Intellinjector: Technology Platform for Information Management and Operation of a Plastic Injector Park

Lucas Costa * **Caio Cesar** ** **Cilis Benevides** *** **Isabel Frota**
Paulo R. C. Araújo Marcelle Maia Pedro Oliveira Rhuan Santiago
Rafael Araújo Andrio Antônio Henrique Inneco Jânio Oliveira
Alberto Alexandre André L. C. Araújo Rogério D. Guerra

* *IFCE, Fortaleza, CE, (e-mail: lucas.costa@lit.ifce.edu.br).*
** *FUNCEPE, Fortaleza, CE, (e-mail: cchc32@gmail.com)*
*** *FUNCEPE, Fortaleza, CE, (e-mail: cilis.benevides@gmail.com)*

Abstract: The acquisition of information from automated systems and their rapid processing is essential for industry productivity and competitiveness, especially in the age of technologies such as the Internet of Things (IoT) and Industry 4.0. Information obtained from the equipment at the industrial park can be used to assist decision making, improve productivity, reduce waste, and other functionalities. It directly or indirectly aims at reducing costs and increasing the quality of the product. Companies that manufacture electronic meters, such as Eletra Energy Solutions, have a plastic injector park. This study will present a technological platform for the information management of the Eletra's plastic injection park operation. This platform is constituted of hardware, software, and embedded electronics. It is based on computational techniques, such as artificial intelligence and inference, to communicate, obtain data, and process signals from plastic injection machines. From these technologies, the data from the injection machines are transformed into commands and valuable information to automate the operation of Eletra's injection molding park, as well as to generate for the operational chain (operators, supervisors, etc.) information and alerts to be dealt with preventive and corrective way.

Keywords: Automation; Artificial Intelligence; Industrial Management and Operation; Real-Time; Error Detection.

1. INTRODUCTION

The information acquisition from automated systems is essential for industry productivity and competitiveness. Industries with large-scale productions increasingly rely on the best possible functioning of their productive sectors, and technology comes as a great ally when innovating. Information obtained from equipment at industrial parks is used to assist decision making, improve productivity, reduce waste, and other functionalities. It directly or indirectly aims at reducing costs and increasing the quality of products. Companies that manufacture electronic meters, such as Eletra Energy Solutions, have a plastic injector park with dozens of injector machines. They need to reduce problems, reduce waste, and improve the quality of their injected parts.

Some failures can occur during an injection process: defective parts, injection time longer than configured time, increased material waste, among others. Much of these problems are detected long after they happen, and professionals can not analyze all the injector machine parameters to identify failures as they occur. More experienced professionals can diagnose and specify what types of problems are happening based on the data shown by the injector screen. Also, depending on the defect, they can stop the machine to correct a particular fault and avoid waste and damage.

Injector machine operators need to know, in real-time, the problems that occur at the injection park. Based on the analysis of parameters informed by the injector machine, a system equipped with artificial intelligence is capable of stopping the machine and sending alerts to operators to correct possible failures. Thus, this article presents an innovative and unprecedented technology platform composed of software, hardware, and electronics. This platform performs a non-offensive and integrated communication with all plastic injectors to acquire their operating data. Based on this accurate and up-to-date information, the parameter control and monitoring from each injection operation are possible. The platform presented in this article, called Intellinjector, uses artificial intelligence concepts for learning technique to correct failures and, in some cases, stop the machines from reducing waste, number of defective parts, and consequently increase productivity.

The relevant contributions of this work can be summarized by the set of hardware and software that includes a hardware for interfacing data with all injector machines, supervisory software for communication interface between the Programmable Logic Controller (PLC) and the injec-

⋆ Reconhecimento do suporte financeiro deve vir nesta nota de rodapé.

tor machine panel, data acquisition software that allow operating data visualization and status of each injector over the Internet or intranet, artificial intelligence software capable of learning from the injector machine behavior and suggesting to operators the best measures of utilization and prevention of failures, software that manage all injector machines, monitoring events related to their full operation, and generate alarms for situations that need corrective and preventive intervention of operators.

2. STATE OF ART REVIEW

The authors in (Müller et al. (2019)) present a method and a communication and knowledge management platform that provide a common understanding of the overall system and a simple, transparent, and media-free communication between all participants as well as efficient knowledge management for small and medium-sized enterprises. This platform can not communicate with equipment or machines, and it can not get any real-time information of the production chain. According to the authors in (Musina et al. (2018)), a digital platform was developed to solve both the production and economic problems of the agro-industrial sector of the economy through communication between the subjects of the system. The software developed by the authors is a particular web portal on the Internet, containing a classifier by type of objects, full-text search, description of objects with photos, video materials and reviews, statistics of the most essential socioeconomic indicators. This communication platform allows only information management about the agro-industry, and it can not interact with machines or equipment to monitor e management of the agro-industry processes. Hao et al. (2015) present an integrated information management platform based on currently advanced cloud computing and IOT (The Internet of Things) technologies for integrated management of chemical industrial parks. This platform can accomplish the whole process, real-time and dynamic follow-up and control of chemicals and hazardous materials, and provide adequate service for users' decision-making by mining and analyzing tremendous data of a chemical industrial park with Agent-middleware technology. This platform can communicate with machines and equipment, but it does not intelligently implement management and monitoring. The authors in (Shahzad et al. (2017)) present a smart IoT communication platform integrated into a SCADA system that significantly increases system efficiency, scalability and reduces cost. They modeled the IoT-SCADA system and deployed a security mechanism, employing a cryptography-based algorithm that provided a secure transmission channel. At the same time, each time communication occurred between the field devices in the SCADA system. The platform developed by the authors allows communication between the machines and server (SCADA), but it does not analyze information to detect the fault in the industrial processes. Wang et al. (2020) propose a collaborative architecture for industrial Internet platform (IIP) called industrial operation system (Ind-OS), which contains the industrial driver, digital thread, and micro-services to provide a better cooperative enterprise information system (EIS) environment for manufacturing systems. This platform aims to manage the interaction between the physical and cyber components of the industry. They proposed an industrial operation system (Ind-OS) for manufacturing systems with the architecture of IIP to enable easier collaboration of EISs. The model presented by the authors aims to facilitate the information flow between the industry sectors and actors. But it does not use any mechanism to analyze and actuate the industrial process based on the information.

