status indicator based on the *RUL / max_RUL* degradation state and the construction of the degradation model for each instance with statistical methods. The degradation trajectory similarity models apply to those systems that are under-maintained, the models intuiting the complete rolling pattern until the complete degradation of the system (running to failure). Also, in cases where we do not have a complete history of running to failure, but we know the lifespan of similar components, we can choose the traditional methods of survival modeling that model the failure time of the component from historical data (similar components from fleet), depending only on the number of hours of operation or possibly also on the state characteristics of the machine (operating mode, batch of its manufacture) using hazard functions.

From the perspective of the design and effective development of the CMMS platform for maintenance management, the thesis includes a series of original contributions as follows:

6. Creating a modular structure so that platform services can be accessed depending on the size of the company and maintenance strategies Preventive, Predictive and Maintenance-based Maintenance modules are built separately in the Model / View / Controller structure of the application and can be activated independently according to the needs of the company. The configurator, the Technical Base, the Utilities module, and the Setup module are basic modules that will be found in the basic component of the platform. The platform can configure and manage one or more work points with one or more production lines.

7. Realization of an asset configurator on the production lines according to a modular structure - components / subassemblies / machines.
With the help of the configurator, you can easily instantiate the equipment from one or more factories, just by selecting and adding the configured equipment. A much better visualization of the maintenance process is obtained as well as the monitoring of the equipment at component and subassembly level, not only at machine level.
The history of maintenance operations and the generated reports are highlighted at the level of machine code, subassembly code, component code.

8.Development of a production line configurator within an enterprise. Production lines are easily defined, equipment is added by selection once the asset structure is already defined in the configurator. By selecting a production line, you get a quick view of the real status of all the machines on the line, by reading the status of the equipment transmitted by the PLCs and displayed in the user interface.

9. Development of a spare parts and consumables management system. Each entity on the production line (machine, subassembly, or component) is identified by a unique ID. The history of component replacement operations (breakdown or planned replacement) as well as equipment maintenance operations builds a clear record of the movement of consumable flows within the company, because in each operation sheet, the technician marks the consumables used (quantities) and / or replacement parts.

10.Generation and planning of maintenance interventions, inspections, and corrective interventions, planning of maintenance teams, reports related to preventive interventions and their operations, automatic timekeeping of the workforce correlated with the working hours of technicians in an integrated system. The maintenance manager has an exact situation of the developer of the interventions in the company, through the records provided by the Intervention Generator. The generator will automatically create on the exact date, according to the setting in the Configurator, the intervention to be performed on each machine, with a warning window. The intervention can consist of one or more operations defined on the subassemblies of the machine and displays the due date. The application allows the assignment of teams for each intervention, the assignment of technicians for each operation as well as supervisors.
The operation of the scheduled tasks within the interventions, through the tools included in the Maintenance module, generates multiple reports related to the activities undertaken by staff in time intervals through time sheets, completion of operation sheets, tickets sent with various requests to departments, discussion history and the files attached to the activities, through the history of the actions performed on the platform captured in the Activities subsection, through the history of the training sessions and the online feedbacks from the questionnaires.
Asset reports are generated, with operations performed on the equipment or subassembly, in time intervals, which together with the status reports obtained by monitoring some status parameters complete the asset status log. In case of damage or inspections, the inspection sheets completed by the technicians generate the corrective interventions to be carried out, with all the necessary operations.

11. A complex visualization of the intervention; any intervention offers through the stored information views from several perspectives: condition, costs, team performance. The summary of the intervention contains a graphical illustration of the status and performance data, the tree structure of operations and fact sheets of operations with their statuses, Gantt chart of the intervention, tickets sent to departments, expenses, file library and discussion archive generated by the team. intervention. This structure gives a clearer view of the maintenance activity.

12. Online asset monitoring through a large cloud data retrieval system and implementation of anomaly detection algorithms, sending and displaying alarms.
The monitoring system allows an active view of the status of the equipment, by displaying the monitored parameters with the help of graphical panels Grafana. At the same time, historical information of the asset is obtained, by displaying all the interventions performed and the status reports regarding the operating / non-operating hours. The data accumulated at this level from the PLCs and from the mounted additional sensors are taken over by the anomaly detection algorithms for the evaluation of the normality state and the transmission of alarms, in case of the identification of some anomalies. The platform implements algorithms for detecting anomaly by unsupervised learning with Autoencoder-type neural networks, by classification using supervised learning feedforward neural networks or can implement the statistical model of Gaussian distribution, depending on industry cases.

13. Tablet application that implements AR technology to quickly identify subassemblies, send and receive maintenance history information when technicians are in the field for technical inspections or performing interventions.

14. Integration of predictive and preventive maintenance applications with the physical processes of effective maintenance operation, by transmitting and recording information directly from the field, through the team of technicians. By implementing codes on subassemblies and machines, which are easily scanned and identified by technicians during interventions, all necessary information on the status and maintenance history is accessed, together with the related technical documentation. At the same time, information is transmitted to the CMMS system and a history of operations, and a library of concrete fault cases is built. The Integrated Discussion Library (maintenance module) can create a solid response base over time by implementing AI algorithms, when investigating the cause in diagnosing faults or looking for information on performing maintenance operations.

15. Integrate a remote support video conferencing solution, accessed by desktop CMMS application, tablet, or AR glasses.
The video conferencing system is a means of communication between field technicians and supervisors and is also integrated with AR glasses for comfort during inspections or interventions.

16. Ticketing support for company departments. The ticketing system integrated in the platform generates a better communication between the maintenance, production, and procurement departments, for the rapid resolution of emergencies related to consumable orders, spare parts, stock inquiry, etc. It can be later extended to suppliers, for better horizontal implementation of the enterprise.

17. Technical libraries accessed at component / subassembly / machine / maintenance level, in various formats (electronic or video documents). The proposed system brings together libraries of technical documentation at several levels. There are file libraries built into the Configurator and Technical Base of the CMMS platform, by the administrators of these modules, who have

creation / editing rights. User-built libraries have been implemented in the Maintenance module - Files, Discussions and Operations Sheets sections, where they can attach various file formats available to all those who have access rights in sections. There are personal document libraries, accessible only to the logged in user, in the Utilities module - File Library section.

18. Tracking of maintenance-related expenses by detailed accounting of the cost of interventions. The implemented system allows a flexible way of calculating the expenses related to the maintenance operations and the labor expenses for each intervention and the constituent operations separately. The system quantifies the labor costs for maintenance, established by the time sheets and working times marked in the system, the costs of spare parts and consumables and other types of costs in the company for maintenance and processes. This system allows detailed reporting of component / subassembly / machine / operation / intervention / production line / technician / factory level costs.

19. Implement a predictive model library programmed in Python and integrated into Jupyter Notebook files.

20. Carrying out a case study to predict the failure of a component in a time window. The case study presented in Chapter 5, paragraph 5.2.7.1, performs the prognosis of the component in systems subject to preventive maintenance, as part of a regular component replacement process. Therefore, the presented method is not an RUL estimate, it is a method of predicting the failure of the component in a time window and it was considered that it can be a method that can be integrated in several industrial companies.

21. Carrying out a case study for the diagnosis of bearing defects using several deep learning algorithms. The case study presented in Chapter 5, paragraph 5.2.7.2, diagnoses bearing defects by vibration analysis using fault classification techniques with neural networks, without a prior selection of characteristics. The study perfectly identifies the examples that carry the anomaly and classifies the incipient defects in the bearings.

The significance of the case studies presented is enhanced by the fact that there is little information available on the application of ML and DL techniques in the industrial field, with a detailed description of the methods of diagnosis and forecasting and an example for computer implementation. Maybe it's because few businesses store data in the cloud, monitoring equipment over long distances. At present, artificial intelligence enjoys a wider applicability in the financial field, retail, banking, cyber-security, but too little is used in the industrial field with concrete examples and techniques. The study included and required the deepening of the dedicated computer language (Python / Anaconda, Octave, MATLAB, etc.) but also the implementation and adaptation of mathematical algorithms to the field of study.

Further Development Perspectives

In this thesis, an integrated approach to the maintenance process was attempted by implementing a flexible and modular software application that best meets the requirements of the smart industry. In the future, it is intended to expand this platform based on new case studies with different scenarios on predictive maintenance and the creation of more general models valid for more customers.

Maintenance and implementation research through the platform is far from over. The IOTIA platform is in a continuous development, completion, improvement, and automation process. Such a tool aims to become the interface of a cyber-physical system. There are various maintenance strategies and operational flows within the companies, and there are different

degrees of automation. The platform must reach a certain degree of flexibility for a more diverse integration.
In Romania, this type of service (e-maintenance) is just beginning. It is intended to develop this product/maintenance management tool, implementation in *SaaS* or *on mode Premises* for as many companies as possible, according to the business plan elaborated in the last year of doctoral study, within the Program *BeAntreprenor*. This plan was awarded the First Prize at the Polytechnic University of Bucharest.