3. METHODOLOGY

This study, based on some project execution steps, was essential to making an efficient platform, called Intellinjector, to monitor a plastic injector park. We executed the following steps to make a platform prototype:

- Requirements research, survey, and specification; modeling and design of the project to define the estimated backlog;
- Research, development and prototyping of hardware architecture for interfacing data with all injector machines;
- Research on the native sensor network and the need to complement it with external sensors;
- Equipment tests for interfacing data with all injector machines;
- Development of supervisory software to interface programmable logic controller (PLC) communication with the injector machine panel;
- Development of data acquisition software capable of enabling data visualization and status of each injector machine over the internet or intranet;
- Research and development of a module to act on an instruction set capable of correcting the injection process;
- Analysis and development of a computational intelligence module capable of learning from injection device behavior, and suggesting to operators the best measures of utilization and prevention of non-compliance;
- Development of software to manage all injector devices, understand events related to their full operation and generate alarms for those responsible for situations that need human intervention, both correctly and preventively.

3.1 Platform Hardware Architecture

Eletra company has 15 injector machines from two different manufacturers (Borche and Haitian), with 6 Haitian machines and 9 Borche machines. Haitian devices use the AK 668 Haitian controller model. Borche devices have 03 equipment using the CP033/T Keba controller model and 06 equipment that use CP033/Z Keba. The control panels used by the injector machines have 6 visualization screens with important information for the manufacturer and operators of Haitian equipment and 05 visualization screens for Borche's operator, according to the equipment used by Eletra's technical team at the plastic injection sector.

Intellinjector platform must capture data from all equipment in the Eletra's injector park and save them in a database, to be replicated on the platform. We installed a network structure in the injector park to connect each equipment to the Eletra solution internal network. Thus, the hardware prototypes developed by this project obtain the data and transport them to a centralized database server. After that, the platform analyzes and interprets this data to represent information of one or more actions performed on the injector machine at a given time.

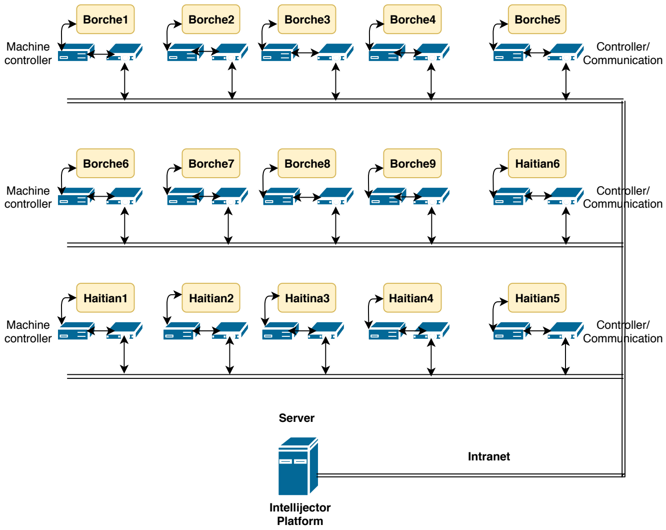


Figura 1. Network architecture and intellinjector platform

Based on some criteria such as specification techniques, cost-benefit, ability to be installed on high-temperature environments, efficient ethernet connection, and environmental testing with the injectors, we chose Beaglebone Black card to collect the injector machine data. That is, the hardware card was designed to intercept data communicated between the control panel and the injector's controller. Haitian's injector machines have a communication interface between its controller and the control panel, where all data travels. Thus, the authors developed this hardware to monitor communication and find a communication pattern. The process to obtain communicated data between the control panel and the injector's controller was carried out to assemble a hardware card that could be used generically in all Haitian's injectors. This equipment must also be easy to maintain and must meet all requirements defined in the Eletra's injector park integration process. Besides, the hardware board was also designed to meet the needs of Borchê's injector machines.

In Figure 1, the injector machines are connected to their controllers, and a hardware card captures the information passed from that controller to the control panel. In turn, this hardware card communicates with a server via ethernet. The firmware runs on this hardware card were also coded to collect data from the injectors. So, to feed the API that receives the protocol, a Python program (firmware), that runs on the BeagleBone Black card, reads data via serial (RS232), assembles the packets and sends its through an HTTP post. This firmware has three threads: ThreadRead, which is responsible for reading monitored data from serial communication. There is an array with data read by the serial interface. Every 200 bytes read, these values are added to this array; ThreadCommand, it reads data from the reading array and assembling the commands. When a command is identified, it is added to an array of commands and the bytes related to this command are removed from the reading array; ThreadPost sends "command by command"to the API and removes the commands sent from the command array. The protocol, the reading date, the command type, and which injector that command belongs are sent in the same packet.