BIBLIOGRAFIE

1. Ahmadi, A., C. Cherifi, V. Cheutet and Y. Ouzrout. 2017. A Review of CPS 5 Components Architecture for Manufacturing Based on Standards. In: *11th IEEE International Conference on Software*, Knowledge, Information Management and Applications (SKIMA 2017): (December) Colombo, Sri Lanka, 6 p. ffhal-01679977
2. Andrew Ng, Note de curs: *Machine Learning*, Stanford University
3. Appana, DK, Islam, MR, and Kim, J.-M. 2017. Reliable fault diagnosis of bearings using distance and density similarity on an enhanced k-nn. In: *Australasian Conference on artificial life and computational intelligence*. Editors M. Wagner, X. Li, and T. Hendtlass (Cham, Switzerland: Springer), 193–203
4. Arpit, D., Zhou, Y., Ngo, HQ, Govindaraju, V., 2016. Why Regularized Auto-Encoders Learn Sparse Representation?, Proceedings of the 33 rd International Conference on Machine Learning, New York, USA. JMLR: W&CP, Vol.48.
5. Arunraj, N., Maiti, J. 2007. Risk -based maintenance – Techniques and applications. *Journal of Hazardous Materials*, Vol. 142, Issue 3 (April): 653-661
6. Ayhan, B., Chow, M.-Y., and Song, M.-H. 2006. Multiple discriminant analysis and neural-network-based monolith and partition fault-detection schemes for broken rotor bar in induction motors. IEEE Trans. Ind. Electron. 53, 1298–1308. DOI:10.1109/tie.2006.878301
7. Baheti, R. and H. Gill. 2011. The Impact of Control Technology: Cyber-physical systems, IEEE, Control Systems Society.
8. Banabic, D. 2016. Industry 4.0 started. Is it ready Romania for the challenges of this new revolution? In: *Revista de Politica Științei și Scientometrie – Serie Nouă*, Vol.5 (September), 194-201
9. Benkercha, R., and Moulahoum, S. 2018. Fault detection and diagnosis based on c4.5 decision tree algorithm for grid connected pv system. Sol. Energy 173, 610–634. DOI:10.1016/j.solener.2018.07.089
10. Berti,, J., 2020. New IBM Maximo Application Suite provides monitoring, maintenance, and reliability applications in one single platform, available at: https://www.ibm.com /blogs/internet-of-things/iot-maximo-asset-management-suite/
11. Burrus, CS, Gopinath, R., Guo, H. 2013. *Introduction to Wavelets and Wavelet Transforms*, Rice University, Houston, Texas
12. Biau, G., Devroye, L. 2015. *Lectures on the Nearest Neighbor Method*. Springer
13. Biehl, M., Hammer, B., Verleysen, M., Villman, T. 2009. *Similarity Based Clustering*. Springer-Verlag [Berlin]
14. Biggio, L., Kastanis, I. 2020. Prognostics and Health Management of Industrial Assets: Current Progress and Road Ahead. *Frontiers in Artificial Intelligence*, Vol.3 (November).

DOI: 10.3389/frai.2020.578613

15. Bousdekis, A., Apostolou, D. and G. Mentzas. 2020. A human cyber physical system framework for operator 4.0 – artificial intelligence symbiosis. *Manufacturing Letters*, Vol. 25 (August): 10-15. https://doi.org/10.1016/j.mfglet.2020.06.001

16. Brettel, M., N. Friederichsen, M. Keller and M. Rosenberg. 2014. How Virtualization, Decentralization and Network Building Change the Manufacturing Landscape: An Industry 4.0 Perspective. *International Journal of Information and Communication Engineering* 8, no. 1.

17. Brownlee, J. 2020, *Introduction to Time Series Forecasting with Python,* Edition v1.8.

18. Bearing Vibration Data, Case Western Reserve University, https//csegroups.case.edu/bearingdatacenter/pages/12k-drive-end-bearing-fault-data

19. Chauvin, Y., Rumelhart, DE, 1995. *Backpropagation – Theory, Architectures, and Applications,* Psychology Press.

20. Chen, J., Patton, RJ, Nielsen, SB 1995. Model-based methods for fault diagnosis, *Transactions of the Institute of Measurement and Control (April),* DOI: 10.1177/014233129501700203

21. Chen XW, Zeng X and Alphen DV, 2006. Multi-class feature selection for texture classification. Pattern Recognit Lett; 27(14): 1685–1691.

22. Chen, A., Kurfess, T. 2017. A new model for rolling element bearing defect size estimation, Measurement 114 (Sept), DOI: 10.1016/j.measurement.2017.09.018

23. Chui, CK, Chen, G., 1998. *Kalman Filtering with Real-Time Applications,* Berlin, Springer

24. Ciocoiu, IB, 2001. Rețele neurale artificiale, Editura Cantes, Iași, ISBN 973-8173-16-7

25. Deac CN, Deac GC, Popa CL, Ghinea M., Coteț EC, 2017. Using Augmented Reality in Smart Manufacturing, *Proceedings of the 28th International DAAAM Symposium on Intelligent Manufacturing and Automation*, 7-11 Nov.2017, DOI: 10.2507/28th.daaam.proceedings.102, Festo Prize for Young Researchers and Scientists.

26. Deac, CN,, Deac, GC, Parpală, R., Popa, C., Coteț, C., 2020. Vibration Anomaly Detection using Deep Neural Network and Convolutional Neural Network, *International Journal of Modeling and Optimization* (February), DOI: 10.7763/IJMO.2021.V11.772

27. Deac, CN, Deac, GC, Parpală, R., Popa, C., Popa, C., Popescu, C., 2021.Vibration Anomaly Detection using Multivariate Time Series, *International Journal of Modeling and Optimization*

28. Debnath, L., 2001. *Wavelet Transforms and Time-Frequency Signal Analysis,* Springer Science + Business Media New York

29. Deng, N, Tian, Y., Zhang, C., 2012. *Support Vector Machines. Optimization Based Theory, Algorithms, and Extensions, Chapman and Hall/*CRC Press

30. Dertat, A., 2017, Applied Deep Learning – Part 3: Autoencoders, https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

31. Dong, S., Luo, T., Zhong, L., Chen, L., and Xu, X., 2017. Fault diagnosis of bearing based on the kernel principal component analysis and optimized k-nearest neighbor model. J. Low Freq. Noise Vib. Act. Contr. 36, 354–365. doi:10.1177/ 1461348417744302

32. Du, C., Yang, Z., 2015. Inspiration of transformation and upgrading of manufacturing industry in China from German Industry 4.0 strategy, *Research on Economics and Management*, Vol. 7: 82-87.

33. Dunn, A., 2015, The Fourth Generation of Maintenance,Assetivity Pty Ltd., https://plant-maintenance.com/articles/4th_Generation_maintenance.pdf

34. Fink, O., 2020. Data-driven intelligent predictive maintenance of industrial assets, in *Women in industrial and systems engineering.* Editor A. Smith (New York, NY: Springer), 589–605

35. Flovik, V., How to use Machine Learning for anomaly detection and condition monitoring, towardsdatascience.com

36. Flovik, V., Machine Learning for anomaly detection and condition monitoring, towardsdatascience.com

37. Galar, D., Goebel, K., Sandborn, P., Kumar, U., 2022. *Prognostics and Remaining Useful Life (RUL) Estimation. Predicting with Confidence*, CRC Press, Taylor & Francis Group.

38. Goto, F., 1989, Maintenance prevention, in Nakajima, S. (Ed.), Total Productive

39. Maintenance Development Program: Implementing Total Productive Maintenance,

40. Gryllias, KC, and Antoniadis, IA, 2012. A support vector machine approach based on physical model training for rolling element bearing fault detection in industrial environments.

41. Gröchenig, K., 2001. *The Short-Time Fourier Transform*, Springer Science + Business Media New York

42. Guillén, AJ,Crespo, A., Macchi, M., Gomez, J., 2016. On the role of Prognostics and Health Management in advanced maintenance systems, *Spanish Ministry of Science and Innovation unde*

43. Guyon, I. Weston, J. and Barnhill S., 2002. Gene selection for cancer classification using support vector machine. Mach Learn; 46(1–3): 389–422.

44. Islam, MMM, and Kim, J.-M., 2019. Automated bearing fault diagnosis scheme using 2d representation of wavelet packet transform and deep convolutional neural network. *Comput. Ind.*

45. Ismail, M., Klausen, A., 2018. Multiple Defect Size Estimation of Rolling Bearings using Autonomous Diagnosis and Vibrational Jerk – *The 7th World Conference on Structural Control and Monitoring*, China

46. Jardine, AK, Lin., D., Banjevic, D., 2006. A review on machinery diagnostics and prognostics implementing condition-based maintenance, *Mechanical Systems and Signal Processing* 20(7) (October). 10.1016/j.ymssp.2005.09.012

47. Jazdi, N., 2014. Cyber Physical Systems in the context of Industry 4.0. *IEEE, International Conference on Automation, Quality and Testing, Robotics* (May): 1-4. doi: 10.1109/AQTR.2014.6857843