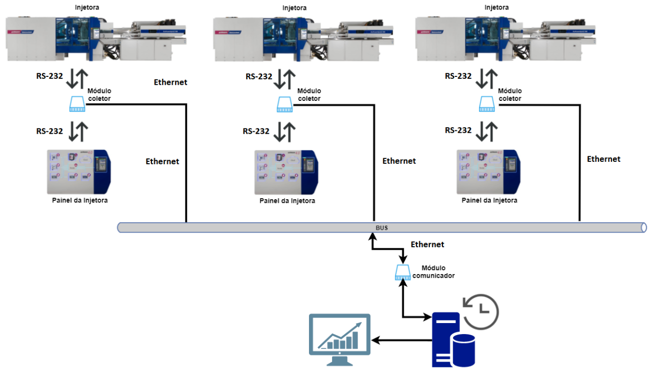


Figura 2. Network architecture and Intellinjector Platform

According to Figure 2, the project architecture contains the collector modules, communication modules, and the data server. The collector module monitors the message exchange between the injector machine controller and the panel, providing the input data to the server. Also, this module, in the future, may act as an actuator. The communication module facilitates the communication between the collector module and the server through the bus.

3.2 Development of Protocols

After the assembling of hardware cards, the second step of the research was to program the interpretation protocol of the data collected from the Haitian and Borche injector machines and the programming code of the data collection process obtained by the injector machines. So at this stage of the research, some tasks were carried out, such as the development of the protocol for interpreting data collected from injector machines; elaborating of interpretation protocol technical documentation; elaborating the documentation of the data collection process; programming code of the data collection process.

At this stage, the first step was to survey the key features used by Eletra's injector park operators. The functionalities obtained at the output screens, where all the injector operations and configurations are displayed, have the following information: 1) equipment operations as well as operating configuration information; 2) the main screen displays the general operation information of the injector in real-time; 3) the quality screen is a table formed by fields that indicate parameters of the machine's operating state in real-time, where each field has a range of values that may vary according to the product developed.

The functionalities obtained at the input screen are the following. 1) ranger x mold table; 2) summary of the whole process; 3) main screen (home) displays operating status; 4) injection and dosage parameter graph; 5) information regarding injection parameters such as time, velocity, pressure, stroke, repression; and dosage parameters such as back-pressure, pressure, velocity and dosage stroke; 6) injector preparation information screen; 7) lubrication information screen; 8) informative screen of injected pieces made; 9) final part injection process screen; 10) temperature information screen; 11) alert screen that displays machine's information such as alarm description, the time it started and ended if the process escapes the parameters; 12) production/time screen displays the calculation information used (cycle/time) on the screen.

With the information produced by the injector machines and after the data collected by the hardware cards, the following methodology was used to find the protocol standard. a) collect the data; b) interpret the data; c) identify and document the pattern of the messages collected and their meaning; c) develop an API to understand the collected data and transform it into information.

The interpretation process of the collected data to identify the injector's communication pattern was a very arduous process, and it followed the following steps. 1) the essential input and output functionalities were defined for the information management and operation process of the ELETRA's plastic injector park; 2) based on the operation of each activity, we verified, in the data collector, the following information: a) start and end of the collected messages, which were identified by the messages exchanged by the machine controller and the panel, with each message starting with the code "02"and end with the code "03"; b) message content, which after the start-byte message can be divided into 2-byte blocks, where each block represents a variable, with BLOCK 1 indicating the message size and the last one (calculated from message size) the checksum; c) checksum, which is the variable responsible for checking whether the message is corrupted or not; d) message size and message type, which are responsible for defining the number and type of message blocks. As a result of this research execution stage, we write a technical document with the protocol mapping of the injector machines with the main functionalities that will be executed in the ELETRA's plastic injector park information management and operation platform. Based on this protocol technical documentation, we develop the Eletra-Intellinjection-API. This API is an API-RestFul service, responsible for receiving, storing, and converting data from the injector hardware. This data is sent in the byte-array format and is stored in the database without any processing. After that, this data is processed and transformed into information regarding the operations, functions, and status of the injector machine. The hardware card uses the method 'host:8001/api/protocol/PostList', which receives a list of commands, that are read by the hardware card, and performs the interpretation process of the collected data. Subsequently, this interpreted data is returned by the API with the information regarding the quality and temperature table values.

3.3 Development of Software Platform

At this research stage, some activities were carried out such as 1) functional and non-functional requirement and architecture and database documents of the Eletra's injector park management, control and monitoring platform; 2) business rules of the Eletra's injector park management, control, and monitoring platform; 3) source code and technical/user manual of the Eletra's injector park management, control, and monitoring platform.

Before developing the source code of the Eletra's injector park management, control, and monitoring platform, we had to propose its architecture. The architecture of the Eletra's injector park management, control, and monitoring platform (called Intellinjector) was designed to have an understanding of the system components and their interrelationships. So, the number of logical and physical layers (and dependency between them), and the definition of a communication protocol were discussed with the Eletra's technical team, so that the solution was compatible and had a good interaction with the Eletra's legacy systems. Therefore, the development of the architecture was based on the Eletra's Framework (called Eletra EA). We proposed the architecture considering, in all modules of the Intellinjector platform, aspects such as performance, reliability, extensibility, scalability, and safety. The following programming languages and frameworks were used: C , .net, mvc 5 with .NET Framework, and Angular JS.

Implementing Platform Database The database modeling, at this stage, used the entity and relationship model. For this, the MySQLWorkBench software was used to create the model file. Because business rules regarding user-profiles and access levels are not well defined, the authors decided to create a permission system that can be managed directly by the system. This system was based on the creation of users, groups, and routes. Since a route is an action to be performed, and the relationship between a route and a group or user means that a particular individual has the necessary permission to perform that action. Using the MVC paradigm facilitates the management and deployment of the security module because it is validated with each user request as well as data is stored in a secure server-side session.

The structure defined in the database allows applying the journaling technique to save every change made in individual documents. Then a table is created to save the latest version of the documents and another to record each new change. The implementation of the platform database followed these steps: 1) assembly of the development environment; 2) implementation, validation, and documentation of the architectural solution; 3) monitoring and validation of implementation during versions, avoiding "architectural erosion"; 4) installation of the production environment; 5) configuration and assembly of the solution; 6) monitoring the solution.

This research stage resulted in the architecture of Eletra's injector park management, control, and monitoring platform (Intellinjector) with a database that followed development good practices ensuring integration, interoperability, modularization, module encapsulation of this platform. Also, architecture and database documentation was produced.

Implementing Eletra's Injector Park Management, Control, and Monitoring Platform (Intellinjector) In this research stage, we elaborated on the following documentation. 1) Functional and non-functional requirements. 2) Business Rules of the management, control, and monitoring platform. 3) Database architecture documentation of management, control, and monitoring platform of Eletra's injector plant. 4) Technical/user manual of the management, control, and monitoring platform. 5) And finally, we developed the management, control, and monitoring platform source code.

After the authors obtained the information in the survey of functional requirements, the Intellinjector Platform was developed with the following modules:

- Management module and user access controls, access profiles to the Intellinjector Platform.
- Operator vision module - It is responsible for presenting, in real-time, the information of the problems found in an injector machine, and the operator must indicate the notes referring to the production that is being executed.
- Problem solving module for problems found by operators in the injector machines - This module presents solutions to solve problems, according to the mapping of solutions offered by the operator team.
- Support module for problem solving - Based on artificial intelligence techniques, the platform learns the best solution to solve problems, according to the characteristics of the injector machines and the molds that are being used in production at the moment. It will direct more efficiently and quickly the solution to the problems presented by the operator team.
- Dashboard module - This module presents, in real-time, the information for all injector machines at the injector park, with general information, such as status of each injector machine (automatic, manual and off); injector machine temperature; planned X production graph; OEE graphs, which are based on indicators according to goals used at the company.
- Injector machine detailing module - In the Dashboard module of the Intellinjector Platform, operators can check, in real-time, the detailed information of each injector machine such as information about the machine, molds, cavities, technical sheet, OEE indicators of the injector machine, quality table (screen of the injector machine parameters), production X goal graph, status graph (availability), defect notes, and offenders.
- Technical sheet management module - This functionality aims to centralize the production control information of the injector machine park with all variables necessary to monitor the process.
- OFs (manufacturing orders) management module - On this screen, all OFs that have been finished by the operator will be displayed. Whether due to a pause in the operation or an operator error. It presents two possibilities: edit and reopen.
- Integration module with Eletra's internal systems (Eletra EA) - It aims to ensure the integration of the Intellinjector Platform routines with Eletra's ERP System.

The support module for problem solving was the most significant differential of this development process. It used MLP neural network techniques and the support vector machine (SVM) probabilistic classifier. These methods are the most used techniques for Industry 4.0, and they have vast documentation. We developed algorithms in C# language. The objective of this study has been achieved, because the authors implemented a learning solution to make the best way to solve problems found daily in the injector machines, taking into account the injector machine manufacturer, the injector machine model, and the mold responsible for production. The development of this module allowed the system to assist the operator in finding the most effective solutions in solving the problems encountered at the injector machine park. Since this learning is incremental, all the new solutions for problem solving, found at the injector machine park, will serve as a basis to record the experience and consequently be used as knowledge to assist operators in future problems. This implementation will allow new systematization and automation solutions to be implemented at Eletra's injector machine park.

The authors developed an essential routine that was implemented on the Intellinjector Platform as an application programming interface (API), which allowed software and devices developed on different platforms to establish communication with each other. This API was designed to collect the data packet sent by the hardware device that monitors the operation of the injector machine via TCP/IP. The payload (mass of data that has importance) is subsequently stored on the server to be further processed and displayed on the front-end application.

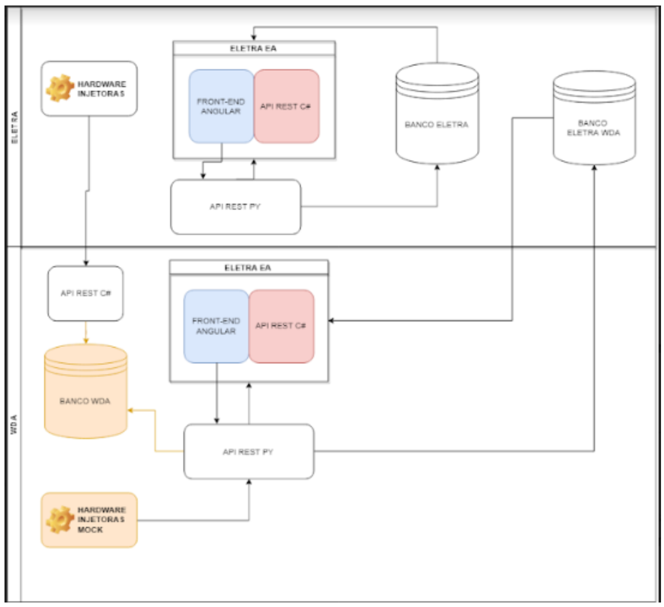


Figura 3. Intellinjector Platform in more detail

In Figure 3, API called API REST PY runs on the hardware card, and it collects data from the injector machines. So, to feed this API that receives the protocol, a Python program (firmware), that runs on the BeagleBone Black card, reads data via serial (RS232), assembles the packets and sends its through an HTTP post. API called API REST C# allows software and devices developed on different platforms to establish communication with each other. Databases called Eletra, Eletra/WDA, and WDA represent the Intellinjector Platform database, which is responsible for storing all the parameters, data, technical sheets, among others. And finally, the module called front-end angular is the interface responsible for showing information to operators, as shown in Figure 4.