48. Jia, Z., Liu, Z., Vong, C.-M., and Pecht, M., 2019. A rotating machinery fault diagnosis method based on feature learning of thermal images. *IEEE Access 7,* 12348–12359. doi:10.1109/access.2019.2893331no 065107

49. Jolliffe, IT, 2002. *Principal Component Analysis,* Second Edition, New York, Springer.

50. Kagermann, H., Helbig, J., Hellinger, A. and Wahlster, W., 2013. *Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0: Securing the Future of German Manufacturing Industry, Final Report of the Industrie 4.0 Working Group,* Forschungsunion, ACATECH - National Academy of Science and Engineering

51. Karumanchi, N., 2016. *Data Structure and Algorithmic - Thinking with Python*,

CareerMonk Publications, 390-399.

52. Khan, S., and Yairi, T., 2018. A review on the application of Deep Learning in system health management. Mech. Syst. Signal Process. 107, 241–265. doi:10. 1016/j.ymssp.2017.11.024

53. Kienzler, R., Manchev, N., Advanced Machine Learning and Signal Processing, *IBM Course* Coursera.org.

54. Kingma, DP, and Welling, M., 2013. Auto-encoding variational bayes. arXiv.

55. Kingma, DP and Ba, J., 2014. Adam: A Method for Stochastic Optimization. arXiv:1412.6980

56. Kirill, E., Ponteves, H., 2021, Deep Learning AZ™: Hands-On Artificial Neural Networks, *Udemy*

57. Kirkwood, JR, 2015. *Markov Processes.* CRC Press.

58. Kukanov, I., Siniscalchi, M., Li, K., 2016. Deep Learning with maximal figure-of-merit cost to advance multi-label speech attribute detection, *IEEE Spoken Language Technology Workshop* (December). doi: 10.1109/SLT.2016.7846308

59. Laurence Moroney, Introduction to TensorFlow for Artificial Intelligence, Machine Learning and Deep Learning, Course 1 of 4 in the *TensorFlow in Practice Specialization,* deeplearning.ai

60. Lee, J., Bagheri, B. and Kao, HA, 2014. Recent Advances and Trends of Cyber-Physical Systems and Big Data Analytics in Industrial Informatics, *In: International Conference on Industrial Informatics* (July). doi: 10.13140/2.1.1464.1920

61. Lee, J., Bagheri, B. and Kao, HA, 2015. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters* 3 (January): 18-23.

62. Lee, J., Bagheri, B., 2015. Cyber-Physical Systems in Future Maintenance. 9th *WCEAM Research Papers* (February): 299-305.

63. Lei, J., Liu, C., and Jiang, D., 2019. Fault diagnosis of wind turbine based on long short-term memory networks. Renew. Energy 133, 422–432. doi:10.1016/j. renene.2018.10.031

64. Li, G., Deng, C., Wu, J., Xu, X., Shao, X., and Wang, Y., 2019. Sensor data-driven bearing fault diagnosis based on deep convolutional neural networks and s-transform. *Sensors 19,* 2750. doi:10.3390/s19122750

65. Liu, Z., Zuo, MJ, and Xu, H., 2013. Feature ranking for support vector machine classification and its application to machinery fault diagnosis. Proc. IME C J. Mech. Eng. Sci. 227, 2077–2089. doi:10.1177/0954406212469757

66. Lu, P.-J., Zhang, M.-C., Hsu, T.-C., and Zhang, J., 2001. An evaluation of engine faults diagnostics using artificial neural networks. J. Eng. Gas Turbines Power 123, 340. doi:10.1115/1.1362667

67. Lu, C., Wang, Z., and Zhou, B., 2017. Intelligent fault diagnosis of rolling bearing using hierarchical convolutional network based health state classification, *Adv. Eng. Informat.* vol.32, pp.139-157

68. Luttmann, L., Mercorelli, P., 2021. Comparison of Backpropagation and Kalman Filter-based Training for Neural Networks, Preprints, 2021040523 (doi: 10.20944/preprints202104.0523.v1)

69. Macaulay, S., 1988. Amazing things can happen if you ... Keep it clean, *Production,* May, pp.72-74

70. Malhotra, P., TV, V., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., Shroff, G., 2016. Multi-Sensor Prognostics using an Unsupervised Health Index based on LSTM Encoder-Decoder, *TCS Research*, New Delhi, India (August)

71. Marta, S., Angelink, M., Juergen, M., Biffl, S., Tu, W., 2018. Collective Intelligence Aspects of Cyber-Physical Social Systems: Results of a Systematic Mapping Study. In Proceedings of the ACM Collective Intelligence Conference, Zurich, Switzerland, 7–8 July 2018; pp. 1–4.

72. Marti, K., 2008. *Stochastic Optimization Methods,* Springer.

73. Mehrotra, KG, Mohan, CK, Huang, H., 2017. *Anomaly Detection Principles and Algorithms,* Syracuse, NY, USA, Springer

74. Miller, FP, Vandome, A., McBrewster, J., 2010. *Mahanalobis Distance*, VDM Publishing

75. Mobley, RK, 2001, *Plant Engineer's Handbook,* Elsevier Inc.

76. Mobley, RK, Higgings, L and Smith, R., 2002, *Maintenance Engineering Handbook*, 6th Edition, McGraw-Hill

77. Moubray, J., 1991. *Reliability-Centred Maintenance*, 2nd Edition, Great Britain, Butterworth-Heinemann

78. Moosavi, SS, N'Diaye, A., Djerdir, A., Ait-Amirat, Y., and Arab Khaburi, D., 2016). Artificial neural network-based fault diagnosis in the AC-DC converter of the power supply of series hybrid electric vehicle. IET Electr. Syst. Transp. 6, 96–106. doi:10.1049/iet-est.2014.0055

79. Mosterman, P., Zander, J., 2016. Industry 4.0 as a Cyber-Physical System Study. *Software and Systems Modelling*, vol (October). doi 10.1007/s10270-015-0493-x

80. Mujica, L., Guemes, A., Rodellar, J., 2008. PCA based measures: Q-statistic and T2-statistic for assessing damages in structures, *Structural Health Monitoring,* doi: 10.1177/1475921710388972

81. NASA, 2000. Reliability Centered Maintenance Guide for Facilities and Collateral Equipment. National Aeronautics and Space Administration, Washington, DC

82. Ng, A., 2011, Sparse autoencoder. CS294A Lecture notes 72, 1–19

83. Ng, A., 2018, Machine Learning, *Online Course authorised by Stanford University and offered through Coursera*

84. Nowlan, F. et al., 1978. *Reliability-Centred Maintenance,* United Airlines, San Francisco, California, US Department of Commerce.

85. O'Donovan, K., P. Leahy, K. Bruton and O'Sullivan, DTJ, 2015. An industrial big data pipeline for data-driven analytics maintenance applications in large-scale smart manufacturing facilities. *Journal of Big Data,* 2:25(1), 1. doi:10.1186/s40537-015-0034-z

86. Oppenheim, AV, Willsky, AS, Nawab, SH, 1983. *Signals and Systems.* New Jersey, Prentice Hall.

87. Pang, G., Shen, C., Cao.,L., Hengel, AVD., 2017. Deep Learning for Anomaly detection: A review, ACM Computing Surveys, Vol.54

88. Patton, RJ, Frank, PM and Clark, RN (eds), 1989. Fault Diagnosis in Dynamic Systems, Theory and Application, Control Engineering Series, Prentice Hall, London.

89. Peng, H., Long, F., Ding, C., 2005. Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(8):1226-38,

DOI:10.1109/TPAMI.2005.159

90. Pintelon, L., Gelders, L., 1992. Maintenance management decision making, *European Journal of Operations Research*, no.3,Vol.58: 301-317

91. Prihatno, AT, YM Jang and H. Nurcahyanto. 2020. Smart Factory Based on IoT Platform, Conference Paper (September)

92. Ray, A. and Tangirala, S., Stochastic modeling of fatigue crack dynamics for on-line failure prognostics. *IEEE Transactions on Control Systems Technology*, 4(4):443– 451, JUL 1996. ISSN 1063-6536.

93. Rokach, L., Maimon, O., 2014. *Data Mining with decision tress*. World Scientific Publishing Co Pte Ltd

94. Rong, Z., Shandong, D., Xin, N., Shiguang, X., 2018. Feedforward Neural Network for Time Series Anomaly Detection, Cornell University, arXiv:1812.08389 [cs.LG]

95. Saha, B., Goebel, K. and Christophersen, J., 2009. Comparison of prognostic algorithms for estimating remaining useful life of batteries. *Transactions of the Institute of Measurement and Control*, 31(3-4):293–308, JUN-AUG 2009. ISSN 0142-3312. doi: 10.1177/0142331208092030.