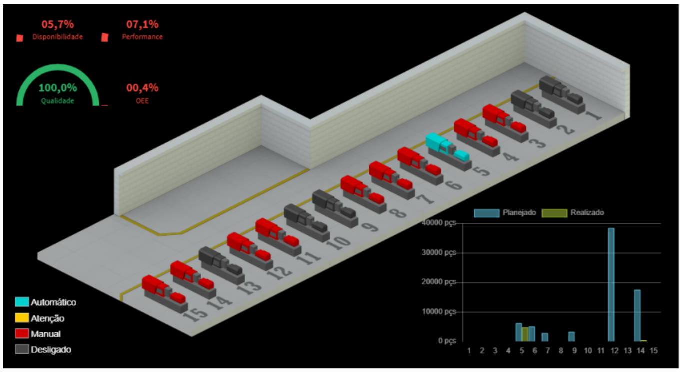


Figura 4. Intellinjector Platform front-end

4. RESULTS

Based on the software integration and the final tests, we see through the results that the Intellinjector Platform achieved the expected goals. The authors created services (Webservices) to guarantee the integration process of the management, control, and monitoring platform of Eletra's injector park (Intellinjector Platform) with other Eletra's systems, and possible integration in the future. Also, a final testing process was carried out for the entire Intellinjector Platform after the integration process. As a result of this step, the authors elaborated on an integration process and internal testing documentation of the Intellinjector Platform functionalities. Some results that compose this documentation will be presented next.

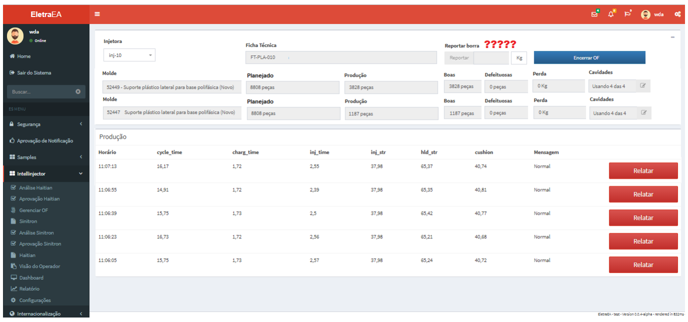


Figura 5. Testing operator vision module

Figure 5 represents the screen of the operator vision module. This module is crucial because it allows the operator to obtain essential information of the injector machine park OF such as injector machine identifier, mold type, number of planned x produced objects, process technical sheet, and production parameters. This functionality was extensively tested using the following methodology: a sequence with one hundred operator vision requests was performed, and, for each request, the response was compared with the parameters read from the human-machine interface (HMI) screen of injector machine. These tests presented as result accuracy of 98%.

We only carry out compliance tests to register process technical sheets. These test was not relevant to evaluate the platform performance, and it was run to check this functionality. As shown in Figure 6, this functionality allows the operator to set technical sheet parameters such as injector machine identifier, mold type, number of cavities, molding, and global temperatures, operation mode, among others.

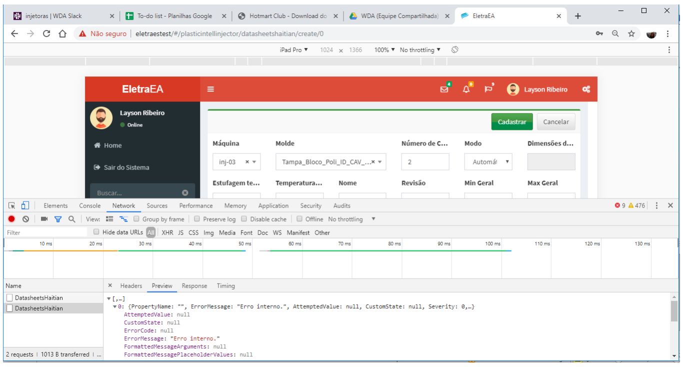


Figura 6. Testing process technical sheet registration

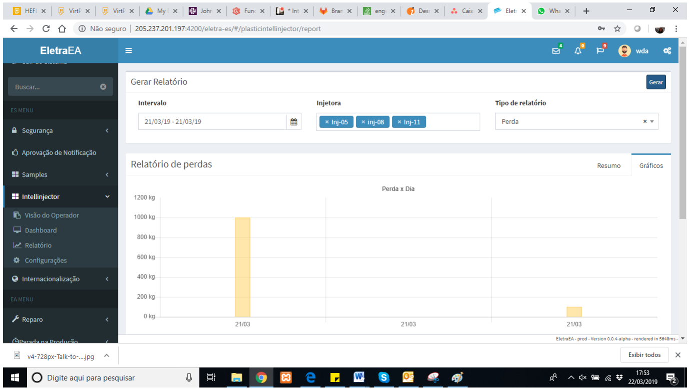


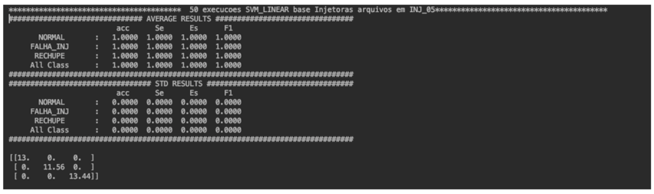
Figura 7. Testing loss report functionality

In the testing stage, the authors validated an essential functionality of the Intellinjector Platform, called report functionality. Figure 7 presents a shooting screen of the loss report functionality, whose parameters are date and time interval, report type, injector machine identifier, and a loss x defect graph. Four types of reports are possible: quality, loss, production, and offenders. This functionality was tested using the following methodology: in the report screen, the authors set the report type (in this case, loss report), injector machine identifier (inj-05, inj-08, inj-11), and the date and interval time (as an example 24h). After that, they run the functionality to obtain the loss response. This response was compared with the loss value collected in the injector machine HMI interface. We requested one hundred loss reports in Intellinjector Platform to compare with the losses obtained in the injector machine HMI interface. These tests resulted in an accuracy of 98%.