96. Sakthivel, NR, Sugumaran, V., and Babudevasenapati, S., 2010. Vibration based fault diagnosis of monoblock centrifugal pump using decision tree. Expert Syst. Appl. 37, 4040–4049. doi:10.1016/j.eswa.2009.10.002

97. Samanta, B., 2004. Gear fault detection using artificial neural networks and support vector machines with genetic algorithms. Mech. Syst. Signal Process. 18, 625–644. doi:10.1016/s0888-3270(03)00020-7

98. Saxena, A., Roychoudhury, I., Celaya, JR, Saha, S., Saha, B., Goebel, K., 2010. Requirements specifications for prognostics: An overview. AIAA Infotech at Aerospace 2010, art.no. 2010-3398

99. Shah, TN, Khan, MZ, Mumtaz, A., Khan, B., 2020. CART, J-48graft, J48, ID3, Decision Stump and Random Forest: A comparative study. In: *University of Swabi Journal (USJ); Open access*, (November).

100. Shamrat, JM, Hasib, KM, Siddique, A., Yadav, A., 2021, Performance evaluation among ID3, C4.5, and CART Decision Tree Algorithms. In: *International Conference on Pervasive Computing and Social Networking, salem, 19-20,* India (March)

101. Stone, JV, 2013. *Bayes's Rule: A Tutorial Introduction to Bayesian Analysis*, Sebtel Press

102. Sufian, AT, Abdullah, BM, Ateeq, M., Wah, R. and Clements, D., 2019. A Roadmap towards the Smart Factory. In: *12th International Conference on Developments in eSystems Engineering* (October): 978-983. doi:10.1109/DeSE.2019.00182

103. Sullivan, G., Pugh, R. and Hunt, W., 2010. *Operations & Maintenance Best Practices - A Guide to Achieving Operational Efficiency*. Release 3.0., Pacific Northwest National Laboratory for the Federal Energy Management Program, US Department of Energy, Vers.3: 49-57

104. Sundararajan, D., 2001. *The Discrete Fourier Transform. Theory, Algorithms and Applications*, https://doi.org/10.1142/4610, World Scientific.

105. Svard, C., 2009. *Residual Generation Methods for Fault Diagnosis with Automotive Applications*, Linköping, Sweden, LiU-Tryck.

106. Tan, CC, 2009, *Autoencoder Neural Networks : A Performance Study Based on Image Reconstruction, Recognition and Compression*, LAP Lambert Academic Publishing.

107. Tian, Y., Yuan, J., 2019. An intelligent fault diagnosis method using GRU neural network toward sequential data in dynamic processes. *Processes 7*, 152. doi:10. 3390/pr7030152

108. The Economist, 2012. Special report: Manufacturing and innovation, A third indistrial revolution (April), https://web.mit.edu/pie/news/Economist.pdf

109. Vachtsevanos, G., Lewis, F., Roemer, M. and A. Hess., 2006. *Intelligent fault diagnosis and prognosis for engineering systems*. John Wiley & Sons, Inc., 1 edition.

110. Verma, V., 2020. A comprehensive guide to Feature Selection using Wrapper methods in Python, *Data Science Blogathon* (https://www.analyticsvidhya.com/blog/2020/10/a-comprehensive-guide-to-feature-selection-using-wrapper-methods-in-python/)

111. Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A., 2008. "Extracting and composing robust features with denoising autoencoders," In *Machine Learning, proceedings of the twenty-fifth international conference (ICML 2008)*, Helsinki, Finland, June 5–9, 2008, 1096–1103. doi:10.1145/1390156.1390294

112. Vinzi, VE, Chin, WW, Henseler, J., Wang, H., 2010. *Handbook of Partial Least Squares. Concepts, Methods and Applications*, Springer.

113. Vegard Flovik, 2019. Machine Learning for anomaly detection and condition monitoring, *Towards Data Science* (towardsdatascience.com/machine-learning-for-anomaly-detection-and-condition-monitoring-d4614e7de770)

114. Xia, M., Li, T., Xu, L., Liu, L. and CW de Silva, 2018. "Fault diagnosis for rotating machinery using multiple sensors and convolutional neural networks," IEEE/ASME Trans. Mechatronics, vol. 23, no. 1, pp. 101– 110, (February)

115. Xia, M., Li, T., Shu, T., Wan, J., de Silva, CW, and Wang, Z., 2019. A two-stage approach for the remaining useful life prediction of bearings using deep neural networks. IEEE Trans. Ind. Inf. 15, 3703–3711. doi:10.1109/tii.2018. 2868687

116. Zhao, Z., Anand, R., Wang, M., 2019. Maximum Relevance and Minimum Redundancy Feature Selection Methods for a Marketing Machine Learning Platform. *arXiv:1908.05376v1 [stat.ML] (August)*

117. Zhang, B., Georgoulas, G., Orchard, M., Saxena, A., Brown, D., 2008. Rolling Element Bearing Feature Extraction and Anomaly Detection Based on Vibration Monitoring, 16 th Mediterranean Conference on Control and Automation Congress Centre, June 25-27.