Many researchers have used SVM in several applications, but few have attempted to apply this classifier in error detection at the injector machine park. We have used this method to implement a learning solution to make the best way to solve problems found daily in the injector machines, taking into account the injector machine manufacturer, the injector machine model, and the mold responsible for production. This method allowed the system to assist the operator in finding the most effective solutions in solving the problems encountered at the injector machine park. After the feature extraction of the injector machine parameters when a error is detected, the classification process is followed; SVM (Support Vector Machine), in this work.

SVM are classifiers that are based on statistical learning theory, which takes into account the structural error minimization, calculated for the training vectors, and not only the minimization of the mean squared error (MSE). The following figures represent the shoot screen of the tests carry out on the some parameters: normal operation, injector failure, rechupe, rebarba, and all classes.

Resultados Linear com Normalização.



```
**************************************  50 execucoes SVM_LINEAR base Injetoras arquivos em INJ_05**************************************
############################## AVERAGE RESULTS ################################
                        acc      Se      Es      F1
      NORMAL      :   1.0000  1.0000  1.0000  1.0000
    FALHA_INJ     :   1.0000  1.0000  1.0000  1.0000
     RECHUPE      :   1.0000  1.0000  1.0000  1.0000
    All Class     :   1.0000  1.0000  1.0000  1.0000
################################################################################
################################ STD RESULTS ##################################
                        acc      Se      Es      F1
      NORMAL      :   0.0000  0.0000  0.0000  0.0000
    FALHA_INJ     :   0.0000  0.0000  0.0000  0.0000
     RECHUPE      :   0.0000  0.0000  0.0000  0.0000
    All Class     :   0.0000  0.0000  0.0000  0.0000
################################################################################

[[13.    0.    0.  ]
 [ 0.   11.56  0.  ]
 [ 0.    0.   13.44]]
```

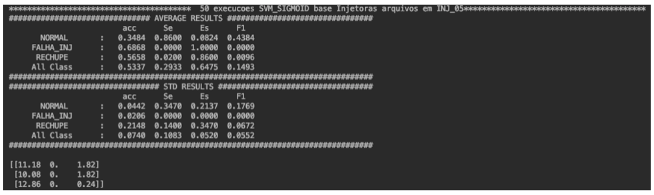
Resultados polinomial com Normalização.

```
**************************************  50 execucoes SVM_POLY base Injetoras arquivos em INJ_05**************************************
############################## AVERAGE RESULTS ################################
                        acc      Se      Es      F1
      NORMAL      :   1.0000  1.0000  1.0000  1.0000
    FALHA_INJ     :   1.0000  1.0000  1.0000  1.0000
     RECHUPE      :   1.0000  1.0000  1.0000  1.0000
    All Class     :   1.0000  1.0000  1.0000  1.0000
################################################################################
################################ STD RESULTS ##################################
                        acc      Se      Es      F1
      NORMAL      :   0.0000  0.0000  0.0000  0.0000
    FALHA_INJ     :   0.0000  0.0000  0.0000  0.0000
     RECHUPE      :   0.0000  0.0000  0.0000  0.0000
    All Class     :   0.0000  0.0000  0.0000  0.0000
################################################################################

[[13.    0.    0.  ]
 [ 0.   11.84  0.  ]
 [ 0.    0.   13.16]]
```

Figura 8. Linear and polynomial results with normalization

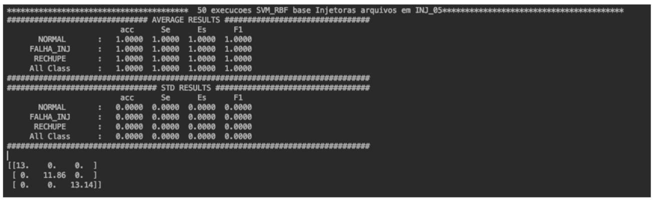
Resultados Senoidal com Normalização.



```
**************************************  50 execucoes SVM_SIGMOID base Injetoras arquivos em INJ_05**************************************
############################## AVERAGE RESULTS ################################
                        acc      Se      Es      F1
      NORMAL      :   0.3484  0.8600  0.0824  0.4384
    FALHA_INJ     :   0.6868  0.0000  1.0000  0.0000
     RECHUPE      :   0.5658  0.0200  0.8600  0.0096
    All Class     :   0.5337  0.2933  0.6475  0.1493
################################################################################
################################ STD RESULTS ##################################
                        acc      Se      Es      F1
      NORMAL      :   0.0442  0.3470  0.2137  0.1769
    FALHA_INJ     :   0.0206  0.0000  0.0000  0.0000
     RECHUPE      :   0.2148  0.1400  0.3470  0.0672
    All Class     :   0.0740  0.1083  0.0520  0.0552
################################################################################

[[11.18  0.    1.82]
 [10.08  0.    1.82]
 [12.86  0.    0.24]]
```

Resultados RBF com Normalização.



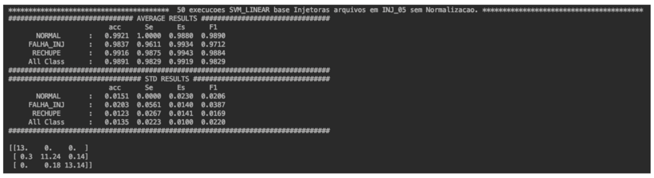
```
**************************************  50 execucoes SVM_RBF base Injetoras arquivos em INJ_05**************************************
############################## AVERAGE RESULTS ################################
                        acc      Se      Es      F1
      NORMAL      :   1.0000  1.0000  1.0000  1.0000
    FALHA_INJ     :   1.0000  1.0000  1.0000  1.0000
     RECHUPE      :   1.0000  1.0000  1.0000  1.0000
    All Class     :   1.0000  1.0000  1.0000  1.0000
################################################################################
################################ STD RESULTS ##################################
                        acc      Se      Es      F1
      NORMAL      :   0.0000  0.0000  0.0000  0.0000
    FALHA_INJ     :   0.0000  0.0000  0.0000  0.0000
     RECHUPE      :   0.0000  0.0000  0.0000  0.0000
    All Class     :   0.0000  0.0000  0.0000  0.0000
################################################################################

[[13.    0.    0.  ]
 [ 0.   11.86  0.  ]
 [ 0.    0.   13.14]]
```

Figura 9. Senoidal and RBF results with normalization

Figure 8 represents the average and standard results for fifty linear and polynomial kernel execution. These executions were based on the following operation types: normal, injector failure, rechupe failure, and all classes. This normalization was carried out to standardize (0 or 1) the input before entering in IA algorithm. Figure 9 shows the average and standard results for fifty sinusoidal and RBF kernel execution. These executions were based on the following operation types: normal, injector failure, rechupe failure, and all classes. This normalization was carried out to standardize (0 or 1) the input before entering in IA algorithm. We can observe that normalization in the sinusoidal kernel execution was not possible. Figures 10, 11 represent the average and standard results for fifty linear, polynomial, sinusoidal and RBF kernels execution without normalization. These executions were based on the following operation types: normal, injector failure, rechupe failure, and all classes. Without normalization, We can observe that the values are not standardized (0 or 1), making the IA algorithm training difficult. And normalization (for example, considering values 0 or 1) is necessary to facilitate the IA algorithm training.

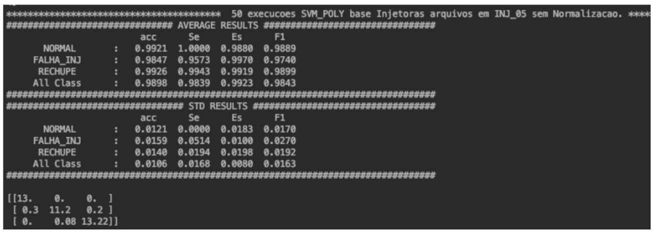
Resultados Linear sem Normalização.



```
**************************************  50 execucoes SVM_LINEAR base Injetoras arquivos em INJ_05 sem Normalizacao. **************************************
############################## AVERAGE RESULTS ################################
                        acc      Se      Es      F1
      NORMAL      :   0.9921  1.0000  0.9880  0.9890
    FALHA_INJ     :   0.9837  0.9611  0.9934  0.9712
     RECHUPE      :   0.9916  0.9875  0.9943  0.9884
    All Class     :   0.9891  0.9829  0.9919  0.9829
################################################################################
################################ STD RESULTS ##################################
                        acc      Se      Es      F1
      NORMAL      :   0.0151  0.0000  0.0230  0.0206
    FALHA_INJ     :   0.0203  0.0561  0.0140  0.0387
     RECHUPE      :   0.0123  0.0267  0.0141  0.0169
    All Class     :   0.0135  0.0223  0.0100  0.0220
################################################################################

[[13.    0.    0.  ]
 [ 0.3  11.24  0.14]
 [ 0.    0.18 13.14]]
```

Resultados polinomial sem Normalização.

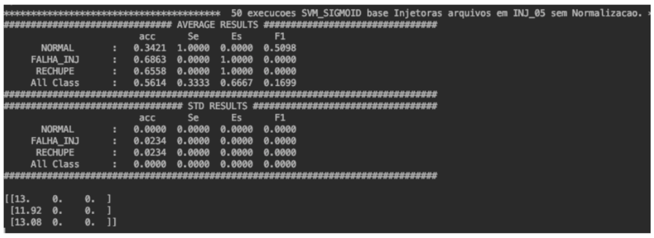


```
**************************************  50 execucoes SVM_POLY base Injetoras arquivos em INJ_05 sem Normalizacao. ****
############################## AVERAGE RESULTS ################################
                        acc      Se      Es      F1
      NORMAL      :   0.9921  1.0000  0.9880  0.9889
    FALHA_INJ     :   0.9847  0.9573  0.9970  0.9740
     RECHUPE      :   0.9926  0.9943  0.9919  0.9899
    All Class     :   0.9898  0.9839  0.9923  0.9843
################################################################################
################################ STD RESULTS ##################################
                        acc      Se      Es      F1
      NORMAL      :   0.0121  0.0000  0.0183  0.0170
    FALHA_INJ     :   0.0159  0.0514  0.0100  0.0270
     RECHUPE      :   0.0140  0.0194  0.0198  0.0192
    All Class     :   0.0106  0.0168  0.0080  0.0163
################################################################################

[[13.    0.    0.  ]
 [ 0.3  11.2   0.2 ]
 [ 0.    0.08 13.22]]
```