118. Zhang, S., Wang, B., Habetler, T., 2020. Deep Learning Algorithms for Bearing Fault Diagnostics – A Comprehensive Review, *arXiv:1901.08247v3 [cs.LG]* (February)

119. Wang, B., Tao, F., Fang, X., Liu, C., Liu Y. and Freiheit, T., 2020. Smart Manufacturing and Intelligent Manufacturing: A Comparative Review. *Journal Elsevier,* China

120. Wang, S., Li, J., Wan, D., Zhang, C., 2016. Implementing Smart Factory of Industry 4.0: An Outlook. *International Journal of Distributed Sensor Networks,* (January).

121. Wang, T., 2010. Trajectory Similarity Based Prediction for Remaining Useful Life Estimation, UMI 3432353, ProQuest LLC. (August)

122. Wang, Z., Zhang, Q., Xiong, J., Xiao, M., Sun, G., and He, J., 2017. Fault diagnosis of a rolling bearing using wavelet packet denoising and random forests. IEEE Sensor. J. 17, 5581–5588. doi:10.1109/jsen.2017.2726011

123. Wen, L., Dong, Y. and L. Gao, 2019. A new ensemble residual convolutional network for remaining useful life estimation, *Mathematical Biosciences and Engineering*, 16(2): 862–880. DOI: 10.3934/mbe.2019040

124. Weeks, M., 2007. *Digital Signal Processing using Matlab and Wavelets*, Infinity Science

Press LLC, Hingham, Massachhusetts

125. Wong, KK, 2019. Mastering Partial Least Squares Structural Equation Modeling (PLS-SEM) with SmartPLS in 38 hours, *iUniverse*, Bloomington.

126. Wortmann, F., Flüchter, K., 2015. Internet of things. Business & Information Systems Engineering, 57 (March): 221-224.

127. Yao, X.; Lin, Y., 2016. Emerging manufacturing paradigm shifts for the incoming industrial revolution. Int. J. Adv. Manuf. Technol, 85, 1665–1676.

128. Yang, B.-S., Di, X., and Han, T. 2008. Random forests classifier for machine fault diagnosis. J. Mech. Sci. Technol. 22, 1716–1725. doi:10.1007/s12206-008-0603-6

129. Yang, B.-S., Han, T., and Hwang, WW 2005. Fault diagnosis of rotating machinery based on multi-class support vector machines. J. Mech. Sci. Technol. 19, 846–859. doi:10.1007/BF02916133

130. Yin, D., Ming, X. and Zhang, X., 2020. Understanding Data-Driven Cyber-Physical-Social System (D-CPSS) Using a 7C Framework in Social Manufacturing Context, *Sensors* – Open Access Journal from MDPI, no.20(18), 5319*:* (September). doi.org/10.3390/s20185319

131. https://www.emerson.com/en/us/news/corporate/iiot-survey

132. https://www.ibm.com/analytics/us/en/technology/predictive-analytics/

133. https://www.coursera.org/learn/tensorflow-sequences-time-series-and-prediction

134. https://www.coursera.org/advanced-machine-learning-signal-processing

135. https://www.coursera.org/learn/introduction-tensorflow

136. https//csegroups.case.edu/bearingdatacenter/pages/12k-drive-end-bearing-fault-data, Bearing Vibration Data, Case Western Reserve University

137. http://www.alteryx.com/?utm_source=PredictiveAnalyticsToday&utm_medium=Review&utm_campaign=PAT

138. https://www.continuum.io/

139. https://www.dataiku.com/

140. http://dmway.com/

141. https://mediacenter.ibm.com/media/t/1_jg5ynql0

142. https://my.vertica.com/register/

143. http://www.oracle.com/technetwork/database/options/advanced-analytics/odm/overview/index.html

144. https://turi.com/

145. https://greenwavesystems.com/

146. https://www.h2o.ai/

147. https://www.datarobot.com/

148. http://www.sap.com/pc/analytics/predictive-analytics/software/predictive-analysis/index.html?utm_source=PredictiveAnalyticsToday&utm_medium=Review&utm_campaign=PAT

149. https://www.getmaintainx.com/learning-center

150. https://www.grafana.com