Figura 10. Linear and polynomial results without normalization

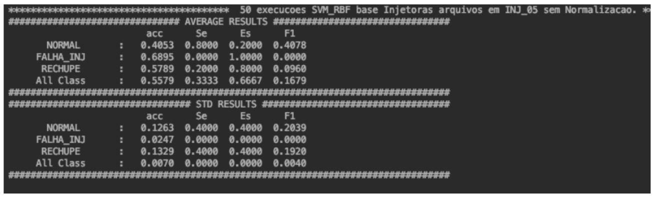
Resultados Senoidal sem Normalização.



```
**************************************  50 execucoes SVM_SIGMOID base Injetoras arquivos em INJ_05 sem Normalizacao. *
############################## AVERAGE RESULTS ################################
                        acc      Se      Es      F1
      NORMAL      :   0.3421  1.0000  0.0000  0.5098
    FALHA_INJ     :   0.6863  0.0000  1.0000  0.0000
     RECHUPE      :   0.6558  0.0000  1.0000  0.0000
    All Class     :   0.5614  0.3333  0.6667  0.1699
################################################################################
################################ STD RESULTS ##################################
                        acc      Se      Es      F1
      NORMAL      :   0.0000  0.0000  0.0000  0.0000
    FALHA_INJ     :   0.0234  0.0000  0.0000  0.0000
     RECHUPE      :   0.0234  0.0000  0.0000  0.0000
    All Class     :   0.0000  0.0000  0.0000  0.0000
################################################################################

[[13.    0.    0.  ]
 [11.92  0.    0.  ]
 [13.08  0.    0.  ]]
```

Resultados RBF sem Normalização.



```
**************************************  50 execucoes SVM_RBF base Injetoras arquivos em INJ_05 sem Normalizacao. *
############################## AVERAGE RESULTS ################################
                        acc      Se      Es      F1
      NORMAL      :   0.4053  0.8000  0.2000  0.4078
    FALHA_INJ     :   0.6895  0.0000  1.0000  0.0000
     RECHUPE      :   0.5789  0.2000  0.8000  0.0960
    All Class     :   0.5579  0.3333  0.6667  0.1679
################################################################################
################################ STD RESULTS ##################################
                        acc      Se      Es      F1
      NORMAL      :   0.1263  0.4000  0.4000  0.2039
    FALHA_INJ     :   0.0247  0.0000  0.0000  0.0000
     RECHUPE      :   0.1329  0.4000  0.4000  0.1920
    All Class     :   0.0070  0.0000  0.0000  0.0040
################################################################################
```

Figura 11. Senoidal and RBF results without normalization

5. CONCLUSION

This paper presented a system equipped with artificial intelligence that is capable of stopping the machine and sending alerts to operators to correct possible failures. These failures represent some losses in the production process. Thus, this article addressed an innovative and unprecedented technology platform composed of software, hardware, and electronics that perform a non-offensive and integrated communication with all plastic injector machines to acquire their operating data. Based on this accurate and up-to-date information, the parameter control and monitoring from each injector machine operation are possible.

ISSN: 2175-8905

DOI: 10.20906/sbai.v1i1.2840

The Intellinjector Platform, presented in this article, uses artificial intelligence (AI) concepts for learning techniques to correct failures and, in some cases, stop the machines from reducing waste, number of defective parts, and consequently increase productivity. This technique also allows implementing a learning solution to make the best way to solve problems found daily in the injector machines, taking into account the injector machine manufacturer, the injector machine model, and the mold responsible for production.

This work presented a set of hardware and software that includes hardware for interfacing data with all injector machines, supervisory software for a communication interface between the Programmable Logic Controller (PLC) and the injector machine panel, and data acquisition software. This platform allows operating data visualization and the status of each injector over the Internet or intranet. We use artificial intelligence software to learn from the injector machine behavior and suggesting to operators the best measures of utilization and prevention of failures. This software is capable of managing all injector machines, monitoring events related to their full operation, and generate alarms for situations that need the corrective and preventive intervention of operators.

After listing the problems encountered in the operational process of the injector machine park, the authors recorded the steps necessary for the development methodology of the Intellinjector platform. 1 - monitoring, observation, and recording of the activities carried out in the injector machine plant; 2 - process flow design; 3 - registration of the functional and non-functional requirements necessary to assemble the Intellinjector platform; 4 - meeting to validate the material produced by the author team together with Eletra Stakeholders; 5 - incremental elaboration of a documentation with the main processes, functional and non-functional requirements, and business rules.

The results presented in the previous section show that the Intellinjector platform developed by the authors improved the process of operating the injector machine park, allowing for a reduction in material losses, lower production costs, and an increase in productivity due to a faster recovery at a failure.

ACKNOWLEDGMENT

This work was partially supported by Fundação de Cultura e Apoio ao Ensino, Pesquisa e Extensão (FUNCEPE) and Ministério de Ciência,Tecnologia e Inovação (MCTI).

References

Rainer Müller, Leenhard Hörauf, Christoph Speicher, Johannes Obele. Communication and Knowledge Management Platform for Concurrent Product and Assembly System Development International Conference on Changeable, Agile, Reconfigurable and Virtual Production*, Vol. 28:107–113, 2019.

Dilara R. Musina, Anatoly G. Turganov, Sergey V. Kharitonov, Gylnara Z. Nizamova. Digital Communication Platform for the AgroIndustrial Complex. *International Conference on Communicative Strategies of Information Societ*, 2018.

Qian Hao, Furen Zhang, Zeling Liu, Lele Qin. Design of Chemical Industrial Park Integrated Information Management Platform Based on Cloud Computing and IOT (The Internet of Things) Technologies. *International Journal of Smart Home*, Vol. 9 no. 4,35–46, 2015.

Aamir Shahzad, Young-Gab Kim, Abulasad Elgamoudi. Secure IoT Platform for Industrial Control Systems. *International Conference on Platform Technology and Service (PlatCon)*, 2017.

Junliang Wang, Chuqiao Xu, Jie Zhang, Jingsong Bao, Ray Zhong. A collaborative architecture of the industrial internet platform for manufacturing systems. *Robotics and Computer-Integrated Manufacturing*, vol. 61, Elsevier, 2020.